Global optimization and simulated annealing

EINDHOVEN UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF MATHEMATICS AND COMPUTING SCIENCE

Memorandum COSOR 88-21

Global optimization and simulated annealing

by

A. Dekkers and E. Aarts

Eindhoven, the Netherlands

October 1988

GLOBAL OPTIMIZATION AND SIMULATED ANNEALING

Anton Dekkers[1)] and Emile Aarts[1-2)]

Abstract
In this paper we are concerned with global optimization, which can be defined as the problem of finding points on a bounded subset of $\mathbb{R}^n$ in which some real valued function f assumes its optimal (i.e. maximal or minimal) value.

We present a stochastic approach which is based on the simulated annealing algorithm. The approach closely follows the formulation of the simulated annealing algorithm as originally given for discrete optimization problems. The mathematic formulation is extended to continuous optimization problems and we prove asymptotic convergence to the set of global optima. Furthermore, we discuss an implementation of the algorithm and compare its performance with other well known algorithms. The performance evaluation is carried out for a standard set of test functions from the literature.

keywords: global optimization, continuous variables, simulated annealing.

1. INTRODUCTION

A global minimization problem can be formalized as a pair (S,f), where $S \subset \mathbb{R}^n$ is a bounded set on $\mathbb{R}^n$ and $f : S \rightarrow \mathbb{R}$ an n-dimensional real-valued function. The problem now is to find a point $x_{min} \in S$ such that $f(x_{min})$ is globally minimal on S. More specifically find an $x_{min} \in S$ with

$$\forall_{x \in S}: f(x_{min}) \leq f(x). \qquad (1.1)$$

Here we restrict ourselves to minimization. This can be done without loss of generallity, since a global maximum can be found the same way by reversing the sign of f.

Global optimization problems arise in many practical appplication areas such as for instance economics and technical sciences. Despite its importance and the efforts invested sofar, the situation with respect to algorithms for solving global minimization problems, is still unsatisfactory. Only for relatively simple functions f, where f is differentiable and the zero points of the derivative can be computed analytically, the situation is satisfactory.

---

1) *Eindhoven University of Technology*
P.O. Box 513
5600 MB Eindhoven, The Netherlands
2) *Philips Research Laboratories*
P.O. Box 80000
5600 JA Eindhoven, The Netherlands

For the minimization of more complicated functions one usually resorts to numerical solution methods. Many of these numerical methods cannot produce exact results, but merely approximate a global minimum by a local minimum that is 'close to' it, where 'close to' can be formalized by the following definitions:

Definition 1.1: For $\varepsilon > 0$, $B_x(\varepsilon)$ is the set of *points close to a minimal point*, i.e.

$$B_x(\varepsilon) = \{ x \in S \mid \exists_{x_{min}} : \|x - x_{min}\| < \varepsilon \}. \quad \square \tag{1.2}$$

Definition 1.2: For $\varepsilon > 0$, $B_f(\varepsilon)$ is the set of *points with a value close to the minimal value*, i.e.

$$B_f(\varepsilon) = \{ x \in S \mid \exists_{x_{min}} : |f(x) - f(x_{min})| < \varepsilon \}. \quad \square \tag{1.3}$$

Definition 1.3: For $\varepsilon > 0$, a point $x \in S$ is *near minimal* if

$$x \in B(\varepsilon) \tag{1.4}$$

where $$B(\varepsilon) = B_f(\varepsilon) \cup B_x(\varepsilon). \quad \square$$

Numerical global optimization methods can be divided into two classes: (i) deterministic and (ii) stochastic methods. In stochastic methods, the minimization process depends partly on probabilistic events, whereas in deterministic methods no probabilistic information is used.

The disadvantage of deterministic methods is, that they find the global minimum only after an exhaustive search over S and additional assumptions on f. The faster among these methods have the additional disadvantage that even more assumptions must be made about f, or that there is no guarantee for success (Rinnooy Kan & Timmer [1984]).

Stochastic methods, on the contrary, can be proved to find a global minimum with an asymptotic convergence guarantee in probability, i.e. these methods are asymptotically successful with probability 1. Furthermore, the computational results of the stochastic methods are, in general, far better than those of the deterministic methods (Gomulka [1978a]). For this reason we concentrate on stochastic methods.

An important problem in global minimization is to recognize a local minimum. To quantify this problem we need the following definition:

Definition 1.4: A *region of attraction* $B_{x_{loc}}$ is defined as a subset of S, surrounding a *local minimum* $x_{loc} \in S$, containing no point with a lower function value than x_{loc}, i.e.

$$\forall_{x \in B_{x_{loc}}} : f(x_{loc}) \leq f(x). \quad \square \qquad (1.5)$$

Clearly, applying a strict descending local search procedure to each point of $B_{x_{loc}}$ will yield x_{loc}.

Local minimality is no guarantee for global minimality. So a fundamental concern in global minimization is to avoid getting stuck in a local minimum.

Up to now, there are two classes of methods known to overcome this difficulty in stochastic minimization: the first class constitutes the so-called *two-phases methods*; the second class is based on *simulated annealing*.

In two-phases methods, the search for a global minimum is divided into two steps: firstly, a number of points is sampled (randomly) from S; secondly, for each of these points a local minimum is detected, i.e. for each point, the local minimum is determined of the region of attraction to which the point belongs, and each of these local minima is considered as a candidate for a global minimum. Determination of a local minimum is done by a local search procedure. Reviews of two-phases methods are given by Dixon & Szegö [1978] and Rinnooy Kan & Timmer [1984]. Local search procedures are reviewed by Scales [1985]. As examples of two-phase methods we mention:

–*Pure Random Search* (Rinnooy Kan & Timmer [1984,1987a]);

–*Controlled Random Search* (Price [1978]);

–*Multistart* (Rinnooy Kan & Timmer [1984,1987a]);

–*Clustering methods* (Törn [1978], Rinnooy Kan & Timmer [1987a], De Biase & Frontini [1978], Gomulka [1978b]);

–*Multi Level Single Linkage* (Rinnooy Kan & Timmer [1984,1987a,1987b]).

Methods based on simulated annealing apply a probabilistic mechanism that enables to search procedures to escape from local minima. This approach is extensively discussed in the remainder of this paper.

This paper is organized as follows: In Sections 2 and 3 a simulated annealing method, which is known from discrete minimization, is transformed into a global minimization method for real-valued functions; Section 2 contains the mathematical model of the algorithm and the proof of the asymptotic convergence to a global minimum; Section 3 describes a detailled implementation of the algorithm, which fits into the theoretical framework of Section 2. In Section 4 the simulated annealing algorithm is compared to some well-known methods by using a set of test functions from the literature. Section 5 concludes the paper with some inferences and remarks.

2. SIMULATED ANNEALING: THEORY

2.1 *Origin of the Algorithm*

Simulated annealing is a stochastic method to avoid getting stuck in local, non global, minima, when searching for global minima. This is done by accepting, in addition to points corresponding to a decrease in function value, points corresponding to an increase in function value. The latter is done in a limited way by means of a stochastic acceptance criterion. In the course of the minimization process, the probability of accepting deteriorations descends slowly towards zero. These 'deteriorations' make it possible to 'climb' out of local minima and explore S entirely. Eventually, this procedure will lead to a (near) global minimum.

Simulated annealing originates from the analogy between the physical annealing process and the problem of finding (near) minimal solutions for discrete minimization problems. The physical annealing process is known in condensed matter physics as a thermal process for obtaining low energy states of a solid in a heat bath. As far back as 1953, Metropolis, Rosenbluth, Rosenbluth, Teller & Teller [1953] proposed a method for computing the equilibrium distribution of a set of particles in a heat bath using a computer simulation method. In this method, a given state with energy E_1 is compared to a state that is obtained by moving one of the particles of the state to another location by a small displacement. This new state, with energy E_2, is accepted if $E_2 - E_1 \leq 0$, i.e. if the move brings the system in a state of lower energy. If $E_2 - E_1 \geq 0$, the new state is not rejected, buto exp($-(E_2 - E_1)$/k*T) where k is the

Boltzmann constant and T the temperature of the heat bath. So a move to a state of higher energy, a 'deterioration', is accepted in a limited way. By repeating this process for a large enough number of moves, Metropolis, Rosenbluth, Rosenbluth, Teller & Teller assumed that the canonical distribution, known as the Boltzmann distribution, is approached at a given temperature.

The first authors that linked the simulated annealing of solids with combinatorial minimization were Kirkpatrick, Gelatt & Vecchi [1983]. They replaced the energy by a cost function, and the states of a physical system by solutions of a combinatorial minimization problem. The perturbation of the particles in the physical system then becomes equivalent to a trial in the combinatorial minimization problem. The minimization is done by firstly 'melting' the solution space at a high effective temperature, (temperature now simply being a control parameter), and then lowering slowly the temperature until the system is 'frozen' into a stable solution.

This algorithm, when applied to combinatorial minimization problems, can be proved to converge to a global minimum with a guarantee in the probabilistic sense. It is generally applicable because no specific information about the cost function or solution space is needed a priori. Furthermore it is easy to implement and has a good performance, altough some applications may require large computational efforts. For an overview of the applications of the simulated annealing algorithm to combinatorial optimization problems the reader is referred to Aarts & Korst [1988] and Van Laarhoven & Aarts [1987].

Because of the success of the simulated annealing algorithm in combinatorial minimization problems, we have been investigating its potential for solving continuous minimization problems.

2.2 *Simulated Annealing for Continuous Minimization*

Application of simulated annealing to the minimization of a continuous valued function has been addressed by a number of authors. The proposed approaches can be divided into the following two classes.

– In the first class, applications of the algorithm are described that follow closely the original physical approach introduced by Kirkpatrick, Gelatt & Vecchi [1983]. For example

Vanderbilt & Louie [1983] use a covariance matrix for controlling the transition probability. This matrix should in some way reflect the topology of the search space and the acceptance criterion. Khachaturyan [1986] presents a method that is closely related to a physical system as described by Metropolis, Rosenbluth, Rosenbluth, Teller & Teller [1953]. Bohachevsky, Johnson & Stein [1986] present a simple and easy to implement method in which the length of a generation step was a constant. Kushner [1987] describes an appropriate method for cost functions, for which the values only can be sampled via a Monte Carlo method. If no sampling noise exists, this method is a regular version of the simulated annealing algorithm.

– In the second class of approaches, the annealing process is described by Langevin equations, and proven to converge to the set of global minima. A global minimum is then found by solving stochastic differential equations. Aluffi-Pentini, Parisi & Zirilli [1985] propose to compute global minima by following the paths of a system of stochastic differential equations. They use a time-dependent function for the acceptance criterium which tends to zero in a suitable way. Their method finds a global minimum for all test functions that were used. The papers of Geman & Hwang [1986] and Chiang, Hwang & Sheu [1987] consider the same concept. A continuous path seeking a global minimum will in general be forced to 'climb hills', with a standard n-dimensional Brownian motion, as well as follow down-hill gradients. The Brownian motion is controlled by a time dependent factor, tending to zero as time goes to infinity. The convergence proof given by Geman & Hwang is based on Langevin equations. They make use of an inhomogeneous Markov chain and the probability distribution function they use is the same as probability distribution function used in Theorem 2.2 (see below).

The simulated annealing algorithm, as described in this paper, fits in neither of these two classes. Our algorithm is a transformation of the simulated annealing method for discrete minimization to one for continuous minimization. The definition and the convergence proof of the algorithm are analogous to the ones given for the algorithm when applied to discrete optimization problems, and are based on the equilibrium distribution of Markov chains (see Aarts & Korst [1988] and Van Laarhoven & Aarts [1987]).

2.3 *The Mathematical Model of the Algorithm*

We now present a mathematical model of the simulated annealing algorithm for continuous optimization based on the ergodic theory of Markov chains.

Definition 2.1: X(k) is a *random variable* denoting the outcome of the k-th trial by simulated annealing. The outcome of a trial is a point $x \in S$ and depends only on the outcome of the previous trial. A *Markov chain* in the simulated annealing algorithm then is a sequence of trials. □

Definition 2.2: $g_{xy}(c)$ is the *generation probability distribution function*, i.e. the probability distribution function for generating a point y from point x at a fixed value of the control parameter $c \in \mathbb{R}^+$. □

Definition 2.3: $A_{xy}(c)$ is the *acceptance probability*, i.e. the probability for accepting point y if x is the current point in a Markov chain and y is generated as a possible new point. □

Definition 2.4: The *transition probability* of transforming $x \in S$ into a point $y \in T \subset S$ is the probability of generating and accepting a point in T if $x \notin T$. Thus if x is the current point of the Markov chain then the probability that an element out of T is the next point of the Markov chain is

$$P(T \mid x;c) = \begin{cases} \int_{y \in T} p_{xy}(c)dy & \text{for } x \notin T \\ \int_{y \in T} p_{xy}(c)dy + (1 - \int_{y \in S} p_{xy}(c)dy) & \text{for } x \in T \end{cases} \qquad (2.1)$$

where

$$p_{xy}(c) = g_{xy} \cdot A_{xy}(c) \qquad (2.2)$$

and

$$P(T \mid x;c) = \Pr\{ X(l) \in T \mid X(l-1) = x;c \}. \ \square \qquad (2.3)$$

Note that $p_{xy}(c)$ is no proper probability distribution function, for

$$\int_{y \in S} p_{xy}(c)dy \neq 1. \qquad (2.4)$$

Therefore hereafter p_{xy} is called the *quasi probability distribution function.*

In this paper, the acceptance probability $A_{xy}(c)$ is chosen equal to the Metropolis criterion, i.e.

$$A_{xy}(c) = \min\{ 1, \exp(-(f(y) - f(x))/c) \}. \qquad (2.5)$$

2.4 *Asymptotic Convergence of the Algorithm*

In this section it will be shown that the procedure given above converges asymptotically to a point x, where $x \in B_f(\varepsilon)$ (definition 1.2), i.e we prove that:

$$\forall_{\varepsilon>0}: \lim_{c \downarrow 0} \lim_{k \to \infty} \Pr\{ X(k) \in B_f(\varepsilon) \mid c \} \geq 1 - \varepsilon \qquad (2.6)$$

for all starting points X(0).

The proof is based on the convergence proof of the simulated annealing algorithm when applied to the discrete minimization problem (see Aarts & Korst [1988] and Van Laarhoven & Aarts [1987]).

Essential to the convergence proof of the algorithm is the fact that under certain conditions there exists a unique stationary probability distribution function of a homogeneous Markov chain.

Definition 2.5: A probability distribution function r(x,c) is *stationary* if

$$\forall_{x \in S}: r(x,c) = \int_{y \in S} r(y,c) p_{yx}(c) dy + r(x,c)\left(1 - \int_{y \in S} p_{xy}(c) dy\right) \qquad (2.7)$$

and

$$\int_{x \in S} r(y,c) dy = 1. \ \square \qquad (2.8)$$

Definition 2.6: The probability to transform a point $x \in S$ into a point $y \in T \subset S$ in k trials is

$$P^{(k)}(T \mid x;c) = \begin{cases} \displaystyle\int_{y \in T} p_{xy}^{(k)}(c) dy & \text{for } x \notin T \\ \displaystyle\int_{y \in T} p_{xy}^{(k)}(c) dy + \left(1 - \int_{y \in S} p_{xy}(c) dy\right)^k & \text{for } x \in T \end{cases} \qquad (2.9)$$

where

$$p_{xy}^{(k)}(c) = \int_{z \in S} p_{xz}^{(k-1)}(c) p_{zy}(c) dz + p_{xy}^{(k-1)}(c)\left(1 - \int_{z \in S} p_{yz}(c) dz\right) + \left(1 - \int_{z \in S} p_{xz}(c) dz\right)^{k-1} p_{xy}(c) \qquad (2.10)$$

i.e. $p_{xy}^{(k)}(c)$ is the quasi probability distribution function of transforming x into y in k trials and hence $p_{xy}^{(k)}(c)$ is equal to the summation of three terms:

(i) the first term is the quasi probability distribution function of transforming x into z in (k–1) trials and from z to y in the next trial integrated over all z;

(ii) the second term is the quasi probability distribution function of transforming x into y in

(k–1) trials and then reject the k–th trial;

(iii) the third term is the quasi probability distribution function of transforming x into y in one trial after (k–1) rejected trials from x. □

Lemma 2.1: *For the Markov chain, given by definition 2.1, S is the only ergodic set and S has no cyclically moving subsets (Doob* [1953]*), if*

$$\forall_{x_o \in S} \forall_{T \subset S}: \; m(T) > 0 \Rightarrow \int_{y \in T} g_{x_o y}(c)\, dy > 0, \tag{2.11}$$

where m(T) is the Lebesgue measure of the set T (Weir [1973]).

Proof: For each $x_o \in S$ we have

$$\forall_{T \subset S}: \; m(T) < m(S) \Rightarrow$$
$$1 = P^{(k)}(S|x_o;c) = P^{(k)}(T|x_o;c) + P^{(k)}(S \setminus T|x_o;c). \tag{2.12}$$

Condition *(2.11)* assures that $P^{(k)}(S \setminus T|x_o;c) > 0$, and hence

$$\forall_{x_o} \forall_{T \subset S}: \; P^{(k)}(T|x_o;c) < 1. \tag{2.13}$$

So *S* is the only *consequent* of x_o and *S* is the only *invariant set* (Doob [1953]).

Now *S* has to be decomposed into disjoint invariant sets and a transient set (Doob [1953]), but *S* is the only invariant set and the complement of *S* is empty and therefore *S* is the only ergodic set.

Furthermore *S* cannot be divided into t disjunct sets $T_1,\ldots,T_t$ such that

$$\forall_{x_o \in T_i}: \; P(T_{i+1}|x_o;c) = 1, \quad 1 \le i \le t, \tag{2.14}$$

(where T_{t+1} is interpreted as T_1) (Doob [1953]), because of *(2.11)*. Hence *S* has no cyclically moving subsets. This completes the proof of lemma 2.7. □

Theorem 2.1: *(A continuous analogon of Feller's theorem (Feller* [1957]*, pp. 356-357))*
The stationary probability distribution function of a homogeneous Markov chain as in definition 2.1 exists if S is the only ergodic set and has no cyclically moving subsets. Moreover this probability distribution function q is defined as

$$q(x,c) = \lim_{k \to \infty} p_{yx}^{(k)}(c) \tag{2.15}$$

and is uniquely determined by the following equations:

(i) $\forall_{x \in S}: \; q(x,c) > 0;$ *(2.16)*

(ii) $\int_{x\in S} q(x,c)dx = 1;$ *(2.17)*

(iii) $\forall_{x\in S}$: $q(x,c) = \int_{y\in S} q(y,c)p_{yx}(c)dy + q(x,c)(1-\int_{y\in S} p_{xy}(c)dy).$ *(2.18)*

Reformulation: If the above holds, then for an arbitrary initial probability distribution function (u_x), we obtain as $k \to \infty$:

$$u_x^{(k)} = \int_{y\in S} u_y p_{yx}^{(k)}(c)dy + u_x(1-\int_{y\in S} p_{xy}(c)dy)^k \to q(x,c). \tag{2.19}$$

Proof: Note that for all $n > 0$ we have

$$P^{(n)}(S|x,c) = \int_{y\in S} p_{xy}^{(n)}(c)dy + (1 - \int_{y\in S} p_{xy}(c)dy)^n = 1, \tag{2.20}$$

which implies that

$$\int_{y\in S} p_{xy}^{(n)}(c)dy \le 1. \tag{2.21}$$

Since S is the only ergodic set and S has no cyclically moving subsets, $\lim_{n\to\infty} p_{xy}^{(n)}(c)$ exists as an ordinary limit and is independent of x (Doob [1953]). Hence we obtain

$$\int_{y\in S} q(y,c)dy = \int_{y\in S} \lim_{n\to\infty} p_{xy}^{(n)}(c)dy = \lim_{n\to\infty} \int_{y\in S} p_{xy}^{(n)}(c)dy \le 1. \tag{2.22}$$

Furthermore, we have

$$p_{xy}^{(m+1)}(c) = \int_{z\in S} p_{xz}^{(m)}(c)p_{zy}(c)dz + p_{xy}^{(m)}(c)(1 - \int_{z\in S} p_{yz}(c)dz) + (1-\int_{z\in S} p_{xz}(c)dz)^m p_{xy}(c). \tag{2.23}$$

Now, as $m \to \infty$ we obtain

$$\begin{aligned} q(y,c) &= \lim_{m\to\infty} p_{xy}^{(m+1)}(c) \\ &= \lim_{m\to\infty} \int_{z\in S} p_{xz}^{(m)}(c)p_{zy}(c)dz + \lim_{m\to\infty} p_{xy}^{(m)}(c)(1-\int_{z\in S} p_{yz}(c)dz) + \lim_{m\to\infty} (1-\int_{z\in S} p_{xz}(c)dz)^m p_{xy}(c) \\ &= \int_{z\in S} q(z,c)p_{zy}(c)dz + q(y,c)(1 - \int_{z\in S} p_{yz}(c)dz) + 0. \end{aligned} \tag{2.24}$$

Note that $\int_{y\in S} q(y,c)dy \le 1$.

Next, define

$$r(y,c) = \frac{q(y,c)}{\int_{z\in S} q(z,c)dz}, \tag{2.25}$$

then

(i) $r(y,c) > 0$, because S is the only ergodic set; *(2.26)*

$$\text{(ii)}\quad \int_{y\in S} r(y,c)dy = \frac{\int_{y\in S} q(y,c)dy}{\int_{z\in S} q(z,c)dz} = 1; \tag{2.27}$$

$$\text{(iii)}\quad r(y,c) = \frac{q(y,c)}{\int_{z\in S} q(z,c)dz} = \frac{\int_{x\in S} q(x,c)p_{xy}(c)dx + q(y,c)(1-\int_{x\in S} p_{yx}(c)dx)}{\int_{z\in S} q(z,c)dz}$$

$$= \int_{x\in S} r(x,c)p_{xy}(c)dx + r(y,c)(1-\int_{x\in S} p_{yx}(c)dx) \tag{2.28}$$

Hence, at least one stationary probability distribution function exists.

Lemma 2.2: *Let r(z,c) be any distribution satisfying Definition 2.5. Then we have*

$$r(z,c) = \int_{x\in S} r(x,c)p_{xz}^{(k)}(c)dx + r(z,c)(1-\int_{x\in S} p_{zy}(c)dx)^k. \tag{2.29}$$

Proof: By induction.

For $k = 1$ *(2.29)* holds. Now assume *(2.29)* is correct for k. Then multipling *(2.29)* by $p_{zy}(c)$ and integrating over $z \in S$ yields

$$\int_{z\in S} r(z,c)p_{zy}(c)dz = \int_{z\in S}\int_{x\in S} r(x,c)p_{xz}^{(k)}(c)p_{zy}(c)dx\,dz + \int_{z\in S} r(z,c)p_{zy}(c)(1-\int_{x\in S} p_{zy}(c)dx)^k dz. \tag{2.30}$$

Next, using Definition 2.5 and *(2.10)* we obtain

$$r(y,c) - r(y,c)(1-\int_{x\in S} p_{yx}(c)dx)$$

$$= \int_{x\in S} r(x,c)\left\{p_{xy}^{(k+1)}(c) - p_{xy}^{(k)}(c)(1-\int_{z\in S} p_{yz}(c)dz) - (1-\int_{z\in S} p_{xz}(c)dz)^k p_{xy}(c)\right\}dx + \int_{z\in S}\{r(z,c)p_{zy}(c)(1-\int_{x\in S} p_{zx}(c)dx)^k\}dz. \tag{2.31}$$

So, using *(2.29)* for k:

$$r(y,c)$$

$$= \int_{x\in S} r(x,c)p_{xy}^{(k+1)}(c)dx - (1-\int_{z\in S} p_{yz}(c)dz)\left\{r(y,c) - r(y,c)(1-\int_{x\in S} p_{yx}(c)dx)^k\right\} + r(y,c)(1-\int_{x\in S} p_{yx}(c)dx)$$

$$= \int_{x\in S} r(x,c)p_{xy}^{(k+1)}(c)dx + r(y,c)(1-\int_{z\in S} p_{yz}(c)dz)^{k+1}. \tag{2.32}$$

Thus *(2.29)* is correct for k+1. This completes the proof of Lemma 2.2. □

We now complete the proof of Theorem 2.1. As $k \to \infty$, *(2.29)* transforms into

$$\lim_{k\to\infty} r(z,c) = \lim_{k\to\infty} \left\{ \int_{x\in S} r(x,c)p_{xz}^{(k)}(c)dx + r(z,c)(1- \int_{x\in S} p_{zy}(c)dx)^k \right\}$$

$$= \int_{x\in S} r(x,c)dx + 0 = q(z,c)(\int_{x\in S} r(x,c)dx) = q(z,c). \qquad (2.33)$$

Hence any distribution satisfying Definition 2.5 is equal to the probability distribution function q. So q is unique. This completes the proof of Theorem 2.1. □

Theorem 2.2: *Let $p_{xy}(c)$ be given by Definition 2.4 and let S be the only ergodic set not having any cyclically moving subsets for the Markov chain induced by P(T|x;c) (Definition 2.4). Furthermore, let the following conditions be satisfied:*

(i) $\forall_{x,y\in S}$: $g_{xy}(c) = g_{yx}(c)$; *(2.34)*

(ii) $g_{xy}(c)$ *is not depending on c (and can therefore be written as* g_{xy}*).* *(2.35)*

Then the stationary probability distribution function is given by:

$$q(x,c) = \frac{exp(-(f(x)-f_{min})/c)}{\int_{y\in S} exp(-(f(y)-f_{min})/c)dy}, \qquad (2.36)$$

where f_{min} is the minimal function value, i.e $f_{min} = f(x_{min})$ for all x_{min} (see (1.1)).

Proof: If q(x,c) satisfies *(2.16)*, *(2.17)* and *(2.18)*, it is the unique stationary probability distribution function (Theorem 2.1):

(i) $\forall_{x\in S}$: $q(x,c) = \dfrac{\int_{x\in S} \exp(-(f(x)-f_{min})/c)dx}{\int_{y\in S} \exp(-(f(y)-f_{min})/c)dy} > 0;$ *(2.37)*

(ii) $\int_{x\in S} q(x,c)dx = \dfrac{\int_{x\in S} \exp(-(f(x)-f_{min})/c)dx}{\int_{y\in S} \exp(-(f(y)-f_{min})/c)dy} = 1;$ *(2.38)*

(iii) Let N(c), $S^-(x)$ and $S^+(x)$ be defined as follows:

$$N(c) = \int_{y\in S} \exp(-(f(y)-f_{min})/c)dy; \qquad (2.39)$$

$$S^-(x) = \{ y \in S \mid f(y) \le f(x) \}; \qquad (2.40)$$

$$S^+(x) = \{ y \in S \mid f(y) > f(x) \}. \qquad (2.41)$$

Then

$$\int_{y\in S} q(y,c)p_{yx}(c)dy$$

$$= \int_{y\in S^-(x)} \frac{1}{N(c)} \exp(-(f(y)-f_{min})/c) g_{yx} \min\{1,\exp(-(f(x)-f(y))/c)\}dy$$
$$+ \int_{y\in S^+(x)} \frac{1}{N(c)} \exp(-(f(y)-f_{min})/c) g_{yx} \min\{1,\exp(-(f(x)-f(y))/c)\}dy$$
$$= \int_{y\in S^-(x)} \frac{1}{N(c)} \exp(-(f(x)-f_{min})/c) g_{xy}dy + \int_{y\in S^+(x)} \frac{1}{N(c)} \exp(-(f(y)-f_{min})/c) g_{xy}dy$$
$$= q(x,c) \int_{y\in S^-(x)} g_{xy}dy + \int_{y\in S^+(x)} q(y,c) g_{xy}dy, \qquad (2.42)$$

and

$$q(x,c)(1- \int_{y\in S} p_{xy}(c)dy)$$
$$= q(x,c)\left\{1- \int_{y\in S^-(x)} g_{xy} \min\{1,\exp(-(f(y)-f(x))/c)\}dy - \int_{y\in S^+(x)} g_{xy} \min\{1,\exp(-(f(y)-f(x))/c)\}dy\right\}$$
$$= q(x,c) - q(x,c) \int_{y\in S^-(x)} g_{xy}dy$$
$$- \int_{y\in S^+(x)} \frac{1}{N(c)} \exp(-(f(x)-f_{min})/c) g_{xy} \min\{1,\exp(-(f(y)-f(x))/c)\}dy$$
$$= q(x,c) - q(x,c) \int_{y\in S^-(x)} g_{xy}dy - \int_{y\in S^+(x)} q(y,c) g_{xy}dy. \qquad (2.43)$$

Combining *(2.42)* and *(2.43)* yields

$$\forall_{x\in S}: \int_{y\in S} p_{yx}(c) g_{yx}dy + q(x,c)(1- \int_{y\in S} p_{xy}(c)dy) = q(x,c). \qquad (2.44)$$

This completes the proof of Theorem 2.2. □

We now proof that the simulated annealing algorithm converges to a near minimal solution if the stationary probability ditribution fraction is given by *(2.36)*.

Theorem 2.3:

$$\forall_{\varepsilon>0}: \lim_{c\downarrow 0} \int_{y\in B_f(\varepsilon)} q(y,c)dy > 1-\varepsilon \qquad (2.45)$$

if the number of local minima is finite and f is uniformly continuous.

Proof: Since the number of local minima is finite we have:

$$\exists_{\varepsilon_1>0}: |f(x_{loc}) - f_{min}| > \varepsilon_1, \qquad (2.46)$$

$$\exists_{\varepsilon_2>0} \forall_{x_{min}}: \| x_{loc} - x_{min} \| > \varepsilon_2, \qquad (2.47)$$

where $f_{min} = f(x_{min})$ for all x_{min} (see *(1.1)*) and x_{loc} is a local, non global, minimum.

Now choose ε, such that

$$0 < \varepsilon < \min\{\tfrac{1}{4}\varepsilon_1, \tfrac{1}{4}\varepsilon_2\}. \qquad (2.48)$$

(If all minima are global then ε should be chosen such that $\exists_{x\in S}: f(x) - f_{min} > \varepsilon$).

Because f is uniformly continuous we have:

$$\exists_{\delta_1>0}\forall_{x,y\in S}:\ \| x-y \| \le \delta_1 \Rightarrow |f(x)-f(y)| < \tfrac{1}{2}\varepsilon. \tag{2.49}$$

Let δ be chosen as follows:

$$\delta = \min\{\ \tfrac{1}{2}\delta_1, \varepsilon\ \}. \tag{2.50}$$

Then we have

$$\forall_{y\in B_x(\delta)}:\ f(y) - f_{min} < \tfrac{1}{2}\varepsilon, \tag{2.51}$$

where $B_x(\delta)$ is given by Definition 1.1.

Now take a point

$$x_o \in S\backslash B_x(\delta),$$

with (2.52)

$$f(x_o) - f_{min} = \varepsilon.$$

(This is possible, because f is continuous.)

Then

$$\lim_{c\downarrow 0} q(x_o,c) = \lim_{c\downarrow 0} \frac{\exp(-(f(x_o)-f_{min})/c)}{\int_{y\in S}\exp(-(f(y)-f_{min})/c)dy}$$

$$= \lim_{c\downarrow 0} \frac{\exp(-\varepsilon/c)}{\int_{y\in S}\exp(-(f(y)-f_{min})/c)dy} = \lim_{c\downarrow 0} \frac{1}{\int_{y\in S}\exp((\varepsilon-(f(y)-f_{min}))/c)dy}$$

$$= \lim_{c\downarrow 0} \frac{1}{\int_{y\in S\backslash B_x(\delta)}\exp((\varepsilon-(f(y)-f_{min}))/c)dy + \int_{y\in B_x(\delta)}\exp((\varepsilon-(f(y)-f_{min}))/c)dy}$$

$$\le \lim_{c\downarrow 0} \frac{1}{\int_{y\in B_x(\delta)}\exp((\varepsilon-(f(y)-f_{min}))/c)dy} \le \frac{1}{\lim_{c\downarrow 0}\int_{y\in B_x(\delta)}\exp((\varepsilon-\frac{1}{2}\varepsilon)/c)dy}$$

$$= \frac{1}{\lim_{c\downarrow 0}\exp(\tfrac{1}{2}\varepsilon/c)\ m(B_x(\delta))} \to 0. \tag{2.53}$$

So, with $m(S)$ as before the Lebesgue measure of S,

$$\exists_{c_o>0}\ \forall_{c<c_o}:\ q(x_o,c) < \frac{\varepsilon}{m(S)}. \tag{2.54}$$

Hence

$$\forall_{c<c_o}\ \forall_{x\in S^+(x_o)}:\ q(x,c) \le q(x_o,c) < \frac{\varepsilon}{m(S)}, \tag{2.55}$$

and

$$\forall_{c<c_o} \forall_{x\in S^-(x_o)}: f(x) - f_{min} < \varepsilon, \tag{2.56}$$

where $S^-(x_o)$ and $S^+(x_o)$ as in *(2.40)* and *(2.41)*.

Now for all $c < c_o$ we have

$$\begin{aligned} 1 &= \int_{y\in S} q(y,c)dy = \int_{y\in S^-(x_o)} q(y,c)dy + \int_{y\in S^+(x_o)} q(y,c)dy \\ &< \int_{y\in B_f(\varepsilon)} q(y,c)dy + \int_{y\in S^+(x_o)} \frac{\varepsilon}{m(S)}dy \le \int_{y\in B_f(\varepsilon)} q(y,c)dy + \varepsilon. \end{aligned} \tag{2.57}$$

Note that $B_f(\varepsilon) = S^-(x_o)$ and that there is no local minimum in $B_f(\varepsilon)$ because of *(2.47)* and *(2.48)*.

Hence we have

$$\lim_{c\downarrow 0} \int_{y\in B_f(\varepsilon)} q(y,c)dy > 1 - \varepsilon, \tag{2.58}$$

which completes the proof of Theorem 2.3. □

In conclusion, we have shown in this section that the simulated annealing algorithm for continuous minimization, modeled as a Markov chain with the following transition probability (Definition 2.4):

$$P(T|x;c) = \begin{cases} \int_{y\in T} p_{xy}(c)dy & \text{for } x \notin T \\ \int_{y\in T} p_{xy}(c)dy + (1 - \int_{y\in S} p_{xy}(c)dy) & \text{for } x \in T \end{cases}$$

where

$$p_{xy}(c) = g_{xy} \cdot A_{xy}(c)$$

and

$$P(T|x;c) = \Pr\{ X(l) \in T \mid X(l-1) = x;c\},$$

converges to the set of minimal points of a function $f: S \to \mathbb{R}$.

Thus

$$\lim_{c\downarrow 0} \lim_{l\to\infty} \Pr\{ X(l) \in B_f(\varepsilon) \mid c \} > 1 - \varepsilon \tag{2.59}$$

if the following conditions are met:

(i) $f: S \to \mathbb{R}$ is uniform continuous;

(ii) S is a bounded subset of $\mathbb{R}^n$ and all the minima are interior points of S;

(iii) the number of minima is finite;

(iv) the acceptance criteion $A_{xy}(c)$ is *((2.5))*:

$$A_{xy}(c) = \min\{1, \exp(-(f(y)-f(x))/c)\};$$

(v) the generation probability distribution function $g_{xy}(c)$ induces:

$$\forall_{x_o \in S} \forall_{T \subset S}: m(T) > 0 \Rightarrow \int_{y \in S} g_{x_o y}(c)dy > 0 \ ((2.11));$$

$$g_{xy}(c) = g_{yx}(c) \ ((2.34));$$

$g_{xy}(c)$ does not depend on c *((2.35))*.

Finally, we mention that these conditions are sufficient but not necessary.

3. SIMULATED ANNEALING: PRACTICE

3.1 *Cooling schedule*

The simulated annealing algorithm described in the previous section can be viewed as an infinite number of homogeneous Markov chains of infinite length. This is due to the two limits of *(2.59)*, i.e. $\lim_{k \to \infty}$ and $\lim_{c \downarrow 0}$. Clearly an implementation of the algorithm according to this prescription is impractible. In this section a more explicit and practicable approach is given, which is similar to the approach given by Aarts & Van Laarhoven [1985] for discrete minimization. This approach realizes a finite-time implementation of the simulated annealing algorithm by generating homogeneous Markov chains of finite length at a finite sequence of (descending) values of the control parameter. To achieve this, a set of parameters must be specified that governs the convergence of the algorithm. This set of parameters constitutes a so-called cooling schedule.

Definition 3.1: A cooling schedule specifies:

– An initial value of the control parameter c_o;

– A decrement function for decreasing the value of the control parameter;

– A final value of the control parameter, i.e. a stop criterion;

– A finite length, L, of each Markov chain. □

The above leads to the following simulated annealing algorithm in pseudo– PASCAL:

PROCEDURE SIMULATED ANNEALING; *(3.1)*

```
begin "initialize (c,x)";
      stopcriterion := false;
      while stopcriterion = false do
      begin for i :=1 to L do
            begin "generate y from x";
                  if Δf_xy ≤ 0 then accept
                              else if exp(–Δf_xy/c) > random [0,1) then accept;
                  if accept then x:= y
            end;
            "lower c"
      end
end.
```

Below, we elaborate these parameters in more detail. We mention beforehand that the guarantee that this finite-time implementation of the simulated annealing algorithm will eventually succeed in finding a global minimum no longer holds; this is because of the finite length and finite number of Markov chains. However, the probability of finding a global minimum is still large and can be raised by using longer Markov chains and/or a more careful decrease of the control parameter. This will however effect the efficiency and therefore a compromise has to be made between reliability and efficiency.

We now briefly summarize the cooling schedule as introduced by Aarts & Van Laarhoven. For a detailed description see Aarts and Van Laarhoven [1985].

– initial value of the control parameter

The basic assumption underlying the calculation of the initial value of the control parameter is that c_o should be sufficiently large, such that approximately all transitions are accepted at this value. This can be achieved by generating a number of trials, say m_o, and requiring that the *initial acceptance ratio* $\chi_o = \chi(c_o)$ is close to 1 ($\chi(c)$ is defined as the ratio between the number of accepted transitions and the number of proposed transitions). The initial value of c_o is then obtained from the following expression:

$$c_o = \overline{\Delta f^+} \left[\ln \frac{m_2}{m_2 \chi \quad (1-\chi) m_1} \right]^{-1} \qquad (3.2)$$

where m_1 and m_2 denote the number of trials ($m_1 + m_2 = m_o$) with $\Delta f_{xy} \leq 0$ and $\Delta f_{xy} > 0$, respectively, and $\overline{\Delta f^+}$ the average value of those Δf_{xy}-values for which $\Delta f_{xy} > 0$ ($\Delta f_{xy} = f(x) - f(y)$).

–decrement of the control parameter

The new value of c, say c', is calculated from the following expression:

$$c' = c \left[1 + \frac{c \ln(1 + \delta)}{3 \sigma(c)} \right]^{-1}, \qquad (3.3)$$

where $\sigma(c)$ denotes the standard deviation of the values of the cost function of the points in the Markov chain at c, and δ is a small positive real number. The constant δ is called the *distance parameter* and determines the speed of the decrement of the control parameter.

–final value of the control parameter

The stop criterion is based on the idea that the average function value $\bar{f}$ of a Markov chain is an increasing function of c, i.e. if c is lowered then $\bar{f}$ will lower too, such that $\bar{f}(c)$ converges to $f(x_{min})$ as $c \downarrow 0$.

The algorithm is terminated if:

$$\left| \frac{d\bar{f}_s(c)}{dc} \frac{c}{\bar{f}(c_o)} \right| < \varepsilon_s, \qquad (3.4)$$

where $\bar{f}(c_o)$ is the mean value of the points found in the initial Markov chain, $\bar{f}_s(c)$ is the smoothed value of $\bar{f}$ over a number of chains in order to reduce the fluctuations of $\bar{f}(c)$ and ε_s is a small positive real number, called the *stop parameter.*

–length of the Markov chains

The length of the Markov chains is based on the assumption that they should be sufficiently large in order to enable the algorithm to exploire the neighbourhood of a given point in all directions. A straightforward choice therefore is given by the following relation

$$L = L_o \cdot n, \tag{3.5}$$

where n denotes the dimension of S and L_o a constant called the *standard length.* Note that this choice leads to a chain length which is constant for a given problem instance.

3.2 *Generation of Points*

There are several possibilities for generating new points from a given point. The only requirement is that the generation mechanism should satisfy *(2.11)*, *(2.34)* and *(2.35)*. We discuss two alternatives.

Altenative *A*: A uniform ditribution on S, i.e.

$$g_{xy}(c) = \frac{1}{m(S)}. \tag{3.6}$$

Clearly this alternative satisfies conditions *(2.11)*, *(2.34)* and *(2.35)*. An obvious disadvantage of this choise is that no structural information about function values is used. This disadventage can be circumvented by introducing an additional mechanism that uses descent directions. For each new generation there are two possibilities, a point is drawn from a uniform distribution over S or a step is made into a descent direction from the current point, i.e.

Alternative *B*:

$$g_{xy}(c) = \begin{cases} \frac{1}{m(S)} & \text{if } w \le t \\ LS(x) & \text{if } w > t \end{cases} \tag{3.7}$$

where t is a fixed number in the interval [0,1) and w a random number drawn from U[0,1). LS(x) is a Local Search procedure that generates a point y in a descent direction of x, thus with $f(y) \le f(x)$ (y is not necessarily a local minimum). This generation mechanism seems more efficient, because of its local search steps. There is one drawback to this generation mechanism: $g_{xy(c)} \neq g_{yx(c)}$ and thus *(2.34)* is no longer satisfied. It can be shown however that this method still converges to $B_f(\varepsilon)$ (Definition 1.2).

Theorem 3.1: *Let P denote the transition probability associated with the simulated annealing algorithm (Definition 2.4), and let the random variables X(k) and Y(k) be defined* as the outcomes of the trials in the simulated annealing algorithm using *alternative A and alternative B, respectively. Then*

$$\forall_{\varepsilon>0}: \lim_{c\downarrow 0} \lim_{k\to\infty} Pr\{ Y(k) \in B_f(\varepsilon) | c\} \ge \lim_{c\downarrow 0} \lim_{k\to\infty} Pr\{ X(k) \in B_f(\varepsilon) | c\} > 1 - \varepsilon. \ \square \tag{3.8}$$

Proof:

$$\Pr\{Y(k) \in B_f(\varepsilon) \,|\, Y(k-1) \in B_f(\varepsilon);c\} = t \Pr\{X(k) \in B_f(\varepsilon) \,|\, X(k-1) \in B_f(\varepsilon);c\}$$
$$+ (1-t) \Pr\{LS(Y(k-1)) \in B_f(\varepsilon) \,|\, Y(k-1) \in B_f(\varepsilon);c\}$$
$$= t \Pr\{X(k) \in B_f(\varepsilon) \,|\, X(k-1) \in B_f(\varepsilon);c\} + (1-t); \tag{3.9}$$

$$\Pr\{Y(k) \in B_f(\varepsilon) \,|\, Y(k-1) \notin B_f(\varepsilon);c\} = t \Pr\{X(k) \in B_f(\varepsilon) \,|\, X(k-1) \notin B_f(\varepsilon);c\}$$
$$+ (1-t) \Pr\{LS(Y(k-1)) \in B_f(\varepsilon) \,|\, Y(k-1) \notin B_f(\varepsilon);c\}$$
$$= t \frac{m(B_f(\varepsilon))}{m(S)} + (1-t) \Pr\{LS(Y(k-1)) \in B_f(\varepsilon) \,|\, Y(k-1) \notin B_f(\varepsilon);c\}; \tag{3.10}$$

$$\Pr\{Y(k) \notin B_f(\varepsilon) \,|\, Y(k-1) \in B_f(\varepsilon);c\} = t \Pr\{X(k) \notin B_f(\varepsilon) \,|\, X(k-1) \in B_f(\varepsilon);c\}$$
$$+ (1-t) \Pr\{LS(Y(k-1)) \notin B_f(\varepsilon) \,|\, Y(k-1) \in B_f(\varepsilon);c\}$$
$$= t (1 - \Pr\{X(k) \in B_f(\varepsilon) \,|\, X(k-1) \in B_f(\varepsilon);c\}), \tag{3.11}$$

$$\Pr\{Y(k) \notin B_f(\varepsilon) \,|\, Y(k-1) \notin B_f(\varepsilon);c\} = t \Pr\{X(k) \notin B_f(\varepsilon) \,|\, X(k-1) \notin B_f(\varepsilon);c\}$$
$$+ (1-t) \Pr\{LS(Y(k-1)) \notin B_f(\varepsilon) \,|\, Y(k-1) \notin B_f(\varepsilon);c\}$$
$$= t \left[1 - \frac{m(B_f(\varepsilon))}{m(S)}\right] + (1-t)(1 - \Pr\{LS(Y(k-1)) \in B_f(\varepsilon) \,|\, Y(k-1) \notin B_f(\varepsilon);c\}). \tag{3.12}$$

Consequently using

$$PB(c) = \Pr\{ X(k) \in B_f(\varepsilon) \mid X(k-1) \in B_f(\varepsilon);c\}, \tag{3.13}$$

$$PLS(c) = \Pr\{ LS(Y(k-1)) \in B_f(\varepsilon) \mid Y(k-1) \notin B_f(\varepsilon);c\}: \tag{3.14}$$

$$E(\text{waiting time of } Y(k) \text{ in } B_f(\varepsilon);c)$$
$$= \sum_{k=0}^{\infty} k \Pr\{ \forall_{0 \le i \le k}: Y(i) \in B_f(\varepsilon) \text{ and } Y(k) \notin B_f(\varepsilon) \mid Y(0) \in B_f(\varepsilon);c\}$$
$$= \sum_{k=0}^{\infty} k (t^*PB(c) + (1-t))^{k-1} (t (1 - PB(c))) = t (1 - PB(c)) \sum_{k=0}^{\infty} k (t^*PB(c) + (1-t))^{k-1}$$
$$= t (1 - PB(c)) \frac{1}{(t (1 - PB(c)))^2} = \frac{1}{t (1 - PB(c))} \tag{3.15}$$

Similarly:

$$E(\text{waiting time of } Y(k) \text{ in } S \backslash B_f(\varepsilon);c) = \frac{1}{t \frac{m(B_f(\varepsilon))}{m(S)} + (1-t)PLS(c)}, \tag{3.16}$$

$$E(\text{waiting time of } X(k) \text{ in } B_f(\varepsilon);c) = \frac{1}{(1 - PB(c))}, \tag{3.17}$$

$$E(\text{waiting time of } X(k) \text{ in } S \backslash B_f(\varepsilon);c) = \frac{m(S)}{m(B_f(\varepsilon))}. \tag{3.18}$$

From Theorem 2.2 we have

$$\forall_{\varepsilon>0}: \lim_{c \downarrow 0} \lim_{k \to \infty} \Pr\{ X(k) \in B_f(\varepsilon) \mid X(0) \in S;c\} > 1 - \varepsilon, \tag{3.19}$$

Furthermore we have

$$\lim_{k\to\infty} \Pr\{ X(k) \in B_f(\varepsilon) \mid X(0) \in S;c\}$$
$$= \frac{E(\text{waiting time of } X(K) \text{ in } B_f(\varepsilon);c)}{E(\text{waiting time of } X(k) \text{ in } B_f(\varepsilon);c)+E(\text{waiting time of } X(k) \text{ in } S\backslash B_f(\varepsilon);c)}$$
$$= \frac{\frac{1}{(1-PB(c))}}{\frac{1}{(1-PB(c))} + \frac{m(S)}{m(B_f(\varepsilon))}}. \qquad (3.20)$$

Hence

$$\forall_{\varepsilon>0}: \lim_{c\downarrow 0} \frac{\frac{1}{(1-PB(c))}}{\frac{1}{(1-PB(c))} + \frac{m(S)}{m(B_f(\varepsilon))}} > 1-\varepsilon. \qquad (3.21)$$

Finally, we obtain

$$\forall_{\varepsilon>0}: \lim_{c\downarrow 0} \lim_{k\to\infty} \Pr\{ Y(k) \in B_f(\varepsilon) \mid Y(0) \in S;c\}$$
$$= \frac{E(\text{waiting time of } Y(k) \text{ in } B_f(\varepsilon);c)}{E(\text{waiting time of } Y(k) \text{ in } B_f(\varepsilon);c)+E(\text{waiting time of } Y(k) \text{ in } S\backslash B_f(\varepsilon);c)}$$
$$= \frac{\frac{1}{t\,(1-PB(c))}}{t\left[\frac{m(S)}{m(B_f(\varepsilon))}\right] + (1-t)\,PLS(c) + \frac{1}{t\,(1-PB(c))}} \geq \frac{\frac{1}{t\,(1-PB(c))}}{t\left[\frac{m(S)}{m(B_f(\varepsilon))}\right] + \frac{1}{t\,(1-PB(c))}}$$
$$= \frac{\frac{1}{(1-PB(c))}}{\frac{1}{(1-PB(c))} + \frac{m(S)}{m(B_f(\varepsilon))}} > 1-\varepsilon. \qquad (3.22)$$

So

$$\forall_{\varepsilon>0}: \lim_{c\downarrow 0} \lim_{k\to\infty} \Pr\{ Y(k) \in B_f(\varepsilon) \mid Y(0) \in S;c \} > 1-\varepsilon. \qquad (3.23)$$

This completes the proof of the theorem. □

4. NUMERICAL RESULTS

The performance of the simulated annealing algorithm presented in Sections 2 and 3 is compared with the performance of a number of two-phase methods known from literature. There are three criteria that determine the performance of an algorithm: (i) the number of function evaluations, (ii) the running time and (iii) the quality of the final result. The latter criterion can be

quantified by the difference in the value of the cost function between the obtained minimum and the global minimum. Our performance analysis is carried out for a set of test functions known from the literature. The test functions are taken from Dixon & Szegö [1978b] and from Aluffi- - Pentini, Parisi & Zirilli [1985] (see Appendix A and Appendix B, respectively). Because all methods were implemented on different machines we used the standard unit of time as introduced by Dixon & Szegö [1978b]. One unit of time then is the running time needed for 1000 evaluations of the Shekel 5 function in the point (4,4,4,4) (see Appendix A).

It should be mentioned that a comparison between the various methods never will be entirely fair. The implementation of the methods is done by different persons on different machines and this always gives rise to some discrepancies in the results. Furthermore, different implementations emphasize different aspects, i.e: a compromise is made between efficiency and reliability, (where reliability refers to the probability of obtaining a (near) global minimum). Choosing for efficiency will affect the reliability and vice-versa.

4.1 *Implementation of the simulated annealing algorithm*

The simulated annealing algorithm is implemented on the Burroughs B7900 of the Eindhoven University of Technology using the programming language PASCAL. For the cooling schedule we used the following parameters (see Section 3.1): $\chi_0 = 0.9$, $\delta = 0.1$, $\varepsilon_s = 10^{-4}$ and $L_0 = 10$. Generation of points was done according to alternative *B* where $t = 0.75$.

The local search procedure is taken as a combination of steepest descent in the early stages of the optimization and Quasi-Newton in the latter stages. The Quasi-Newton procedure is implemented as the Broyden-Fletcher-Goldfarb-Shanno procedure as presented in Scales [1985]. This local search is done along one descent direction.

4.2 *Results*

In this section the computational results of the methods listed in table 4.1 are summarized.

table 4.1 Listing of different methods used in the comparison:

method	name	reference
A	Multistart	Rinnooy Kan & Timmer [1984]
B	Controlled Random Search	Price [1978]
C	Density clustering	Törn [1978]
D	Clustering with distribution function	De Biase & Frontini [1978]
E	Multi Level Single Linkage	Rinnooy Kan & Timmer [1987b]
F	Simulated Annealing	this paper
G	Simulated Annealing based on stochastic differential equations	Aluffi-Pentini, Parisi & Zirilli [1985]

In tables 4.2 and 4.3 the results are given of methods $A - F$ for the set of test functions proposed by Dixon & Szegö [1978b] (see Appendix A). For method *G* no results for this set of test functions are available. Table 4.2 gives the number of function evaluations and table 4.3 gives the running time in units of standard time.

table 4.2 Number of function evaluations:

function / method	GP	BR	H3	H6	S5	S7	S10
A	4400	1600	2500	6000	6500	9300	11000
B	2500	1800	2400	7600	3800	4900	4400
C	2499	1558	2584	3447	3649	3606	3874
D	378	597	732	807	620	788	1160
E	148	206	197	487	404	432*	564
F	563	505	1459	4648	365*	558	797

* the global minimum was not found in one of the four runs.

table 4.3 Running time in units of standard time:

function / method	GP	BR	H3	H6	S5	S7	S10
A	4.5	2	7	22	13	21	32
B	3	4	8	46	14	20	20
C	4	4	8	16	10	13	15
D	15	14	16	21	23	20	30
E	0.15	0.25	0.5	2	1	1*	2
F	0.9	0.9	5	20	0.8*	1.5	2.7

* the global minimum was not found in one of the four runs.

It should be mentioned that for most methods the number of function evaluations and the running time used for the generation of the initial random sample are not taken into account. This benefits some methods. The Multi Level Single Linkage method for instance uses 1000 function evaluations for the random sample, and consequently the correspoding running time is not negligible; whereas for simulated annealing the initialization uses $m_o = 10 \cdot n$ function evaluations (see *(3.2)*), where n is the dimension. This number is clearly less than in the Multi Level Single Linkage method.

Tables 4.2 and 4.3 shows that Multi Level Single Linkage is the best method, and that our simulated annealing algorithm is a good alternative. However, the Multi Level Single Linkage algorithm is implemented in an efficient dynamic way: the data are handled without extra cost in running time. Simulated annealing, on the other hand, is tested using a rather primitive implementation, which is not fully optimized. Hence, we may anticipate an increase in efficiency of the latter algorithm by using a more sophisticated implementation.

In table 4.4 the results of methods *F* and *G* are given for some of the test functions used by Aluffi-Pentini, Parisi & Zirilli [1985] (see Appendix B). For method *F*, both the running time and the number of function evaluations are given; for method *G* only the number of function evaluations is presented. Table 4.4 shows a striking difference in the number of function evalua-

tions used by both methods. Unfortunately, no figures are available on the running time of method *G*, which disables us to draw any further conclusions. Though it seems that our simulated annealing method is much faster.

table 4.4 Results for methods F and G :

function	*F*		*G*
	# f.e.**	running time	# f.e.**
P 3	780*	3.5*	241 215
P 8	2 667	7	72 851
P 16	9 018	33	66 365
P 22	1 677	2.3	74 194

* the global minimum was not found in one of the four runs;

** # f.e. is the number of function evaluations.

The effectivity of all methods seems acceptable for this set of test functions we have been investigating. These functions (especially those of Dixon & Szegö [1978b]) have only a few local minima and their dimensions range from 2 to 6. For functions with more local minima or higher dimensions the performance may be worse: Multistart, both clustering methods, and Multi Level Single Linkage have to store all minima found during execution of the algorithm, (this can be as many as 30^n for some functions, where n is the dimension, see for instance Aluffi-Pentini, Parisi & Zirilli [1985], problem 12). For higher dimensions this number is too large to handle and this will cause those methods to fail. Simulated annealing has the advantage that Markov chains are used, for which only the last point has to be stored. But the convergence of simulated annealing may become slow for these kind of functions.

5. CONCLUSION AND DISCUSSION

The problem discussed in this paper concerns the global minimization of real valued functions on $\mathbb{R}^n$. There are several methods available from the literature to solve this problem. The best method, up to now, is the Multi Level Single Linkage method developed by Rinnooy Kan & Timmer [1987a, 1987b]. This method is capable of finding the global minimum with a high probability in a reasonable amount of computer time, as long as the function has a moderate number of minima and the dimension of the search space is small. For higher dimensional spaces, problems occur due to the enormous amount of data that has to be stored; to cope with this problem a different approach seems to be necessary. Simulated annealing is proposed as such an approach. The amount of data that has to be stored while running the simulated annealing algorithm is negligible; only the current point in a Markov chain and some data used for updating some parameters are needed. Furthermore, if the number of local minima or the dimension increases, this has no effect on the amount of data stored. Therefore simulated annealing is a method that can cope with such problems. The simulated annealing algorithm performs slightly worse than the Multi Level Single Linkage method in the sense that, for most functions, a slightly larger running time is required. However, there is evidence that the total running time (including the initialization overheads) compares favourably.

The simulated annealing algorithm presented in this paper should be seen as a first step. Preliminary results show that the method is rather effective and efficient. However, further research may yield more efficient generation mechanisms. Perhaps a more sophisticated step than a uniform distributed one can be found, in which information gathered during the minimizing is used. It also might be possible to make local search steps at more suitable moments, to avoid that a relatively expensive local search step is followed by the acceptation of a large deterioration.

It is certainly possible to improve the implementation, (the local search procedure was implemented in a rather primitive way), remedying this will influence the performance positively.

It can be concluded that there are several stochastic algorithms for global minimization that perform satisfactorily, but none of these algorithms is perfect. Global optimization, therefore, remains a challenging research topic.

APPENDIX A

Test functions proposed by Dixon & Szegö [1978b] (x_i denotes the i–th coordinate of x):

GP (Goldstein and Price):

$$f(x_1,x_2) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]*$$
$$[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)].$$

$S = \{ x \in \mathbb{R}^2 \mid -2 \le x_i \le 2, i = 1, 2\}$, $x_{min} = (0,-1)$, $f(x_{min}) = 3$. There are four local minima.

BR (Branin):

$f(x_1,x_2) = a (x_2 - bx_1^2 + cx_1 - d)^2 + e (1 - f) \cos x_1 + e$

where $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $d = 6$, $e = 10$, $f = 1/(8\pi)$.

$S = \{ x \in \mathbb{R}^2 \mid -5 \le x_1 \le 10$ and $0 \le x_2 \le 15 \}$, $x_{min} = (-\pi,12.275)$; $(\pi,2.275)$; $(3\pi,2.475)$, $f(x_{min}) = 5/(4\pi)$. There are no more minima.

H3 and H6 (Hartmann's family):

$$f(x) = -\sum_{i=1}^{m} c_i \exp\left(-\sum_{j=1}^{n} a_{ij} (x_i - p_{ij})^2\right)$$

H3 (n = 3 and m = 4):

i	a_{ij}			c_i	p_{ij}		
1	3	10	30	1	0.3689	0.1170	0.2673
2	0.1	10	35	1.2	0.4699	0.4387	0.7470
3	3	10	30	3	0.1091	0.8732	0.5547
4	0.1	10	35	3.2	0.03815	0.5743	0.8828

H6 (n = 6 and m = 4):

i	a_{ij}						c_i
1	10	3	17	3.5	1.7	8	1
2	0.05	10	17	0.1	8	14	1.2
3	3	3.5	1.7	10	17	8	3
4	17	8	0.05	10	0.1	14	3.2

and

i	p_{ij}					
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

.

$S = \{ x \in \mathbb{R}^n \mid 0 \le x_i \le 1, \quad 1 \le i \le n\}$. These functions both have four local minima, $x_{loc} \approx (p_{i1},\ldots,p_{in}), f(x_{loc}) \approx -c_i$.

S5, S7 and S10 (Shekel's family):

$$f(x) = -\sum_{i=1}^{m} \frac{1}{(x - a_i)^T (x - a_i) + c_i}$$

with the dimension n = 4, m = 5, 7, 10 for S5, S7, 10 respectively, $x = (x_1,\ldots,x_n)^T$ and $a_i = (a_{i1},\ldots,a_{in})^T$.

i	a_{ij}				c_i
1	4	4	4	4	0.1
2	1	1	1	1	0.2
3	8	8	8	8	0.2
4	6	6	6	6	0.4
5	3	7	3	7	0.4
6	2	9	2	9	0.6
7	5	5	3	3	0.3
8	8	1	8	1	0.7
9	6	2	6	2	0.5
10	7	3.6	7	3.6	0.5

$S = \{ x \in \mathbb{R}^4 \mid 0 \le x_j \le 10, \ 1 \le j \le 4\}$. These functions have 5, 7 and 10 local minima for S5, S7 and S10 respectively, $x_{loc} \approx a_i, f(x_{loc}) \approx -\frac{1}{c_i}$ $(1 \le i \le m)$.

APPENDIX B

In this appendix, 4 of the 24 test functions used by Aluffi-Pentini, Parisi & Zirilli [1985] are given. These functions contain a penalty term, for Aluffi-Pentini, Parisi & Zirilli minimized over $\mathbb{R}^n$. For simulated annealing the minimization is done on S, where S just contains all unpenalized points. The penalty function is defined by

$$u(x_i, a, k, m) = \begin{cases} k\,(x_i - a)^m, & x_i > a, \\ 0 & -a \le x_i \le a, \\ k\,(-x_i - a)^m, & x_i < -a. \end{cases}$$

Problem 3 (Two-Dimensional Penalized Shubert Function):

$$f(x_1,x_2) = \left\{\sum_{i=1}^{5} i \cos[(i+1)x_1 + 1]\right\}\left\{\sum_{i=1}^{5} i \cos[(i+1)x_2 + 1]\right\} + u(x_1,10,100,2) + u(x_2,10,100,2).$$

$S = \{ x \in \mathbb{R}^n \mid -10 \le x_i \le 10, i = 1,2\}$. This function has 760 local minima, 18 of them are global.

Problem 8:

$$f(x) = (\pi/n) \left\{ k_1 \sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_1 - k_2)^2 [1 + k_1 \sin^2(\pi y_{i+1})] + (y_n - k_2)^2 \right\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4).$$

where $y_i = 1 + (x_i + 1)/4$, $k_1 = 10$ and $k_2 = 1$.

$S = \{ x \in \mathbb{R}^3 \mid -10 \le x_i \le 10, i = 1,2,3\}$, $x_{min} = (1,1,1)$, $f(x_{min}) = 0$. This funtion has roughly 5^3 local minima.

Problem 16:

$$f(x) = k_3 \left\{ \sin^2(\pi k_4 x_1) + \sum_{i=1}^{n-1} (x_i - k_5)^2 [1 + k_6 \sin^2(\pi k_4 x_{i+1})] + (x_n - k_5)^2 [1 + k_6 \sin^2(\pi k_7 x_n)] \right\}$$
$$+ \sum_{i=1}^{n} u(x_i, 5, 100, 4)$$

with $k_3 = 0.1$, $k_4 = 3$, $k_5 = 1$, $k_6 = 1$, $k_7 = 2$.

$S = \{ x \in \mathbb{R}^5 \mid -5 \le x_i \le 5, i = 1,...,5\}$, $x_{min} = (1,1,1,1,1)$, $f(x_{min}) = 0$. This function has roughly 15^n local minima.

Problem 22:

$f(x) = 10^k x_1^2 + x_2^2 - (x_1^2 + x_2^2)^2 + 10^l (x_1^2 + x_2^2)^4$ with $k = 5$ and $l = -5$.

$S = \{ x \in \mathbb{R}^2 \mid -20 \le x_i \le 20, i = 1,2\}$, $x_{min} = (0,15); (0,-15)$, $f(x_{min}) \approx -24\,775$. The origin is a local minimum.

REFERENCES

Aarts, E.H.L. & P.J.M. van Laarhoven [1985], Statistical Cooling: A General Approach to Combinatorial Optimization Problems, *Philips Journal of Research 40*, 193–226.

Aarts, E.H.L. & J.H.M. Korst [1988], *Simulated Annealing and Boltzmann machines*, Wiley, Chichester.

Aluffi-Pentini, F., V. Parisi & F. Zirilli [1985], Global Optimization and Stochastic Differential Equations, *Journal of Optimization Theory and Applications 47*, 1–16.

Bohachevsky, I.O., M.E. Johnson & M.L. Stein [1986], Generalized Simulated Annealing for Function Optimization, *Technometrics 28*, 209–217.

Chiang, T.-S., C.-R. Hwang & S.-J. Sheu [1987], Diffusion for Global Optimization in $\mathbb{R}^n$, *SIAM Journal on Control and Optimization 25*, 737–753.

De Biase, L. & F. Frontini [1978], A Stochastic Method for Global Optimization: Its Structure and Numerical Performance, in L.C.W. Dixon & G.P. Szegö (Eds.), *Towards Global Optimisation 2*, North-Holland, Amsterdam, 85–102.

Dixon, L.C.W. & G.P. Szegö (Eds.) [1978a], *Towards Global Optimisation 2*, North-Holland, Amsterdam.

Dixon, L.C.W. & G.P. Szegö [1978b], The Global Optimisation Problem: An Introduction, in L.C.W. Dixon & G.P. Szegö (Eds.), *Towards Global Optimisation 2*, North-Holland, Amsterdam, 1–15.

Doob, J.L. [1953], *Stochastic Processes*, John Wiley & Sons, New York.

Feller, W. [1957], *An Introduction to Probability Theory and Its Applications, Vol 1*, John Wiley & Sons, New York.

Geman, S. & C.-R. Hwang [1986], Diffusions for Global Optimization, *SIAM Journal on Control and Optimization 24*, 1031–1043.

Gomulka, J. [1978a], Deterministic Versus Probabilistic Approaches to Global Optimisation, in L.C.W. Dixon & G.P. Szegö (Eds.), *Towards Global Optimisation 2*, North-Holland, Amsterdam, 19–30.

Gomulka, J. [1978b], A Users Experience with Törn's Clustering Algorithm, in L.C.W. Dixon & G.P. Szegö (Eds.), *Towards Global Optimisation 2*, North-Holland, Amsterdam, 63–70.

Khachaturyan, A. [1986], Statistical Mechanics Approach in Minimizing a Multivariable Function, *Journal of Mathematical Physics 27*, 1834–1838.

Kirkpatrick, S., C.D. Gelatt Jr. & M.P. Vecchi [1983], Optimization by Simulated Annealing, *Science 220*, 671–680.

Kushner, H.J. [1987], Asymptotic Global Behavior for Stochastic Approximation and Diffusions with Slowly Decreasing Noise Effects: Global Minimization via Monte Carlo, *SIAM Journal on Applied Mathematics 47*, 169–185.

Laarhoven, P.J.M. van & E.H.L. Aarts [1987], *Simulated Annealing: Theory and Applications*, Reidel, Dordrecht.

Metropolis, N, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller & E. Teller [1953], Equation of State Calculations by Fast Computing Machines, *The Journal of Chemical Physics 21*, 1087–1092.

Price, W.L. [1978], A Controlled Random Search Procedure for Global Optimisation, in L.C.W. Dixon & G.P. Szegö (Eds.), *Towards Global Optimisation 2*, North-Holland, Amsterdam, 71–84.

Rinnooy Kan, A.H.G. & G.T. Timmer [1984], Stochastic Methods for Global Optimization, *American Journal of Mathematical and Management Sciences 4*, 7–40.

Rinnooy Kan, A.H.G. & G.T. Timmer [1987a], Stochastic Global Optimization Methods. Part I: Clustering Methods, *Mathematical Programming 39*, 27–56.

Rinnooy Kan, A.H.G. & G.T. Timmer [1987b], Stochastic Global Optimization Methods. Part II: Multi Level Methods, *Mathematical Programming 39*, 57–78.

Scales, L.E. [1985], *Introduction to Non-linear Optimization*, Macmillan, London.

Törn, A.A. [1978], A Search-Clustering Approach to Global Optimization, in L.C.W. Dixon & G.P. Szegö (Eds.), *Towards Global Optimisation 2*, North-Holland, Amsterdam, 49–62.

Vanderbilt, D. & S.G. Louie [1984], A Monte Carlo Simulated Annealing Approach to Optimization over Continuous Variables, *Journal of Computational Physics 56*, 259–271.

Weir, A.J. [1973], *Lebesgue Integration and Measure*, Cambridge University Press, Cambridge.