Comparing the adaptive Boltzmann selection schedule *SDS* to truncation selection

Thilo Mahnig
GMD – Schloss Birlinghoven
53754 Sankt Augustin, Germany
mahnig@gmd.de

Heinz Mühlenbein
GMD – Schloss Birlinghoven
53754 Sankt Augustin, Germany
muehlen@gmd.de

Abstract

FDA (the Factorized Distribution Algorithm) is an evolutionary algorithm that combines mutation and recombination by using a distribution. The distribution is estimated from a set of selected points. It is then used to generate new points for the next generation. *FDA* uses a factorization to be able to compute the distribution in polynomial time. Previously, we have shown a convergence theorem for *FDA*. But it is only valid using Boltzmann selection. Boltzmann selection was not used in practice because a good annealing schedule was lacking. Using a Taylor expansion of the average fitness of the Boltzmann distribution, we have developed an adaptive annealing schedule called *SDS*. The inverse temperature β is changed inversely proportional to the standard deviation. In this work, we compare the resulting scheme to truncation selection both theoretically and experimentally with a series of test functions. We find that it behaves similar in terms of complexity, robustness and efficiency.

Keywords: *genetic algorithms, Boltzmann distribution, Boltzmann selection, truncation selection*

1 Introduction

It is well known that evolutionary algorithms have difficulties in optimizing functions with nonlinear interacting variables. In order to optimize these functions, many variables have to be changed together in a certain manner to obtain an improvement.

In previous work [MMO99], we have introduced the *FDA* (Factorized Distribution Algorithm) which uses a probability distribution that captures the dependencies between the variables. The Boltzmann distribution turned out to be especially suited for theoretical analysis. The distribution remains a Boltzmann distribution after selection, if Boltzmann selection is used. We have achieved a convergence result for this algorithm. But the convergence theorem holds only for Boltzmann selection. This selection scheme is not often used, because similarly to simulated annealing it needs an annealing schedule, which is difficult to choose.

Using the results from [MM01], we will use a Taylor expansion of the average fitness of the Boltzmann distribution. This expansion allows us to develop an efficient adaptive annealing schedule for populations. With this annealing schedule, *FDA* is invariant under positive linear transformations of the function to optimize.

The outline of the paper is as follows. In section 2 the concept of optimization using distributions is introduced. In section 3, the Boltzmann distribution is defined and we discuss why it is a suitable distribution for optimization. However, in general, calculation of the Boltzmann distribution requires an exponential computational effort. In section 4, we describe how to factorize the Boltzmann distribution to obtain a polynomial algorithm. The expansion of the average fitness and the resulting annealing schedule follow in section 5. Because truncation selection has proven to be both robust and efficient, we finally compare the resulting selection scheme *SDS* to truncation selection in section 6.

2 Definitions

Our goal is optimize (maximize) a discrete function $f(x)=f(x_1,\dots,x_n)$, called the fitness function. For notational simplicity , we consider binary variables $x_i \in \{0,1\}$ only. The range of the function is thus $\mathcal{R}=2^n$. Let $\mathcal{M}$ be the set of optima (the optimum need not be unique).

All stochastic population based optimization methods can be expressed by a probability model, where a probability distribution $p(x)$ describes the distribution of the individuals in the search space $\mathcal{R}$. Mutation, crossover and selection then become operators that transform these probability distributions. Important for us are the marginal and conditional probabilities of the distribution. The marginal distributions are defined as

$$p(x_i) := p_i(x_i) = \sum_{y|y_i=x_i} p(y) \tag{1}$$

This definition can be naturally extended to sets of variables as $p(x_i,x_j)$. Sometimes we will also write $p(x_i=1)$ to denote the marginal frequency $p_i(1)$. When $p(x_j) > 0$, we can define the conditional probability

$$p(x_i|x_j) = \frac{p(x_i,x_j)}{p(x_j)} \tag{2}$$

Instead of determining the distributions generated by a genetic algorithm, we can also choose a distribution first and then try to find the matching algorithm. A good candidate for optimization is the Boltzmann distribution.

3 The Boltzmann distribution

Definition 1. *For $\beta \geq 0$ define the **Boltzmann distribution** of a function $f(x)$ as*

$$p_{\beta,f}(x) := \frac{e^{\beta f(x)}}{\sum_y e^{\beta f(y)}} \qquad =: \frac{e^{\beta f(x)}}{Z_f(\beta)} \tag{3}$$

where $Z_f(\beta)$ is the partition function. To simplify the notation β and/or f can be omitted.

The Boltzmann distribution is usually defined as $e^{-\frac{g(x)}{T}}/Z$. The term $g(x)$ is called the free energy and $T = 1/\beta$ the temperature. The Boltzmann distribution has a number of properties, among them

Lemma 1. *Let $x_m \in \mathcal{M}$ be a global optimum of the function $f(x)$ and x_l a point with $f(x_l) < f(x_m)$. Then*

- *$p_{\beta=0,f}$ is the uniform distribution for any f.*
- *$p_\beta(x_m) \geq p_\beta(x_l)$, for $\beta > 0$ the inequality is strict.*
- *Let $g(x) := f(x) + c$. Then $p_{\beta,f}(x) = p_{\beta,g}(x)$.*
- *Let $g(x) := c \cdot f(x)$. Then $p_{\beta,g}(x) = p_{c\beta,f}(x)$.*

The third property means that the distribution is invariant under addition of a constant. It is, however, not invariant under multiplication. We will discuss how to overcome this shortcoming in section 5.

The Boltzmann distribution is a suitable distribution for optimization because it concentrates its weight with increasing β around the global optima of the function. In theory, if it were possible to sample efficiently from this distribution for arbitrary β, optimization would be trivial.

3.1 Boltzmann selection

Closely related to the Boltzmann distribution is Boltzmann selection:

Definition 2. *Given a distribution p and a selection parameter γ, **Boltzmann selection** calculates the distribution of the selected points according to*

$$p^s(x) = \frac{p(x)e^{\gamma f(x)}}{\sum_y p(y)e^{\gamma f(y)}} \tag{4}$$

Boltzmann selection is important because of the following theorem:

Theorem 1. *Let $p_\beta(x)$ be a Boltzmann distribution. If Boltzmann selection is used with parameter γ, then the distribution of the selected points is again a Boltzmann distribution with*

$$p^s(x) = \frac{e^{(\beta+\gamma)f(x)}}{\sum_y e^{(\beta+\gamma)f(y)}}. \tag{5}$$

The (simple) proof can be found in [MMO99].

This allows us to define the *BEDA* (Boltzmann Estimated Distribution Algorithm).

BEDA is a conceptional algorithm, because the calculation of the distribution requires a sum over exponentially many terms. We have proven the following important convergence theorem for it:

Theorem 2 (Convergence). *Let $\Delta\beta(t)$ be an annealing schedule, i.e. for every t an increase in the inverse temperature β by $\Delta\beta(t)$. Then for BEDA the distribution at time t is given by*

$$p(x,t) = \frac{e^{\beta(t)f(x)}}{Z_f(\beta(t))} \tag{6}$$

with the inverse temperature $\beta(t) = \sum_{\tau=1}^{t} \Delta\beta(\tau)$. If $\beta(t) \to \infty$, then with $\mathcal{M}$ the set of optima

$$\lim_{t\to\infty} p(x,t) = \begin{cases} 1/|\mathcal{M}| & x \in \mathcal{M} \\ 0 & \textit{else} \end{cases} \tag{7}$$

***BEDA* – Boltzmann Estimated Distribution Algorithm**

```
t ⇐ 0. Generate $N$ points according to the uniform
  distribution $p(x,0)$ with $\beta(0) = 0$.
do {
    With a given $\Delta\beta(t) > 0$, let
        $p^s(x,t) = \frac{p(x,t)e^{\Delta\beta(t)f(x)}}{\sum_y p(y,t)e^{\Delta\beta(t)f(y)}}$.
    Generate $N$ new points according to the distribution $p(x,t+1) = p^s(x,t)$.
    $t \Leftarrow t+1$.
} until (stopping criterion reached)
```

Proof: Let $x^m \in \mathcal{M}$ be a point with maximal fitness and $x \notin \mathcal{M}$ a point with $f(x) < f(x^m)$. Then

$$\begin{aligned} p(x,t) &= \frac{e^{\beta(t)f(x)}}{\sum_y e^{\beta(t)f(y)}} \leq \frac{e^{\beta(t)f(x)}}{|\mathcal{M}| \cdot e^{\beta(t)f(x^m)}} \\ &\leq \frac{1}{|\mathcal{M}| \cdot e^{\beta(t)[f(x^m)-f(x)]}} \end{aligned} \tag{8}$$

As $\beta(t) \to \infty$, $p(x,t)$ converges to 0. Because $p(x,t) = p(y,t)$ for all $x^m, y^m \in \mathcal{M}$, the limit distribution is the uniform distribution on the set of optima. ■

We can also make an estimate for the rate of convergence:

Lemma 2. *Let there be a δ such that for any non-optimal point x we have with $x^m \in \mathcal{M}$*

$$f(x) \leq f(x^m) - \delta \tag{9}$$

Then

$$\beta \geq \frac{n \cdot \ln 2}{\delta} \quad \Longrightarrow \quad p_\beta(\mathcal{M}) \geq 0.5. \tag{10}$$

This means that when the indicated inverse temperature is reached, drawing from p_β will generate global optima with probability bigger than $1/2$.

Proof: Let $|\mathcal{M}|$ be the number of optima. The number of terms in the partition function is smaller than 2^n. For $x^m \in \mathcal{M}$ we have with $M := f(x^m)$

$$\begin{aligned} p_\beta(x^m) &= \frac{e^{\beta M}}{\sum_y e^{\beta f(y)}} \geq \frac{e^{\beta M}}{2^n \cdot e^{\beta(M-\delta)} + |\mathcal{M}| \cdot e^{\beta M}} \\ &= \frac{1}{e^{n \ln 2 - \beta\delta} + |\mathcal{M}|} \overset{!}{\geq} \frac{1}{2|\mathcal{M}|}, \end{aligned} \tag{11}$$

So, to have $p_\beta(\mathcal{M}) \geq {}^1\!/_2$, we need

$$e^{n \ln 2 - \beta\delta} \leq 2|\mathcal{M}| \;\Leftrightarrow\; \beta \geq \frac{n \cdot 2 - \ln(2|\mathcal{M}|)}{\delta} \tag{12}$$

or as a sufficient condition (10). ■

Corollary 1. *For a binary fitness function with integer values, with $\beta \geq 0.7n$ half of the generated points will have maximum fitness, independent of the fitness function.*

Without a schedule the corollary doesn't tell us very much, as this value of β can be reached in any number of steps. It can be used with fixed schedules, however, and as a stopping criterion.

4 Factorization of the distribution

In this section we describe a method for computing a factorization of the probability, given an additive decomposition of the function:

Definition 3. *Let $s_1, \ldots, s_m$ be index sets, $s_i \subseteq \{1, \ldots, n\}$. Let f_{s_i} be functions depending only on the variables x_j with $j \in s_i$. Then*

$$f(x) = \sum_{i=1}^{m} f_{s_i}(x) \tag{13}$$

*is an **additive decomposition** of the fitness function f.*

We also need the following definitions

Definition 4. *Given $s_1, \ldots, s_m$, we define for $i = 1, \ldots, m$ the sets d_i, b_i and c_i:*

$$d_i := \bigcup_{j=1}^{i} s_j, \quad b_i := s_i \setminus d_{i-1}, \quad c_i := s_i \cap d_{i-1} \tag{14}$$

We set $d_0 = \emptyset$.

In the theory of decomposable graphs, d_i are called *histories*, b_i *residuals* and c_i *separators* [Lau96]. In [MMO99], we have shown the following important

Theorem 3 (Factorization Theorem). *Let $p(x)$ be a Boltzmann distribution with*

$$p(x) = \frac{e^{\beta f(x)}}{Z_f(\beta)} \tag{15}$$

and $f(x) = \sum_{i=1}^{m} f_{s_i}(x)$ be an additive decomposition. If

$$b_i \neq \emptyset \quad \forall i = 1, \ldots, l; \quad d_l = \tilde{X}, \tag{16}$$

$$\forall i \geq 2 \; \exists j < i \;\; \text{such that } c_i \subseteq s_j \tag{17}$$

then

$$p(x) = \prod_{i=1}^{m} p(x_{b_i} | x_{c_i}) \tag{18}$$

***FDA* – Factorized Distribution Algorithm**

```
Calculate $b_i$ and $c_i$ from the decomposition of the function.
Generate an initial population with $N$ individuals from the uniform distribution.
do {
    Select $\hat{N} \leq N$ individuals using Boltzmann selection.
    Estimate the conditional probabilities $p(x_{b_i}|x_{c_i}, t)$ from the selected points.
    Generate new points according to $p(x, t+1) = \prod_{i=1}^{m} p(x_{b_i}|x_{c_i}, t)$.
    $t \Leftarrow t + 1$.
} until (stopping criterion reached)
```

The constraint defined as equation (17) is called the running intersection property [Lau96].

With the help of the factorization theorem, we can turn the conceptional algorithm *BEDA* into *FDA*, the Factorized Distribution Algorithm [MMO99].

As the factorized distribution is identical to the Boltzmann distribution if the conditions of the factorization theorem are fulfilled, the convergence proof of *BEDA* also applies to *FDA*.

Not every additive decomposition leads to a factorization using the factorization theorem. In these cases, more sophisticated methods have to be used. *FDA* can also be used with an approximate factorization. In [MM00], we have used Bayesian networks to describe the dependencies and used the *minimum description length* [FG99] to calculate the network and thus the factorization from the data, i.e. without prior knowledge of a decomposition. This algorithm is called the *LFDA* (Learning Factorized Distribution Algorithm).

FDA can also be used with any other selection scheme, but then the convergence proof is no longer valid. We think that Boltzmann selection is an essential part in using the *FDA*. This is even more true for the *LFDA*, because the theory of learning the Bayesian network tries to gather independencies from the data points. But these dependencies are exactly the same as the ones implied by the additive decomposition of the function defining the Boltzmann distribution [Lau96]. That means, if we start with a function fulfilling the factorization theorem and generate points, then the learned model will (with enough data points) lead to the same factorization as the factorization theorem. But this is only true using Boltzmann selection.

Because *FDA* uses finite samples of points to estimate the conditional probabilities, convergence to the optimum will depend on the population size.

Note that the *UMDA* (univariate marginal distribution algorithm) [Müh98] is the same as *FDA* using the distribution $p(x) = \prod_{i=1}^{n} p_i(x_i)$ regardless of the fitness function. For *UMDA*, the convergence proof is therefore only valid for linear functions.

5 A new annealing schedule for the Boltzmann distribution

Boltzmann selection needs an annealing schedule. Lemma 2 has shown how fast we have to anneal in order to reach convergence within a given time frame. But if we anneal too fast, the approximation of the Boltzmann due to the sampling error can be very bad. For an extreme case, if the annealing parameter is very large, the second generation should consist only of the global maxima.

In order to control the annealing schedule, we make a Taylor expansion of the average fitness of the Boltzmann distribution.

5.1 Taylor expansion of the average fitness

For *UMDA*, we have shown that the average fitness completely determines the behaviour of the algorithm [MM00]. In reference to S. Wright it is labelled W_f:

Definition 5. *The **average fitness** of a fitness function and a distribution is*

$$W_f(p) = \sum_x f(x)p(x) \tag{19}$$

For the Boltzmann distribution, we use the abbreviation $W_f(\beta) := W_f(p_{\beta,f})$.

Theorem 4. *The average fitness of the Boltzmann distribution $W_f(\beta)$ has the following expansion in β:*

$$W_f(\tilde{\beta}) = W_f(\beta) + \sum_{i \geq 1} \frac{(\tilde{\beta} - \beta)^i}{i\,!} M_{i+1}^c(\beta) \tag{20}$$

where M_i^c are the centred moments

$$M_i^c(\beta) := \sum_x \left[f(x) - W_f(\beta)\right]^i p(x) \tag{21}$$

They can be calculated using the derivatives of the partition function:

$$M_{i+1}^c(\beta) = \left(\frac{Z_f'(\beta)}{Z_f(\beta)}\right)^{(i)} \qquad \textit{for } i \geq 1, \quad M_1^c = 0 \tag{22}$$

The proof can be found in [MM01].

Corollary 2. *We have approximately*

$$W_f(\tilde{\beta}) \approx W_f(\beta) + (\tilde{\beta} - \beta) \cdot \sigma_f^2(\beta) \tag{23}$$

where $\sigma_f^2(\beta)$ is the variance of the distribution, defined as $\sigma_f^2(\beta) := M_2^c(\beta)$.
This approximation is known from the theory of simulated annealing, see [KGV83].

Lemma 3. *The variance of the Boltzmann distribution obeys*

$$f(x) \neq \textit{const.} \implies \sigma_f^2(\beta) > 0 \tag{24}$$

Proof: We have $\forall x : p_\beta(x) > 0$. In order to have

$$\sigma_f^2(\beta) = \sum_x \left[f(x) - W_f(\beta)\right]^2 p_\beta(x) \overset{!}{=} 0, \tag{25}$$

we must have for all x: $f(x) = W_f$ in contradiction to the assumption. ■

Corollary 3. *With $f(x) \neq$ const. we have*

$$\tilde{\beta} > \beta \implies W_f(\tilde{\beta}) > W_f(\beta) \tag{26}$$

This important corollary tells us that the average fitness is never decreasing for Boltzmann selection. A similar result was already obtained for proportional selection, see for example [MM00].

5.2 The new annealing schedule

From (23) we can derive an adaptive annealing schedule. The variance (and the higher moments) can be estimated from the generated points. As long as the approximation is valid, one can choose a desired increase in the average fitness and set $\beta(t+1)$ accordingly. So we can set

$$\Delta\beta(t) := \beta(t+1) - \beta(t) = \frac{W_f^{\text{new}}(t) - W_f(\beta(t))}{\sigma_f^2(\beta(t))}$$

From (23) we see that choosing $\Delta\beta$ proportional to the inverse of the variance leads in the approximation to a constant increase in the average fitness. This is much too fast, especially near the optimum. As truncation selection has proven to be a robust and efficient selection scheme, we can try to approximate the behaviour of this method. For truncation selection, one can show that the *response to selection $R_f(t)$* is approximatively given by [Müh98]

$$\begin{aligned} R_f(t) &:= W_f\big(\beta(t+1)\big) - W_f\big(\beta(t)\big) \\ &\approx I_\tau b \sqrt{\sigma_f^2} \end{aligned} \tag{27}$$

I_τ is the selection intensity, depending on the truncation threshold τ, and b is called heritability. Therefore, we will use a schedule proportional to the inverse of the square root of the variance:

Lemma 4. $\Delta\beta(t) = c/\sqrt{\sigma_f^2(\beta(t))}$ *leads to an annealing schedule where the average fitness increases approximatively proportional to the standard deviation:*

$$R_f(t) = W_f\big(\beta(t+1)\big) - W_f\big(\beta(t)\big) \tag{28}$$

$$\approx c \cdot \sqrt{\sigma_f^2(\beta(t))} \tag{29}$$

*This annealing schedule is called SDS, the **standard deviation schedule**.*

We already know that *FDA* with Boltzmann selection remains unchanged when we add a constant to the fitness function. Now we have additionally

Lemma 5. *For Boltzmann selection with SDS, BEDA is invariant under linear transformations of the fitness function with a positive factor.*

Proof: This lemma is true because the standard deviation scales linearly under multiplication. Let $f(x)$ be a fitness function, consider $\hat{f}(x) = \hat{c} \cdot f(x)$. The claim is that $\hat{\beta}(t) = \beta(t)/\tilde{c}$, then the distributions are the same for every t. With $t=0$, β and $\hat{\beta}$ are 0, so it is true. Let now t and $\beta=\beta(t)$ be given. From the previous iteration we know that $\hat{\beta} = \beta/\hat{c}$.

According to lemma 1, we have $p_{\beta,f}(x) = p_{\hat{\beta},\hat{f}}(x)$. Also, $\sigma^2_f(\beta) = \hat{c}^2 \cdot \sigma^2_{\hat{f}}(\hat{\beta})$. Hence it follows that $\Delta\hat{\beta}(t) = \Delta\beta(t)/\hat{c}$. ■

Corollary 4. *Let $\sigma(t) := \sqrt{\sigma^2_f(\beta(t))}$, the standard deviation. Then the response to selection for Boltzmann selection with the SDS is given by*

$$R_f(t) = \sum_{i\geq 1} \frac{c^i}{i!\,\sigma(t)^i} M^c_{i+1} \tag{30}$$

$$= c \cdot \sigma(t) + \frac{c^2 M^c_3}{2\,\sigma(t)^2} + \frac{c^3 M^c_4}{6\,\sigma(t)^3} + \ldots \tag{31}$$

Note that this annealing schedule cannot be easily used for simulated annealing, as the estimation of the variance of the distribution requires samples that are independently drawn. But the sequence of samples generated by simulated annealing are not independent.

6 Examples

Truncation selection has proven to be an efficient and robust selection scheme. It is therefore interesting to study the differences between truncation selection and Boltzmann selection. Note that the computational complexity of both selections schemes is $O(N \log N)$.

6.1 The function *OneMax*

For *OneMax* we can calculate the Boltzmann distribution. We have

$$Z_f(\beta) = \sum_x e^{\beta|x|} = \sum_{i=0}^{n} \binom{n}{i} \left(e^\beta\right)^i = (1+e^\beta)^n \tag{32}$$

with $|x| = \sum_{i=1}^n x_i$. All marginal distributions remain the same if started from the uniform distribution.

$$Z_f(\beta) \cdot p_\beta(x_1=1) = \sum_{x,x_1=1} e^{\beta|x|} = \sum_{i=0}^{n-1} \binom{n-1}{i} e^{\beta(i+1)}$$
$$= e^\beta \cdot (1+e^\beta)^{n-1}$$

and therefore

$$p_\beta(x_1=1) = \frac{e^\beta}{1+e^\beta} =: \hat{p}_\beta \tag{33}$$

With theorem 4 the average fitness $W_f(\beta)$ can be calculated using $W_f(\beta) = Z'_f(\beta)/Z_f(\beta)$:

$$W_f(\beta) = \frac{n(1+e^\beta)^{n-1}e^\beta}{(1+e^\beta)^n} = \frac{ne^\beta}{1+e^\beta} \quad \left(= n \cdot \hat{p}_\beta\right) \tag{34}$$

The variance is given by (22):

$$\sigma^2_f(\beta) = \frac{Z''_f(\beta)}{Z_f(\beta)} - \left(\frac{Z'_f(\beta)}{Z_f(\beta)}\right)^2 = \frac{ne^\beta}{(1+e^\beta)^2} \tag{35}$$

We get the following difference equation for the inverse temperature:

$$\beta(t+1) = \beta(t) + c \cdot \frac{1+e^{\beta(t)}}{\sqrt{n} \cdot e^{\beta(t)/2}} \tag{36}$$

Higher moments can also be calculated, the response can be approximated by

$$R_f(t) = c\frac{e^{\beta/2}\sqrt{n}}{1+e^\beta} - c^2\frac{(e^\beta-1)}{2(e^\beta+1)} + \ldots \tag{37}$$

The annealing schedule uses only the first term in this expansion, so we have

$$R_f(t) \approx c\frac{e^{\beta/2}\sqrt{n}}{1+e^\beta} \tag{38}$$

For a closed (approximative) solution we can convert the difference equation into a differential equation and get

$$\frac{d\beta}{dt} = \frac{c}{\sigma(t)} = \frac{2c}{\sqrt{n}}\cosh\bigl(\beta(t)/2\bigr) \tag{39}$$

The solution of the differential equation with $\beta(0) = 0$ yields

$$\beta(t) = 4\,\mathrm{argtanh}\left[\tan\left(\frac{c \cdot t}{2\sqrt{n}}\right)\right] \quad \text{with } t \leq \frac{\pi\sqrt{n}}{2c} \tag{40}$$

Together with equation (33) we get

$$\hat{p}_\beta(t) = \frac{1}{2}\left(1 + \sin\left(\frac{c \cdot t}{\sqrt{n}}\right)\right) \tag{41}$$

If we compare this to previous results obtained for truncation selection [Müh98], we see that the dynamic equation (41) is exactly the same. The constant c now plays the role of I_τ. So from the theory of truncation selection, we can get a suitable range for c, namely $c \in [0.8, 1.3]$.

In figure 1, this theoretical result is compared to a simulation run. The simulation was done using a population size of 30000 individuals to get a good statistics, $c=1$, $n=16$. The standard deviation needed for the schedule was estimated from the population. The differences between theory and simulation at the end are due to the simplification of using a differential equation instead of the difference equation. The theoretical curve is of course the same as the one for truncation selection.

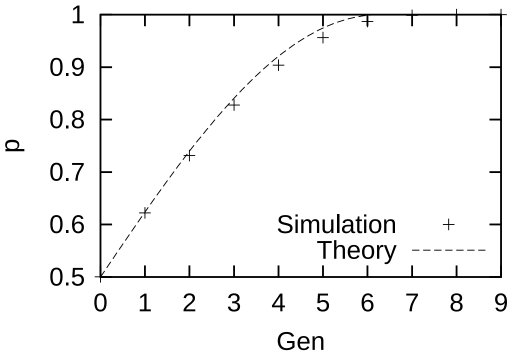

Figure 1: Univariate marginal distribution for *OneMax* with Boltzmann selection, $n = 16$

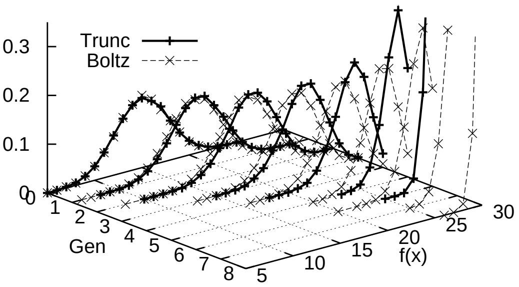

Figure 2: Fitness distribution for *OneMax*: Truncation selection and Boltzmann selection lead to similar distributions.

In figure 2 we can see the distributions of the fitness values for *OneMax* and the two selection schemes with $n{=}30$ and a populations size of 10000 individuals. We used corresponding selection intensity $\tau = .3$ and constant $c{=}1.159$. As expected, the two distributions are very similar. Only in the end there are distortions.

So for *OneMax*, despite a completely different mechanism for selecting individuals, the theoretical results for infinite population size are the same.

6.2 Kauffman's $n - k$ model

As another example for the similarity between the two selection schemes we consider a function from Kauffman's $n - k$ model [KL87]. These functions consist of n binary variables and n sub-functions f_{s_i} depending on x_i and k further variables. The values of the sub functions are chosen uniformly random from the interval $[0, 1]$:

$$\mathit{NK}_{n,k}(x) = \frac{1}{n}\sum_{i=1}^{n} f_{s_i}(x) \quad |s_i|{=}k+1,\ i \in s_i \qquad (42)$$

In figures 3 and 4 we can see distributions of the fitness values for truncation selection and Boltzmann selection with $n{=}30$, $\tau = 0.3$ resp. $c{=}1.159$, $k{=}2$ with *adjacent* neighbours, i.e. $s_i{=}\{i-1, i, i+1\}$. In this case, the distribution factorizes. As was the case for *OneMax*, the distributions remain similar.

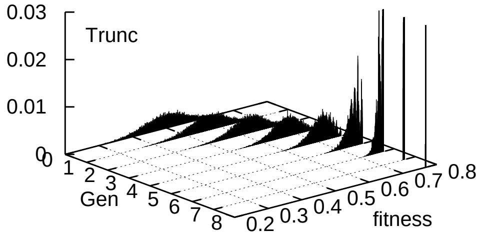

Figure 3: Fitness distribution for *NK*, truncation selection

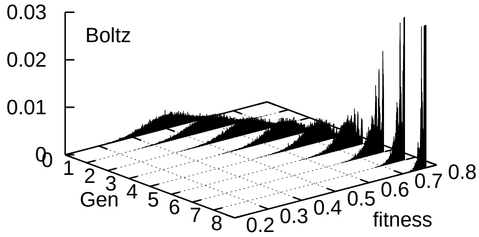

Figure 4: Fitness distribution for *NK*, Boltzmann selection

6.3 The *Jump* function

Next we consider the following *Jump* function.

$$\mathit{Jump}_{n,k}(x) = \begin{cases} |x| & |x| \le n-k \text{ or } |x| = n \\ H - |x| & \text{else} \end{cases} \qquad (43)$$

where $H := 2(n-k)$ and k is a positive integer, the gap size. In figure 5 is a typical example with $n{=}32$ and $k{=}4$.

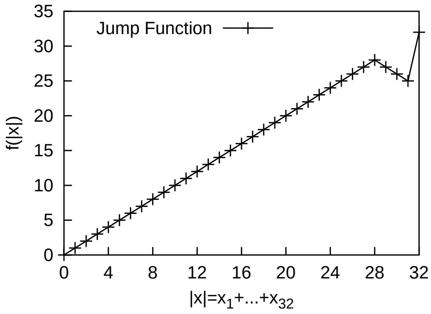

Figure 5: Definition of the *Jump* function with $n = 32$ and $k = 4$.

The function starts like *OneMax*, but near the optimum the fitness decreases. Optimisation algorithms have to cross the gap in order to reach the global optimum. Theoretical analysis of the Boltzmann distribution is complicated, therefore we show only empirical results.

We cannot use the *FDA* for this function, because even with a gap width $k{=}1$ it does not factorize. But we can use

UMDA. In [MM00], we have shown that *UMDA* operates on the landscape of the average fitness, parameterized by the univariate marginal distributions (bit frequencies). This landscape smoothes the discrete optimization problem into a continuous one. The details are out of the scope of this paper.

It is important to note that the convergence proof of *BEDA* does not apply here, as the probability distribution used is *not* the Boltzmann distribution, because the factorization is not valid. Nonetheless it is possible to use Boltzmann selection.

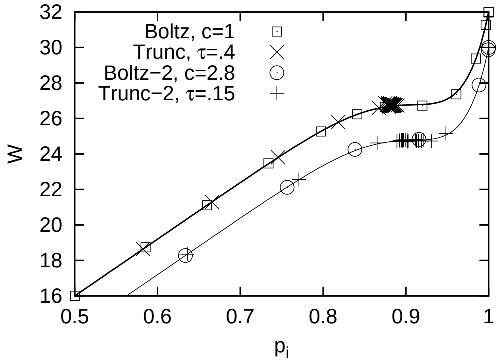

Figure 6: Comparison of Boltzmann selection and truncation selection for *Jump*. Two graphs are shifted 2 down for readability.

In figure 6, this smoothing together with four runs of the *UMDA* are shown. Plotted are the average fitness against the average bit frequency p with a population size $N = 1000$. The solid line shows the theoretical values calculated by setting all frequencies p_i to p. This is justified by the fact that the problem is symmetrical in the variables, see [MM00].

The values of τ and c were chosen to correspond to roughly the same behaviour in the beginning. One can see that truncation selection with $\tau = 0.4$ is not able to cross the local optimum. Only with a much higher selection threshold of $\tau = 0.15$ does this happen. Boltzmann selection, on the other hand, is not deceived in this case. It seems to be able to adapt better to this situation.

Note that this comes at a price: in order to have a good estimate for the standard deviation, we usually need a higher population size than is needed for truncation selection. Also, due to the exponential growth, this method is more sensitive to fixation, because a single good individual can dominate the population more easily than with truncation selection: there the weights are all equal among the selected set.

As the speed of convergence is so different, we have not plotted the evolution of the fitness distributions.

6.4 Comparison of population size and number of generations for several test functions

Instead of comparing the distributions, we can also directly compare the performance of the two selection methods, i.e. the success rate for a given population size and the number of generations. This is done in table 1. n is the bit length of the problem, N the population size, Succ the number of times the optimum was found in 100 runs, Gen1 the number of generations till the optimum was generated for the first time (only the successful runs were counted), Gen the number of generations until the population converged and σ the standard deviation of this value. For a discussion of the scaling of the algorithms see [MM99].

The analysed functions are the following. We have already introduced *OneMax*,*NK* and *Jump*. For *NK*, we used this time 100 different instances with $k = 2$ and again the neighbours were chosen adjacent. In this case, the global optimum can be easily calculated using dynamic programming. Each of the 100 runs was thus faced with a different problem.

We used only one instance of the *Jump* function, because doubling the bit length without adjusting the gap size leads to a much simpler problem. The same is true for the *Saw* function which can be seen in figure 7.

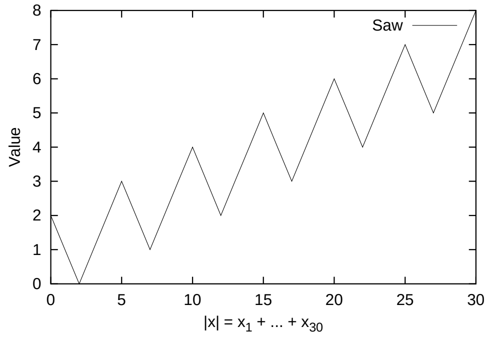

Figure 7: Definition of the *Saw* function.

Dec is the deceptive function. It is the sum of several subfunctions shown, for example with 32 bits

$$Dec(x_1, \ldots, x_{32}) = \sum_{i=1}^{8} Dec_4(x_{4i-3}, x_{4i-2}, x_{4i-1}, x_{4i})$$

where Dec_4 is defined in table 2. The function has the global maximum when all bits are 1, but local information points towards 0.

$\lvert x\rvert$	0	1	2	3	4
$Dec(\lvert x\rvert)$	3	2	1	0	4

Table 2: Definition of Dec_4

IsoC, finally, is defined as

$$IsoC(x) = \sum_{i=1}^{m-1} IsoC_1(x_{2i-1}, x_{2i}) + IsoC_2(x_{2m-1}, x_{2m})$$

with

x	00	01	10	11
$IsoC_1$	m	0	0	$m-1$
$IsoC_2$	0	0	0	m

and $n = m + 1$. The global optimum is again $(1, 1, \ldots, 1)$ with value $m \cdot (m-1) + 1$. This optimum is triggered by

		Trunc, $\tau = 0.3$					Boltz, $c = 1.159$				
Func	n	N	Succ	Gen1	Gen	σ	N	Succ	Gen1	Gen	σ
OneMax	32	40	89/100	5.2	7.5	0.64	40	89/100	7.0	12.6	1.54
OneMax	64	60	91/100	8.4	10.9	0.65	65	89/100	10.4	16.6	1.87
Jump	32	80	96/100	5.0	30.0	0.00	80	99/100	6.4	22.3	8.18
Saw	30	120	94/100	8.5	23.9	6.40	120	97/100	9.4	23.7	5.01
Dec	32	140	90/100	4.6	7.7	0.75	160	87/100	5.0	12.0	2.13
Dec	64	280	98/100	7.5	10.6	0.60	320	97/100	8.3	15.4	1.38
NK	32	220	92/100	5.3	9.2	0.93	250	90/100	6.5	28.1	3.80
NK	64	400	96/100	9.3	13.1	0.97	500	92/100	11.4	30.0	0.00
IsoC	32	400	95/100	4.2	8.7	0.89	350	95/100	5.1	14.7	1.48
IsoC	64	1200	94/100	7.3	13.0	1.00	1000	94/100	8.4	19.4	1.62

Table 1: Success rates and generations with population size N for different problems with bit length n. The program was stopped after 30 generations if not convergenced.

IsoC$_2$ and strongly isolated. The second best value occurs several times, but only with individuals with leading zeroes, for example $(0,\ldots,0)$ with value $m \cdot (m-1)$. All of these good points are very far away from the global optimum in the fitness landscape.

We used exact factorizations for all cases except the *Jump* function and the *Saw*, where *UMDA* was used.

From table 1 several observations can be made. First of all, the populations sizes needed for a given success rate and the number of generations were similar for both selection methods. The number of generations until the optimum was generated was about 20% higher for the whole range of functions considered. This gap was larger when time to convergence was considered. This means that truncation selection is faster at the very end. Note that in some cases, the algorithms did not converge at all. This can be seen from average generation numbers greater than 20, because the algorithms were stopped in generation 30. The population size needed for about 90% success was higher in some and lower in others like the *Jump* function (as predicted from section 6.3).

7 Conclusions

FDA has been shown to be an efficient optimization algorithms when interactions between variables have to be considered to reach the global optimum. The convergence proof of *FDA* requires that Boltzmann selection is used. But Boltzmann selection critically depends on a good annealing schedule. Therefor we have previously used truncation selection. We have now invented an adaptive annealing schedule *SDS* that leads to an optimization algorithm that is almost as robust as truncation selection and for which the convergence proof remains valid. We have seen in several examples that the behaviour of Boltzmann selection is similar to truncation selection.

Boltzmann selection with the *SDS* has proven to be comparable in computational complexity, robustness and efficiency to truncation selection and we have a convergence proof for it for exact factorizations.

Bibliography

[FG99] N. Friedman and M. Goldzmidt. Learning bayesian networks with local structure. In M.I. Jordan, editor, *Learning in Graphical Models*, pages 421–459. MIT Press, Cambrigde, 1999.

[KGV83] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[KL87] St.A. Kauffman and S. Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128:11–45, 1987.

[Lau96] St. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

[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.

[MM00] H. Mühlenbein and Th. Mahnig. Evolutionary algorithms: From recombination to search distributions. In L. Kallel, B. Naudts, and A. Rogers, editors, *Theoretical Aspects of Evolutionary Computing*, Natural Computing, pages 137–176. Springer Verlag, 2000.

[MM01] Th. Mahnig and H. Mühlenbein. A new adaptive boltzmann selection schedule SDS. In *Proceedings of the 2001 Congress on Evolutionary Computation*, 2001. submitted for publication.

[MMO99] H. Mühlenbein, Th. Mahnig, and A. Rodriguez Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):215–247, 1999.

[Müh98] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5:303–346, 1998.