Size of Neighborhood More Important than Temperature for Stochastic Local Search

Heinz Mühlenbein Jörg Zimmermann
Real World Computing Partnership Theoretical Foundation GMD Laboratory
GMD Schloss Birlinghoven
53754 Sankt Augustin, Germany
E-mail: muehlenbein@gmd.de

Abstract- In the paper we investigate stochastic local search by Markov chain analysis in a high and a low dimensional discrete space. In the n-dimensional space B^n a function called Jump is considered. The analysis shows that an algorithm using a large neighborhood and never accepting worse points performs much better than any local search algorithm accepting worse points with a certain probability. We also investigate functions in the space B^n with many local optima. Here we compare stochastic local search using large neigborhoods with a local search using optimal temperature schedules which depend on the state of the Markov process.

1 Introduction

In order to understand the motivation behind our investigation we shortly describe the general context. In our research we have transformed evolutionary algorithms which recombine strings to algorithms which use search distributions. For a conceptual algorithm (Boltzmann distribution estimation algorithm BEDA) we have proven convergence to the set of global optima [MMO99]).

BEDA

- **STEP 0:** Set $t \Leftarrow 1$. Generate $N \gg 0$ points randomly.
- **STEP 1:** Select $M \leq N$ points according to Boltzmann selection. Estimate the distribution $p^s(x,t)$ of the selected set.
- **STEP 2:** Generate N new points according to the distribution $p(x,t+1) = p^s(x,t)$. Set $t \Leftarrow t+1$.
- **STEP 3:** If termination criteria are not met, go to STEP 1.

BEDA uses the *Boltzmann distribution* for selecting promising points. It is given by

$$\pi_T(x) = \frac{1}{Z_T} \exp -\frac{f(x)}{T} \tag{1}$$

Z_T is the usual partition function, defined by $\sum_x \exp -\frac{f(x)}{T}$.

It can easily be shown that the search distribution of the selected points $p^s(x,t)$ is a Boltzmann distribution if the total distribution $p(x,t)$ is a Boltzmann distribution [MMO99]. From BEDA a practical algorithm has been derived, called the Factorized Distribution Algorithm (FDA). It assumes that the search distribution can be factored into a Bayesian network $p^s(x,t) = \prod_i p(x_i|pa_i)$. pa_i are called the parents of x_i. The interested reader is referred to [MMO99, MM99]. The computational complexity of FDA depends on the size of the factors pa and the number of points required to approximate the Boltzmann distribution reasonably.

BEDA resembles simulated annealing. It uses a population of points to estimate the Boltzmann distribution instead of a single search point as simulated annealing does. It is well known that simulated annealing approximates the Boltzmann distribution for a fixed temperature. But it needs a huge amount of trials to reach equilibrium. Therfore using simulated annealing to approximate the Boltzmann distribution is not practical. But it might be that simulated annealing itself is an interesting alternative to the population search done by BEDA. From this perspective we decided to analyze simulated annealing in a clearly defined mathematical context.

2 Simple Simulated Annealing

Simulated annealing is a probabilistic method proposed by Kirpatrick et al. [KTV83]. The origin and the choice of the algorithm lie in the physical annealing process. Here we just give a short synopsis. The interested reader is referred to [AKvL97] for a more detailed description. The basic elements of simulated annealing (SA) are the following:

1. A finite set S.
2. A real-valued cost function c defined on S. Let $S^* \subset S$ be the set of global minima of the function c.
3. For each $x \in S$, a set $S(x) \subset S - \{x\}$, called the set of neighbors of x.
4. For every x, a collection of positive coefficients $q_{xy}, y \in S(x)$, such that $\sum_{y \in S(x)} q_{xy} = 1$. It is assumed that $y \in S(x)$ if and only if $x \in S(y)$.
5. A non increasing function $T : N \rightarrow [0, \infty]$, called the cooling schedule. N is the set of integers, and $T(t)$ is called the temperature at time t.
6. An initial state $x(0) \in S$.

Thus SA consists of a discrete-time inhomogeneous Markov chain $x(t)$, whose evolution is as follows. Choose a neighbor y of $x(t)$ at random; the probability that any particular $y \in S$ is selected is equal to q_{xy}. Once y is chosen, the next state $x(t+1)$ is determined as follows:

- If $f(y) \leq f(x)$, then $x(t+1) = y$.
- If $f(y) > f(x)$ then $x(t+1) = y$ with probability $\exp\left(-(f(y) - f(x)/T(t)\right)$

The rationale behind the SA algorithm is best understood by considering a homogeneous Markov chain $x_T(t)$ in which temperature $T(t)$ is held at constant value T. Let us assume that the Markov chain is irreducible and aperiodic and that $q_{xy} = q_{yx}$. Then its invariant probability distribution is given by the *Boltzmann* or *Gibbs* distribution (equation 1).

2.1 Convergence Analysis

Having defined the algorithm, we now address its performance. The main questions are

1. Under which assumptions does $x(t)$ converge to the optimal set S^*?
2. How fast does the convergence to S^* take place?

The first question has been more or less answered completely [AKvL97]. Basically there are two convergence results. The first theorem assumes that for each $T(t)$ the algorithm is run until equilibrium is reached. Then for $\lim_{t\to\infty} T(t) = 0$ convergence to S^* in probability is obviously obtained. This convergence theorem is of limited values because there exists no measures to discover when equilibrium is reached.

Definition: *We say that state x communicates with S^* at height h if there exists a path in S (with each element of the path being a neighbor of the preceding element) that starts at x and ends at some element S^*, and such that the largest value of f along the path is $f(x) + h$. We set d^* to be the smallest number that every $x \in S$ communicates with S^* at height d^*, i.e., d^* is the height of the deepest, non-global minimum.*

A convergence theorem for inhomogeneous Markov chains has been proven in [Haj88].

Theorem 1 (Hajek) *The Simulated Annealing algorithm converges if and only if* $\lim_{t\to\infty} T(t) = 0$ *and*

$$\sum_{t=1}^{\infty} \exp -\frac{d^*}{T(t)} = \infty \tag{2}$$

There are not many cooling schedules which fulfil equation (2). The most popular cooling schedules are of the form

$$T(t) = \frac{d}{logt} \tag{3}$$

Here the theorem states that SA converges if and only if $d \geq d^*$. The constant d^* is a measure of the difficulty for $x(t)$ to escape from local minimum and go from a non optimal state to S^*.

The practical relevance of this result for algorithms using simulated annealing is limited. Using a logarithmic cooling schedule gives a slow convergence speed because T is approaching 0 very slow. We will investigate the speed of convergence for a specific class of functions in the next section.

3 The Optimization Problem

In order to make comparison with our results from evolutionary computation easier, we consider **maximization** instead of **minimization**. We investigate functions defined on $S_1 = B^n$, i.e., $\boldsymbol{x} \in S_1$ is a binary string of size n, and $S_2 = \{1, \ldots, n\}$:

$$f_1(\boldsymbol{x}, n, gap) := \text{Jump}(n, gap, |\boldsymbol{x}|_1) \qquad \boldsymbol{x} \in S_1 \tag{4}$$

$$f_2(i, n, gap) := \text{Jump}(n, gap, i) \qquad i \in S_2 \tag{5}$$

$|x|_1$ is just the sum of 1-bits and $Jump$ is given by:

$$\text{Jump}(n, gap, i) := \begin{cases} i & i < n - gap \\ 2 * (n - gap - 1) - i & n - gap \leq i \\ n & i = n \end{cases} \tag{6}$$

The parameter gap defines the number of steps one has to go downhill in order to reach the unique maximum. It is identical to d^* used in theorem 1. If $gap = 0$, f_1 is the linear function $OneMax$ and f_2 is the identity. B^n can be viewed as the set of vertices of an n-dimensional hypercube, whereas $\{1, \ldots, n\}$ can be interpreted as points on a 1-dimensional straight line. Hence f_1 and f_2 represent a high-dimensional and a low-dimensional optimization problem.

We will investigate the computational complexity of simulated annealing for a fixed temperature and neighborhoods of different size. Our performance criterion will be the expected number of trials to reach the optimum. This is called the *expected first passage time* $E(PT)$ – in the following abbreviated by τ – in Markov chain analysis. We first investigate stochastic local search in $S_1 = B^n$.

4 Markov Chain Analysis in B^n

Let $\mathbf{M} = (m_{ij})$ be the transition matrix of a homogeneous Markov chain with N states, labelled with $1, \ldots, N$. The expected first passage times τ_{ij} from state i to state j can be determined from a set of linear equations:

$$\tau_{ii} = 0\ ,\ 1 \leq i \leq N \tag{7}$$

$$\tau_{ij} = 1 + \sum_{k=1}^{N} m_{ik}\tau_{kj}\ ,\ 1 \leq i, j \leq N,\ i \neq j \tag{8}$$

The right-hand side of equation (8) results from *unfolding* the Markov chain one step. In order to compute the expected first passage time from state 1 to state N it is sufficient to consider the vector of passage times $\vec{\tau}_{N} = (\tau_{1,N}, \ldots, \tau_{N-1,N})^T$. Let $\hat{\mathbf{M}}$ denote the reduced transition matrix resulting from $\mathbf{M}$ by deleting row N and column N. By $\mathbf{1}_{N-1}$ we denote the vector consisting of $N-1$ ones. Translating the above equations in matrix notation and solving for $\vec{\tau}_{N}$ leads to:

Theorem 2 *The vector of expected first passage times from states* $1, \ldots, N-1$ *to state* N *can be computed by the following equation:*

$$\vec{\tau}_{N} = (\mathbf{I} - \hat{\mathbf{M}})^{-1} \cdot \mathbf{1}_{N-1} \tag{9}$$

Theorem 2 shows that the computation of passage times is essentially a matrix inversion problem. The matrix $(\mathbf{I} - \hat{\mathbf{M}})^{-1}$ is called the *fundamental matrix*. Several other system characteristics can be expressed as functions of the fundamental matrix or its elements [Bha84, KS60].

A Markov chain has *neighborhood size* s if direct transitions are only possible between states satisfying $|i-j| \leq s$. For the special case of $s = 1$ an analytical solution for the expected first passage times is possible. In that case, the transition matrix has the following form:

$$\begin{pmatrix} p_1^0 & p_1^+ & 0 & 0 & \ldots & 0 & 0 & 0 \\ p_2^- & p_2^0 & p_2^+ & 0 & \ldots & 0 & 0 & 0 \\ 0 & p_3^- & p_3^0 & p_3^+ & \ldots & 0 & 0 & 0 \\ . & . & . & . & & . & . & . \\ 0 & 0 & 0 & 0 & \ldots & p_{N-1}^- & p_{N-1}^0 & p_{N-1}^+ \\ 0 & 0 & 0 & 0 & \ldots & 0 & p_N^- & p_N^0 \end{pmatrix} \tag{10}$$

Let $\tau_{i,N}$ be abbreviated by τ_i. Using (7) and (8) we get the following system of equations:

$$\begin{aligned} \tau_1 &= 1 + p_1^0 \tau_1 + p_1^+ \tau_2 \\ \tau_i &= 1 + p_i^- \tau_{i-1} + p_i^0 \tau_i + p_i^+ \tau_{i+1} \\ \tau_{N-1} &= 1 + p_{N-1}^- \tau_{N-2} + p_{N-1}^0 \tau_{N-1} \end{aligned}$$

The above system of equations can be rewritten into (note that $p_i^- + p_i^0 + p_i^+ = 1$):

$$(\tau_i - \tau_{i-1}) p_i^- = 1 + (\tau_{i+1} - \tau_i) p_i^+ \quad 1 \leq i \leq N-1 \tag{11}$$

where the new variables τ_0 and τ_N are set to 0. Defining $\delta_i = \tau_i - \tau_{i-1}$, equation (11) leads to a recursion which is solvable by induction:

$$\delta_{i+1} = \frac{p_i^-}{p_i^+} \delta_i - \frac{1}{p_i^+} \quad 1 \leq i \leq N-1 \tag{12}$$

Instead of an initial value we have the following constraint on the δ_i's:

$$\sum_{i=1}^{N} \delta_i = \tau_N - \tau_0 = 0 \tag{13}$$

A solution for recursion (12) and constraint (13) can be retranslated into a solution for the τ_i's. The result is a general formula for the expected first passage times from state i to state N for Markov chains with neighborhood size 1:

$$\tau_i = \sum_{j=i}^{N-1} \frac{1}{p_j^+ P^*(j)} \sum_{k=1}^{j} P^*(k) \quad 1 \leq i \leq N-1 \tag{14}$$

Here $P^*(i)$ denotes the probability of state i in the equilibrium distribution [FHW79]:

$$P^*(i) = \frac{1}{Z} \prod_{j=2}^{i} \frac{p_{j-1}^+}{p_j^-} \quad 1 \leq i \leq N \tag{15}$$

and Z is a normalizing factor, called the *partition function*:

$$Z = \sum_{i=1}^{N} \prod_{j=2}^{i} \frac{p_{j-1}^+}{p_j^-} \tag{16}$$

Let us now consider the case of simulated annealing on a binary string of length n. There are 2^n states. Neighborhood is defined by hamming distance, i.e., two strings are neighbors if their hamming distance is less or equal to a given h. This defines a set of neighborhoods which we will call *h-hamming neighborhoods*. A new search point is generated by sampling uniformly the neighborhood of the current search point.

The Markov chain analysis of this setting is simplified by the following *state space abstraction*: two strings $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are considered equivalent if $|\boldsymbol{x}_1|_1 = |\boldsymbol{x}_2|_1$, i.e., if they have the same number of 1-bits. The decisive fact is now that the probability, that a string in equivalence class j becomes in one step a string in equivalence class k does not depend on the specific string in equivalence class j. Hence there are well-defined transition probabilities between string classes, resulting in a Markov chain having the string classes as atomic states. If we start with strings of length n, the abstract Markov chain has $n+1$ states because $|\boldsymbol{x}|_1$ ranges over $0, \ldots, n$. In order to apply the above equations, we therefore set $N = n+1$. The neighborhood size s of the abstract Markov chain is equal to h, the maximal hamming distance between neighbors in string space. Let $\beta = 1/T$ denote the inverse temperature. Then for $s = 1$ the transition probabilities of the abstract Markov chain can be expressed as follows:

$$p_i^- = \frac{i-1}{n} \exp\left(-\beta \cdot |f(i-1) - f(i-2)|^+\right) \tag{17}$$

$$p_i^+ = \frac{n-i+1}{n} \exp\left(-\beta \cdot |f(i-1) - f(i)|^+\right) \tag{18}$$

$$p_i^0 = 1 - p_i^+ - p_i^- \tag{19}$$

Here $f(\cdot)$ denotes an arbitrary fitness function on the abstract states, i.e., $f(\cdot)$ can be represented as a composition $f(|\boldsymbol{x}|_1)$. $|\boldsymbol{x}|^+$ is an abbreviation for $\max(x, 0)$. Plugging these transition probabilities into formula (15) for the equilibrium distribution and the result into formula (14) yield

Theorem 3 *Stochastic local search acting in B^n with 1-hamming neighborhoods, uniform sampling, inverse temperature β, and a fitness function $f(|\boldsymbol{x}|_1)$ has the following expected first passage times from a string $\boldsymbol{x}$ with $|\boldsymbol{x}|_1 = i$ to the string $\boldsymbol{x}^* = (1, \ldots, 1)$:*

$$\begin{aligned}\tau_i &= \sum_{j=i}^{n} \frac{\exp\left(\beta \cdot |f(j-1) - f(j)|^+\right)}{\binom{n-1}{j-1}} \cdot \\ &\quad \sum_{k=1}^{j} \binom{n}{k-1} \exp\left(\beta \cdot (f(k) - f(j))\right)\end{aligned} \tag{20}$$

In the derivation of formula (20) we used the identity $|x|^+ - |-x|^+ = x$ to simplify the second exponential expression. This general formula will be applied to special cases.

Theorem 4 *For $\beta = 0$ the expected first passage time τ_1 from $\boldsymbol{x} = (0, \ldots, 0)$ to the unique optimum $\boldsymbol{x}^* = (1, \ldots, 1)$ is given by $\tau_1 \approx 2^n$.*

Proof: This is the case of a random walk. The passage times are independent from the fitness function. Applying formula (20) yields:

$$\tau_i = \sum_{j=i}^{n} \frac{1}{\binom{n-1}{j-1}} \sum_{k=0}^{j-1} \binom{n}{k} \qquad 1 \le i \le n \tag{21}$$

From this formula we can get tight bounds on τ_1:

$$2^n - 1 \le \tau_1 \le (2^n - 1)(1 + \frac{4}{n}) \qquad n \ge 1 \tag{22}$$

These bounds determine the asymptotic growth of τ_1. The lower bound results from the last addend in (21), i.e., for $j = n$, and the upper bound can be derived by reordering the addends in (21) and using the inequality

$$\sum_{j=0}^{n-1} \frac{1}{\binom{n-1}{j}} \le 2 + \frac{4(n-3)}{(n-1)(n-2)} \tag{23}$$

■

For a random walk the expected first passage time to the global optimum grows *exponentially* in n. Next we consider stochastic hill climbing.

Theorem 5 *For $f_1(\cdot)$, $gap = 0$ and $\beta = \infty$ the expected first passage time τ_1 from $\boldsymbol{x} = (0, \ldots, 0)$ to the unique optimum $\boldsymbol{x}^* = (1, \ldots, 1)$ is given by*

$$\tau_1 = n \sum_{j=1}^{n} \frac{1}{j} \tag{24}$$

Proof: The condition $gap = 0$ implies that $f_1(k) - f_1(j) = k - j$ for all j, k. Thus, equation (20) instantiates in a first step to:

$$\begin{aligned}\tau_i &= \sum_{j=i}^{n} \frac{1}{\binom{n-1}{j-1}} \sum_{k=1}^{j} \binom{n}{k-1} exp(\beta \cdot (k-j)) \\ &= \sum_{j=i}^{n} \frac{1}{\binom{n-1}{j-1}} \Bigg[\binom{n}{j-1} + \\ &\quad \sum_{k=1}^{j-1} \binom{n}{k-1} \exp\left(-\beta \cdot (j-k)\right) \Bigg]\end{aligned}$$

Now we have for all j:

$$\lim_{\beta \to \infty} \sum_{k=1}^{j-1} \binom{n}{k-1} \exp\left(-\beta \cdot (j-k)\right) = 0$$

and therefore

$$\begin{aligned}\tau_i &= \sum_{j=i}^{n} \frac{\binom{n}{j-1}}{\binom{n-1}{j-1}} = n \sum_{j=1}^{n-i+1} \frac{1}{j} \\ &\approx n \ln(n - i + 1)\end{aligned}$$

Finally for $i = 1$ we get:

$$\tau_1 = n \sum_{j=1}^{n} \frac{1}{j} \approx n \ln n$$

■

The above theorems are valid for neighborhood size $s = 1$. The next conjecture deals with arbitrary neighborhood size s. The approximate analysis is valid for $\beta = \infty$ and $gap = 0$.

Conjecture: For $f_1(\cdot)$, $gap = 0$, $\beta = \infty$ and neighborhood size s the expected first passage time τ_1 to move from the initial point $\boldsymbol{x} = (0, \ldots, 0)$ to the unique optimum $\boldsymbol{x}^* = (1, \ldots, 1)$ is approximately given by

$$\tau_1 \approx \binom{n}{s} \left(\binom{n}{s}^{-1} + \binom{n-s}{s}^{-1} + \cdots + \binom{2s}{s}^{-1} + 1 \right) \tag{25}$$

We assume that n is an integer multiple of s.

We derive the conjecture by the following line of thought: First consider $s = 1$. We use the fact that the expected time till the occurrence of an event with probability p in an independent sequence of trials is $\frac{1}{p}$. Starting from $\boldsymbol{x} = (0, \ldots, 0)$, the probability is 1 to get one bit correct. If one bit is correct, then the probability is $1 - 1/n$ to get a second bit correct, etc. Therefore we obtain:

$$\tau_1 = \frac{n}{n} + \frac{n}{n-1} + \cdots + \frac{n}{1}$$

This is an intuitive derivation of Theorem 5. We next investigate $s = 2$. We assume that only transitions from 0 to 2 to 4, etc., are made. Here the transition probability from state 0 to state 2 is 1, from state 2 to state 4 is $\binom{n-2}{2}/\binom{n}{2}$. There is only one transition from $n-2$ bits to n bits. The probability is given by $\binom{n}{2}^{-1}$. Summing the inverse from these terms gives the conjecture. Similar approximations we do for arbitrary s. ■

5 Results for $Jump$

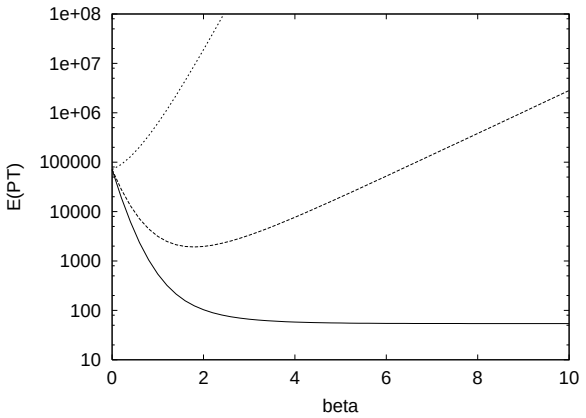

Figure 1. Expected first passage time PT versus β for $gap = 0, 1, 4$.

We first discuss the case $n = 16$ for neighborhood size $s = 1$. In Figure 1 the expected first passage time τ_1 for the string model is shown for three different gap sizes. The following behavior can be observed: all curves start with the same value for $\beta = 0$. This is not surprising, it is just the random walk. There are three types of curves. For $gap = 0$ the expected first passage time τ decreases from $\beta = 0$ till $\beta = \infty$. In this case only improvements should be accepted. For $gap = 1$ there exists an optimal inverse temperature β_{opt} in $(0, \infty)$. τ decreases from $\beta = 0$ till β_{opt} and then increases again. For $gap = 4$ τ increases with β. Here random walk ($\beta = 0$) is the best strategy.

This result can be generally observed. If the gap is large compared to the size of the problem, random walk is the best. If the gap is small then $\beta = \infty$ is the best parameter. For gaps of medium size there exists an optimal parameter β_{opt}.

We now investigate the scaling for two different gap values. We have approximately determined the optimal β, which gives the smallest τ. For the computation we have used equation (9). The results are displayed in Table 1.

n	g	β_{opt}	τ	$\hat{\tau}_{gap}$
8	1	0.8	208.8	71.2
	3	0.0	312.0	170.6
16	1	1.8	1925.7	774.3
	3	0.2	61913.1	4975.5
24	1	2.3	6824.8	2872.5
	3	0.9	1.95E6	205697.4
32	1	2.6	16575.0	6893.4
	3	1.4	1.95E7	2.91E6
64	1	3.3	137909.3	55526.6
	3	2.2	3.68E9	5.13E8
96	1	3.8	471361.5	205983.0
	3	2.6	7.12E10	8.63E9
128	1	4.1	1.12E6	494307.6
	3	2.9	5.67E11	6.71E10

Table 1
Problem size n with two gaps g; $\hat{\tau}_{gap}$ from equation (26).

We have not been able to derive a scaling law from these numbers. In order to obtain a conjecture, we make a simplified analysis.

Theorem 6 *For inverse temperature β, the expected first passage time τ_{gap} to cross a gap of size g can be lower bounded by*

$$\tau_{gap} \geq \hat{\tau}_{gap} = \exp(\beta \cdot g) \frac{n^{g+1}}{(g+1)!} \tag{26}$$

Proof: Let us assume that the gap is crossed and the optimum is reached in one sweep. The probability $\hat{p}_{gap}$ of this event can be computed by multiplying the one step probabilities. We obtain

$$\hat{p}_{gap} = \exp(-\beta \cdot g) \frac{(g+1)!}{n^{g+1}}$$

The resulting passage time is $\hat{\tau}_{gap} = \hat{p}_{gap}^{-1}$. ■

In Table 1 this estimate is compared with exact computations. The estimate shows a comparable scaling behavior.

5.1 Performance for larger neighborhoods

Our analysis leads to a surprising simple conclusion, which we state as an empirical law:

Empirical Law: *For $Jump$ in the space B^n, the expected first passage time to the global optimum is minimal if the size of the neighborhood is equal to $gap + 1$ and the stochastic algorithm does not accept worse points.*

With $s = gap + 1$ the algorithm directly jumps from the local maximum to the global maximum. We have not been able to analytically prove this conjecture. Table 2 displays some numerical values for $s = gap + 1$ and $\beta = 10$.

n	g	τ	g	τ	g	τ
8	1	47.6	2	96.7	3	164.6
16	1	191.0	2	718.6	3	2538.5
24	1	430.0	2	2379.1	3	13026.6
32	1	764.6	2	5590.5	3	41633.0
64	1	3059.4	2	44202.2	3	680863.5
96	1	6885.1	2	148674.6	3	3.4E6
128	1	12243.1	2	351894.1	3	1.1E7

Table 2
τ for $s = g + 1$ and $\beta = 10$.

s	n	τ	n	τ	n	τ
1	32	129.9	64	303.7	128	696.1
2	32	715.7	64	2851.5	128	11383.0
3	32	5587.5	64	44169.6	128	351394.0
4	32	41579.5	64	680146.6	128	1.1E7
5	32	242903.0	64	8.3E6	128	2.7E8

Table 3
τ for $gap = 0$, $\beta = \infty$.

A comparison between Tables 1 and 2 shows that the large neighborhood $s = g + 1$ outperforms the small neighborhood $s = 1$ by order of magnitudes. It is fairly obvious that a large neighborhood smoothes a rugged landscape. All local minima which are less than s away from the next local maxima are jumped over. The algorithm will find the optimum if there is a transition path with long jumps to one of the global optima. Along this path the function values do not decrease. Therefore the numerical results for $gap = 0$, $gap = 1$, etc., are identical as long as the neighborhood size is large enough ($s \geq gap + 1$).

In table 3 we investigate the dependency of τ on the neighborhood size s. Equation (25) can be used as a guidance for an approximate scaling law. We obtained the following approximations:

Empirical Law: *For $\beta = \infty$ the algorithm scales as $\tau \approx c(s)n^s$. The numerical fits are as follows:*

$$\begin{aligned} \tau &\approx n \ln n \quad s = 1 \\ \tau &\approx 0.75n^2 \quad s = 2 \\ \tau &\approx 0.16n^3 \quad s = 3 \\ \tau &\approx 0.04n^4 \quad s = 4 \\ \tau &\approx 0.007n^5 \quad s = 5 \end{aligned}$$

The scaling law shows that the expected first passage times increase polynomial in n. The coefficient $c(s)$ decreases with increasing s, because we need less steps to reach the optimum.

We now analyse stochastic local search in one dimension.

6 Markov Chain Analysis in S_2

We start the discussion with two Theorems.

Theorem 7 *For $\beta = 0$ the expected first passage time τ_1 from $x = 1$ to $x = n$ is given by*

$$\tau_1 = n(n-1) \tag{27}$$

Proof: We apply equation (14). We have $p_j^+ = p_j^- = 1/2$. Therefore we obtain $P^*(i) = 1/N$ and $Z = N$. In S_2 we have $N = n$ states. Using equation (14) gives the conjecture. ■

Hence in case of a random walk the expected first passage time to the global optimum grows *quadratically* in n. The next theorem is proven similarly.

Theorem 8 *For f_2 with $gap = 0$ and $\beta = \infty$ the expected first passage time from $x = 1$ to the unique optimum $x^* = n$ is given by $\tau_1 = 2(n-1)$.*

These two results have to be interpreted properly. Space S_2 has only n states. This means that optimization by complete enumeration takes n steps. Any kind of biased stochastic local search with two neighbors ($s = 1$) needs between n^2 and $2n$ trials. This is obviously worse than complete enumeration. Stochastic local search with small neigborhoods is extremely inefficient in S_2.

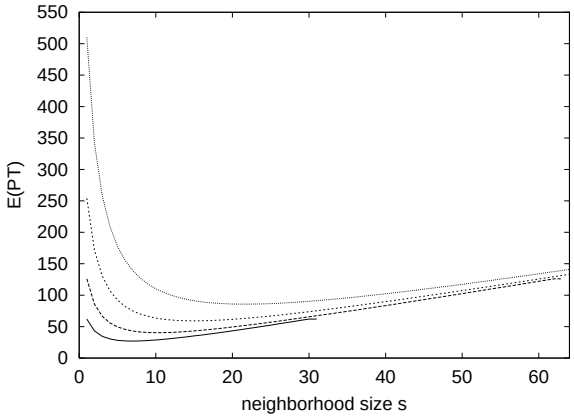

Figure 2. Expected first passage time PT versus neighborhood size s (n = 32, 64, 128, 256, gap = 0, $\beta = \infty$).

In figure 2 the expected first passage time τ_1 is shown for various neighborhood sizes. For each problem size n there exists an optimal neighborhood size. For $n = 256$ the optimal neighborhood size s is 22 with an expected passage time $\tau_1 = 85.73$. This is a substantial reduction compared to neighborhood size $s = 1$. For $s = 1$ we have $\tau_1 = 510$. Whereas for B^n the optimal neighborhood size is $s = 1$ for $\beta = \infty$, we have an optimal neighborhood size $s_{opt} > 1$ for S_2. These results confirm our hypothesis: the size of the neighborhood is extremely important for stochastic local search.

7 Temperature Schedules

Most of the suggested annealing schedules have been from the *apriori* type. They are computed before the run starts and not

adjusted to the actual behavior of the system. Examples are

$$\begin{aligned}\beta(t) &= a\exp(bt)\\ \beta(t) &= a(t+b)\\ \beta(t) &= a\ln(t+b)\end{aligned}$$

The third schedule is used in Theorem 1. All schedules increase β monotonically. Such a strategy is obviously bad for functions where a valley has to be crossed after a large number of trials.

It is very difficult to theoretically analyze τ for all possible schedules for an arbitrary function. We first analyze the artificial function $Jump$, because for this function an optimal schedule can be computed. This schedule is not of the apriori type. The schedule depends on the state of the Markov process.

Theorem 9 *For $Jump(n, gap, |\boldsymbol{x}|_1)$ with neighborhood size $s = 1$, the optimal state dependent schedule is*

$$\begin{aligned}\beta(i) &= \infty \quad i < n - gap - 1\\ \beta(i) &= 0 \quad n - gap - 1 \le i < n - 1\\ \beta(i) &= \infty \quad i = n - 1\end{aligned}$$

Proof: For $i < n - gap - 1$ the probability to get to state $i + 1$ is maximized for $\beta(i) = \infty$. At state $i = n - gap - 1$ we have a local optimum. The fitness values for the neighboring states $i - 1$ and $i + 1$ are equal. Therefore the acceptance probability for both states is the same. In order to leave the state i as soon as possible β should be zero. At state $i = n - gap$ we have to go down further. Any $\beta > 0$ would prefer to go back. Therefore $\beta = 0$ is optimal. ∎

This state dependent schedule is obviously better than any apriori schedule. The optimal state dependent schedule is not monotonic!

In table 4 the first passage time τ is shown for the optimal schedule.

n	g	τ	g	τ
8	1	114	3	265
16	1	771	3	7440
24	1	2478	3	58152
32	1	5745	3	251290
64	1	44644	3	8.44E6
96	1	149438	3	6.53E7
128	1	352887	3	2.78E8

Table 4
τ for optimal schedule.

Comparing the results of table 4 with tables 1 and 2, we see that the optimal schedule reduces τ substantially compared to any fixed β. But for this function local search with optimal neighborhood performs still better.

We have not been able to find an *apriori* temperature schedule which finds the optimum in less than 10^7 trials for $n \ge 24$. The function $Jump$ is a worst case for any monotonic apriori schedule. β has to be very small nearby the optimum. But any monotonic schedule has the largest values for β at the end of the search. This makes it very unlikely that the local search crosses the valley.

8 Numerical results from simulations

The function $Jump$ is very artificial. We now discuss stochastic local search for functions with many local optima.

$$\begin{aligned}f_1 &:= 1,2,1,0,1,2,3,2,1,2,3,4,3,2,3,4,5,4,3,\\ &\quad 4,5,6,5,4,7\\ f_2 &:= 1,2,1,0,1,2,3,2,1,2,3,4,3,2,3,4,5,4,3,\\ &\quad 4,5,6,5,4,5,6,7\\ f_3 &:= 1,2,1,0,2,2,3,2,3,2,3,4,4,2,3,4,5,4,3,\\ &\quad 4,6,6,5,4,7,6,8\end{aligned}$$

The optimal state dependent schedule for f_1 and f_2 can easily be derived. It is given by $\beta(t) = \infty$ if one has to go uphill, or if the search is at the bottom of a valley. It is given by $\beta(t) = 0$ if the local search is at a local optimum, or has to go downhill.

The optimal schedule for f_3 is difficult to determine. Let us consider state $|x|_1 = 24$. Its fitness value is 7. The fitness values of the neighboring states are 4 and 6. In order to go with higher probability to the state with fitness value 6, we should use some value $\beta > 0$ instead of $\beta = 0$. In $sched.2$ we have set $\beta(24) = 2$ instead of 0.

$func$	$meth.$	τ	std
f_1	$sched.$	14400	14400
f_1	$N = 4$	17700	15400
f_2	$sched.$	1300	1240
f_2	$N = 5$	79278	80300
f_3	$sched.1$	279600	2600000
f_3	$sched.2$	46341	40000
f_3	$N = 4$	20690	18000

Table 5
τ for different schedules and neighborhoods.

Table 5 displays the results of simulations. The entrance $sched.$ means that one of the above described state dependent annealing schedules have been used. std denotes the standard deviation.

For function f_1 local search with optimal neighborhood performs as good as local search with optimal schedule. For function f_2 local search with optimal schedule performs far better than optimal neighborhood search. For function f_3 the optimal neighborhood search performs better than the search with schedules $sched.1$ and $sched.2$. Note the large difference in the performance of $sched.1$ and $sched.2$. The interpretation of these results is difficult. Small changes in the fitness function (see f_2 and f_3) may cause a huge difference in the search performance.

For all three functions the optimal state dependent schedule is not monotonic. It mainly switches between $\beta = \infty$ and $\beta = 0$.

We have not been able to solve the optimization problem by any apriori schedule in less than 10^7 steps. This observation shows that *any constant or monotonic increasing schedule is far away from an optimal schedule.*

9 The Problem of Simulated Annealing

Our research indicates the weak point of simulated annealing for fast optimization. SA does not try to develop a model of the function to be optimized. It just performs a biased random walk. The random walk does not have any preferred direction. The bias is given by an external parameter, the temperature. There is no feed back between the structure of the landscape and the temperature.

All landscapes considered have a certain structure. The heights of the hills are increasing if the number of bits on increases. Our population algorithm FDA detects and exploits this information. By generating many points in a certain area, it is possible to select the better ones. The better ones do have on the average more bits on than the whole of the population. In the next step we generate new points in the vicinity of the selected points. This means we are moving into the right direction. All functions considered have been easily solved by our simplest population algorithm UMDA [MM00].

Simulated annealing uses purely local information. It even is memory less. It does blind moves, the acceptance of the moves based only on the temperature. This might be a good technique to approximate important distributions in physics. It is obviously not a good technique for fast optimization.

10 Summary and Discussion

Much of the folklore of simulated annealing is based on arguments and techniques derived from statistical physics. We have tried to mathematically analyze tractable cases. The results all point into one direction: variation of the temperature has not a large influence on the computational efficiency. Much more critical is the size of the neighborhood.

We have shown that in the one dimensional space S_2 any local search with a small neighborhood performs even worse than enumeration. In the space $S_1 = B^n$ only a state dependent optimal schedule performs as good as local search with large neighborhoods. We have shown for the space B^n that *any apriori annealing schedule* performs much worse than both of the mentioned methods.

To make the analysis simple we studied local search algorithms where the neighborhood is large enough so that there exist an uphill path to the optimum. Of course one can argue that the size of a good neighborhood is not known. But the size can be determined adaptively, starting with neighborhood size 1. The neighborhood size is increased if there are too many trials which are unsuccessful. This technique has just been discovered in operations research. Its name is *variable neighborhood search* [MH97].

Another method to introduce a variable neighborhood is to use a population of search points instead of a single point. If the new search points are generated by a probability distribution, than we have a *probabilistic neighborhood*. The neighborhood is dynamic. A new configuration from the neighborhood is not chosen uniformly distributed, but by a problem specific probability distribution. This probability distribution generates more often points nearby the average, and more seldom far away from the average. This is done in our FDA algorithm [MM99, MMO99].

The results of this paper confirm what has already been found empirically: the neighborhood is much more important than the annealing schedule [Fox95]. If the neighborhood is chosen properly, then there is no need to introduce a complicated annealing schedule. The question raised in [HS89]: To cool or not to cool has the answer: Not to cool.

Bibliography

[AKvL97] E.H. Aarts, H.M. Korst, and P.J. van Laarhoven. Simulated annealing. In E. Aarts and J.K. Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 121–136, Chichester, 1997. Wiley.

[Bha84] U. N. Bhat. *Elements of Applied Stochastic Processes*. Wiley, New York, 1984.

[FHW79] F.-J. Fritz, B. Huppert, and W. Willems. *Stochastische Matrizen*. Springer, Berlin, 1979.

[Fox95] B. L. Fox. Faster simulated annealing. *Siam J. Optimization*, 5:488–505, 1995.

[Haj88] B. Hajek. Cooling schedules for optimal annealing. *Math. Oper. Res.*, 13/2:311–329, 1988.

[HS89] B. Hajek and G. Sasaki. Simulated annealing – to cool or not. *Syst. Contr. Lett.*, 12:443–447, 1989.

[KS60] J. G. Kemeny and J. N. Snell. *Finite Markov Chains*. Van Nostrand, Princeton, N.J., 1960.

[KTV83] S. Kirpatrick, G. Toulouse, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[MH97] N. Mladenovic and P. Hansen. Variable neighborhood search. *Computers Oper. Res.*, 24:1097–1100, 1997.

[MM99] H. Mühlenbein and Th. Mahnig. FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376, 1999.

[MMO99] H. Mühlenbein, Th. Mahnig, and A. Rodriguez Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5:215–247, 1999.

[MM00] H. Mühlenbein and Th. Mahnig. Evolutionary Algorithms: From Recombination to Search Distributions. In *Theoretical Aspects of Evolutionary Computing*, L. Kallel, B. Naudts, A. Rogers (eds.), Natural Computing, Springer, New York, pp. 137-176, 2000.