A New Adaptive Boltzmann Selection Schedule *SDS*

Thilo Mahnig **Heinz Mühlenbein**
GMD – Schloss Birlinghoven
53757 Sankt Augustin
Germany
{mahnig,muehlen}@gmd.de

Published in Proceedings of the Congress on Evolutionary Computation 2001, pp 183-190

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. In general a distribution defined for n binary variables has 2^n parameters. Therefore it is too expensive to compute. For additively decomposed discrete functions (ADFs) there exists an algorithm that factors the distribution into conditional and marginal distributions, each of which can be computed 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* (standard deviation schedule) that is introduced in this work. The inverse temperature β is changed inversely proportional to the standard deviation.

Keywords: *genetic algorithms, simulated annealing, Boltzmann distribution, Boltzmann selection*

1 Introduction

It is well known that certain 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.

As all stochastic population-based algorithms can be described with probability distributions, we have introduced a generic framework for optimization using distributions. Simple genetic algorithms can then be described using a simple product distribution.

In previous work [MMO99], we 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 used often because, similarly to simulated annealing, it needs an annealing schedule, which is difficult to choose.

In this work, we make a Taylor expansion of the average fitness of the Boltzmann distribution. This 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 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. Finally, in section 6, we discuss two examples of the theory.

2 Optimization by distributions

Our goal is to optimize (maximize) a discrete function $f(x)=f(x_1,\ldots,x_n)$, called the fitness function. For notational simplicity, we consider binary variables $x_i \in \{0,1\}$ only. The domain 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 [CF99]. 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 extended naturally to sets of variables like $p(x_i,x_j)$. 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

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

EDA, the Estimation of Distributions Algorithm, describes a general framework for this type of algorithms. First we generate a population of search points using the uniform distribution. Then we perform selection with these points, based on the fitness (=function) value. In the next step, a distribution is estimated from the search points. This is usually an approximation step, as an arbitrary distribution needs 2^n

***EDA* – Estimation of Distributions Algorithm**

```
Generate an initial population with $N$ individuals using the uniform distribution, $t \Leftarrow 1$.
do {
    Select $\hat{N} \leq N$ points.
    Estimate the distribution $p^s(x,t)$ from the selected points.
    Generate new points according to $p(x,t+1) = p^s(x,t)$.
    $t \Leftarrow t+1$.
} until (stopping criterion reached)
```

parameters. Finally, new points are generated from this distribution and the process is iterated.

We have shown that a simple genetic algorithm can be approximated by the *UMDA* (univariate marginal distribution algorithm) [MM00]. The *UMDA* is an *EDA* using the simple product distribution

$$p(x) = \prod_{i=1}^{n} p_i(x_i). \tag{3}$$

EDA needs a selection method. This is also the step where the fitness function is involved. In principle, any selection method from the history of genetic algorithms can be used, like proportionate selection or truncation selection. But the underlying distribution should also be considered when choosing the selection method, as we will show in the next section.

3 The Boltzmann distribution

The simple product distribution of *UMDA* cannot capture dependencies between variables. When the exploitation of the dependencies is required to find a global optimum, *UMDA* and simple genetic algorithms fail. We need a more complex distribution to reach this goal. A good candidate is the Boltzmann distribution.

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

$$p_{\beta,f}(x) := \frac{e^{\beta f(x)}}{\sum_y e^{\beta f(y)}} \qquad =: \frac{e^{\beta f(x)}}{Z_f(\beta)} \tag{4}$$

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 [dlMT93]:

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

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

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*

$$p^s(x) = \frac{e^{(\beta+\gamma)f(x)}}{\sum_y e^{(\beta+\gamma)f(y)}}. \tag{6}$$

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

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

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

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

BEDA is a conceptual 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 the difference $\Delta\beta(t)$ between consecutive inverse temperature values β. Then for BEDA the distribution at time t is given by*

$$p(x,t) = \frac{e^{\beta(t)f(x)}}{Z_f(\beta(t))} \tag{7}$$

with the inverse temperature

$$\beta(t) = \sum_{\tau=1}^{t} \Delta\beta(\tau). \tag{8}$$

If $\beta(t) \to \infty$, then

$$\lim_{t\to\infty} p(x,t) = \begin{cases} 1/|\mathcal{M}| & x \in \mathcal{M} \\ 0 & \textit{else} \end{cases} \tag{9}$$

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

$$\begin{aligned} p(x,t) &= \frac{e^{\beta(t)f(x)}}{\sum_y e^{\beta(t)f(y)}} \leq \frac{e^{\beta(t)f(x)}}{|\mathcal{M}| \cdot e^{\beta(t)f(x^m)}} \\ &\leq \frac{1}{|\mathcal{M}| \cdot e^{\beta(t)[f(x^m)-f(x)]}} \end{aligned} \tag{10}$$

As $\beta(t) \to \infty$, $p(x,t)$ converges (exponentially fast) 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. ■

Equation (10) only shows that the distribution converges to 0 for non-optimal points. But 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 $x^m \in \mathcal{M}$*

$$f(x) \leq f(x^m) - \delta \tag{11}$$

Then

$$\beta \geq \frac{n \cdot \ln 2}{\delta} \quad \Longrightarrow \quad p_\beta(\mathcal{M}) \geq 0.5. \tag{12}$$

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 $M := f(x^m)$

$$\begin{aligned} p_\beta(x^m) &= \frac{e^{\beta M}}{\sum_y e^{\beta f(y)}} \geq \frac{e^{\beta M}}{2^n \cdot e^{\beta(M-\delta)} + |\mathcal{M}| \cdot e^{\beta M}} \\ &= \frac{1}{e^{n \ln 2 - \beta\delta} + |\mathcal{M}|} \quad \overset{!}{\geq} \frac{1}{2|\mathcal{M}|}, \end{aligned} \tag{13}$$

So, to have $p_\beta(\mathcal{M}) \geq 1/2$, we need

$$e^{n \ln 2 - \beta\delta} \leq 2|\mathcal{M}| \;\Leftrightarrow\; \beta \geq \frac{n \cdot 2 - \ln(2|\mathcal{M}|)}{\delta} \tag{14}$$

or as a sufficient condition (12). ■

Corollary 1. *For a binary fitness function with integer values (so $\delta = 1$), 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 explain 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 each f_{s_i} be a function depending only on the variables x_j with $j \in s_i$. Then*

$$f(x) = \sum_{i=1}^{m} f_{s_i}(x) \tag{15}$$

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:*

$$d_i := \bigcup_{j=1}^{i} s_j, \quad b_i := s_i \setminus d_{i-1}, \quad c_i := s_i \cap d_{i-1} \tag{16}$$

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*

$$p(x) = \frac{e^{\beta f(x)}}{Z_f(\beta)} \tag{17}$$

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

$$b_i \neq \emptyset \quad \forall i = 1, \ldots, l; \quad d_l = \{1, \ldots, n\} \tag{18}$$

$$\forall i \geq 2 \; \exists j < i \;\; \textit{such that}\; c_i \subseteq s_j \tag{19}$$

then

$$p(x) = \prod\nolimits_{i=1}^{m} p(x_{b_i} | x_{c_i}) \tag{20}$$

The constraint defined as equation (19) 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].

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)
```

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).

For the following examples the theorem leads directly to a factorization:

Example 1. For linear functions

$$\mathit{Linear}(x) = \sum_{i=1}^{n} \alpha_i x_i \tag{21}$$

we have $s_i = \{i\}$ and thus all c_i are empty. This leads to the factorization

$$p(x) = \prod_{i=1}^{n} p(x_i). \tag{22}$$

As this is the distribution used by *UMDA*, *FDA* behaves like *UMDA* (and thus like a simple genetic algorithm) for linear functions.

Example 2. Functions with a chain-like interaction can also be factorized:

$$\mathit{Chain}(x) = \sum_{i=2}^{n} f_i(x_{i-1}, x_i) \tag{23}$$

Here the factorization is

$$p(x) = p(x_1) \prod_{i=2}^{n} p(x_i|x_{i-1}) \tag{24}$$

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 size of the samples (the population size). *FDA* has proven experimentally to be very successful on a number of functions where standard genetic algorithms fail to find the global optimum. In [MM99], the scaling behaviour for various test functions has been studied. The estimation of the probabilities and the generation of new points can be done in polynomial time.

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 will 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*

$$W_f(p) = \sum_x f(x)p(x) \tag{25}$$

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*:*

$$W_f(\beta + \Delta\beta) = W_f(\beta) + \sum_{i \geq 1} \frac{(\Delta\beta)^i}{i\,!} M_{i+1}^c(\beta) \tag{26}$$

where M_i^c are the centred moments

$$M_i^c(\beta) := \sum_x \left[f(x) - W_f(\beta)\right]^i p(x) \tag{27}$$

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

$$M_{i+1}^c(\beta) = \left(\frac{Z_f'(\beta)}{Z_f(\beta)}\right)^{(i)} \qquad \textit{for } i \geq 1, \quad M_1^c{=}0 \tag{28}$$

Proof: The k-th derivative of the partition function obeys for $k \geq 0$:

$$Z_f^{(k)}(\beta) = \sum_x f(x)^k e^{\beta f(x)} \tag{29}$$

Thus the moments for $k \geq 1$ can be calculated as

$$M_k(\beta) := \sum_x f(x)^k p(x) = \frac{Z_f^{(k)}(\beta)}{Z_f(\beta)} \tag{30}$$

and thus

$$W_f(\beta) = M_1(\beta) = Z_f'(\beta)/Z_f(\beta). \tag{31}$$

Direct evaluation of the derivatives of W leads to complicated expressions. Therefore we consider the translated fitness function $\tilde{f}(x) := f(x) + r$. $\tilde{p}_\beta(x)$ is the associated Boltzmann distribution, $\tilde{Z}_f$, $\tilde{M}_k^c$, and $\tilde{W}_f$ the partition functions, moments, and average fitness. Let $C := e^{\beta r}$. Then we have with $k \geq 1$:

$$\tilde{Z}_f(\beta) = \sum_x e^{\beta(f(x)+r)} = C \cdot Z_f(\beta) \tag{32}$$

$$\tilde{p}_\beta(x) = \frac{e^{\beta(f(x)+r)}}{\tilde{Z}_f} = \frac{C \cdot e^{\beta f(x)}}{C \cdot Z_f} = p_\beta(x) \tag{33}$$

$$\tilde{Z}_f'(\beta) = \sum_x \tilde{f}(x) e^{\beta(f(x)+r)} \tag{34}$$

$$= C \cdot Z_f'(\beta) + rC \cdot Z_f(\beta) \tag{35}$$

$$\tilde{W}_f(\beta) = \tilde{M}_1(\beta) = \frac{\tilde{Z}_f'(\beta)}{\tilde{Z}_f(\beta)} = M_1(\beta) + r \tag{36}$$

$$\tilde{W}_f^{(k)}(\beta) = \left(\frac{Z_f'(\beta)}{Z_f(\beta)}\right)^{(k)} = W_f^{(k)}(\beta) \tag{37}$$

$$\tilde{M}_k^c(\beta) = \sum_x \left[(f(x)+r) - (M_1(\beta)+r)\right]^k \tilde{p}_\beta(x) \tag{38}$$

$$= M_k^c(\beta) \tag{39}$$

It follows that the derivatives of W and the centered moments don't change if the fitness function is shifted.

Lemma 3. *If for a β^* we have $\tilde{Z}_f'(\beta^*) = 0$, then with $k \geq 1$:*

$$\left(\frac{\tilde{Z}_f'(\beta^*)}{\tilde{Z}_f(\beta^*)}\right)^{(k-1)} = \frac{\tilde{Z}_f^{(k)}(\beta^*)}{\tilde{Z}_f(\beta^*)} \tag{40}$$

Proof: The proof is by induction. For $k{=}1$ it is true. As the induction step we have

$$\begin{aligned}\left(\frac{\tilde{Z}_f'(\beta^*)}{\tilde{Z}_f(\beta^*)}\right)^{(k)} &= \left[\left(\frac{\tilde{Z}_f'(\beta^*)}{\tilde{Z}_f(\beta^*)}\right)^{(k-1)}\right]' \overset{\text{Ind}}{=} \left(\frac{\tilde{Z}_f^{(k)}(\beta^*)}{\tilde{Z}_f(\beta^*)}\right)' \\ &= \frac{\tilde{Z}_f^{(k+1)}(\beta^*)}{\tilde{Z}_f(\beta^*)} - \underbrace{\frac{\tilde{Z}_f^{(k+1)}(\beta^*) \cdot \tilde{Z}_f'(\beta^*)}{\tilde{Z}_f(\beta^*)^2}}_{=0}\end{aligned}$$

■

In order to finish the proof we just have to put the parts together. Let β be arbitrary, but fixed. With $r := -Z_f'(\beta)/Z_f(\beta) = -M_1(\beta)$ it follows $\tilde{Z}_f'(\beta) = 0$. Then we have $\tilde{M}_1(\beta) = M_1(\beta) + r = 0$, hence

$$\tilde{M}_k^c(\beta) = \tilde{M}_k(\beta) \tag{41}$$

Also we have with $k \geq 1$:

$$\begin{aligned}W_f^{(k)}(\beta) &\overset{(37)}{=} \tilde{W}_f^{(k)}(\beta) = \left(\frac{\tilde{Z}_f'(\beta)}{\tilde{Z}_f(\beta)}\right)^{(k)} \overset{(40)}{=} \frac{\tilde{Z}_f^{(k+1)}(\beta)}{\tilde{Z}_f(\beta)} \\ &\overset{(30)}{=} \tilde{M}_{k+1}(\beta) \overset{(41)}{=} \tilde{M}_{k+1}^c(\beta) \overset{(39)}{=} M_{k+1}^c(\beta)\end{aligned}$$

The proof was done for a specific r. As the first and the last term do not depend on r, the equation holds universally. Together with the Taylor expansion of $W_f(\beta)$ the proof is finished. ■

Corollary 2. *We have approximately*

$$W_f(\tilde{\beta}) \approx W_f(\beta) + (\tilde{\beta} - \beta) \cdot \sigma_f^2(\beta) \tag{42}$$

where $\sigma_f^2(\beta)$ is the variance of the distribution, defined as $\sigma_f^2(\beta) := M_2^c(\beta)$.

This approximation can also be found in [KGV83].

Lemma 4. *The variance of the Boltzmann distribution obeys*

$$f(x) \neq \textit{const.} \implies \sigma_f^2(\beta) > 0 \tag{43}$$

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

$$\sigma_f^2(\beta) = \sum_x \left[f(x) - W_f(\beta)\right]^2 p_\beta(x) \overset{!}{=} 0, \tag{44}$$

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

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

$$\tilde{\beta} > \beta \implies W_f(\tilde{\beta}) > W_f(\beta) \tag{45}$$

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 (42) 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

$$\Delta\beta(t) := \beta(t+1) - \beta(t) = \frac{W_f^{\text{new}}(t) - W_f(\beta(t))}{\sigma_f^2(\beta(t))} \tag{46}$$

From (42) 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 approximately given by [Müh98]

$$R_f(t) := W_f\big(\beta(t+1)\big) - W_f\big(\beta(t)\big) \approx I_\tau b\sqrt{\sigma_f^2} \tag{47}$$

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

$$\begin{aligned} R_f(t) &= W_f\big(\beta(t+1)\big) - W_f\big(\beta(t)\big) && (48)\\ &\approx c\cdot\sqrt{\sigma_f^2(\beta(t))} && (49) \end{aligned}$$

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 6. *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, $\sigma_f^2(\beta) = \hat{c}^2\cdot\sigma_{\hat{f}}^2(\hat{\beta})$. Hence we have $\Delta\hat{\beta}(t){=}\Delta\beta(t)/\hat{c}$. ■

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

$$\begin{aligned} R_f(t) &= \sum_{i\geq 1}\frac{c^i}{i!\,\sigma(t)^i}M_{i+1}^c && (50)\\ &= c\cdot\sigma(t) + \frac{c^2 M_3^c}{2\,\sigma(t)^2} + \frac{c^3 M_4^c}{6\,\sigma(t)^3} + \ldots && (51) \end{aligned}$$

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

The examples in this section are easy problems, but in these cases the dynamics can be studied in detail.

6.1 The function *OneMax*

For $f = \mathit{OneMax}$ we can calculate the Boltzmann distribution. We have

$$Z_f(\beta) = \sum_x e^{\beta|x|} = \sum_{i=0}^{n}\binom{n}{i}\left(e^\beta\right)^i = (1+e^\beta)^n \tag{52}$$

All marginal distributions remain the same if started from the uniform distribution.

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

and thus

$$p_\beta(x_1{=}1) = \frac{e^\beta}{1+e^\beta} =: \hat{p}_\beta \tag{53}$$

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

$$W_f(\beta) = \frac{n(1+e^\beta)^{n-1}e^\beta}{(1+e^\beta)^n} = \frac{ne^\beta}{1+e^\beta} \quad \Big(= n\cdot\hat{p}_\beta\Big) \tag{54}$$

The variance is given by (28):

$$\begin{aligned} \sigma_f^2(\beta) &= \tfrac{Z_f''(\beta)}{Z_f(\beta)} - \left(\tfrac{Z_f'(\beta)}{Z_f(\beta)}\right)^2\\ &= \frac{ne^\beta}{(1+e^\beta)^2} = n\cdot\hat{p}_\beta(1-\hat{p}_\beta) && (55) \end{aligned}$$

We get the following difference equation for the inverse temperature:

$$\beta(t+1) = \beta(t) + c\cdot\frac{1+e^{\beta(t)}}{\sqrt{n}\cdot e^{\beta(t)/2}} \tag{56}$$

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

$$R_f(t) = c\frac{e^{\beta/2}\sqrt{n}}{1+e^{\beta}} - c^2\frac{(e^{\beta}-1)}{2(e^{\beta}+1)} + \dots \tag{57}$$

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

$$R_f(t) \approx c\frac{e^{\beta/2}\sqrt{n}}{1+e^{\beta}} \tag{58}$$

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

$$\frac{d\beta}{dt} = \frac{c}{\sigma(t)} = \frac{2c}{\sqrt{n}}\cosh\bigl(\beta(t)/2\bigr) \tag{59}$$

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

$$\beta(t) = 4\,\mathrm{argtanh}\left[\tan\left(\frac{c\cdot t}{2\sqrt{n}}\right)\right] \quad \text{with } t \le \frac{\pi\sqrt{n}}{2c} \tag{60}$$

where $\mathrm{argtanh}(x) = \frac{1}{2}\ln\bigl((1+x)/(1-x)\bigr)$. Together with equation (53) we get

$$\hat{p}_\beta(t) = \frac{1}{2}\left(1 + \sin\left(\frac{c\cdot t}{\sqrt{n}}\right)\right) \tag{61}$$

If we compare this to previous results obtained for truncation selection [Müh98], we see that the dynamic equation (61) 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 $c \in [0.8, 1.3]$.

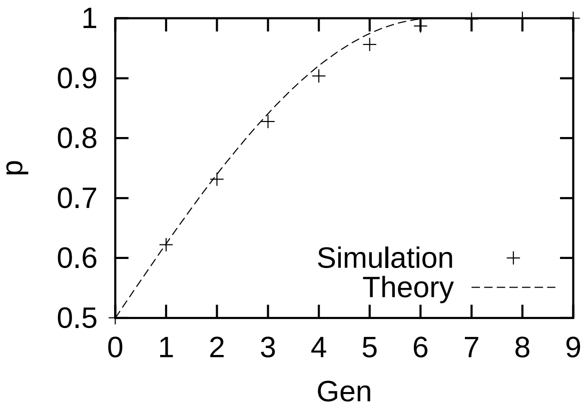

Figure 1: Univariate marginal distribution for *OneMax* with Boltzmann selection, $n = 16$

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.

Gen	β-diff	β-deq	$W_f(\beta)$	ΔW	$R_f(t-1)$
0	0.00	0.00	32.00	0.00	0.00
1	0.25	0.25	35.98	3.98	4.00
2	0.50	0.51	39.87	3.89	3.97
3	0.76	0.77	43.61	3.74	3.88
4	1.03	1.04	47.14	3.53	3.73
5	1.31	1.34	50.42	3.28	3.52
6	1.62	1.66	53.41	2.98	3.27
7	1.95	2.03	56.01	2.65	2.97
8	2.33	2.45	58.34	2.28	2.64
9	2.77	2.97	60.24	1.90	2.27
10	3.30	3.64	61.73	1.50	1.88
11	3.98	4.64	62.83	1.09	1.48
12	4.91	6.68	63.53	0.71	1.07
13	6.38	∞	63.89	0.36	0.68
14	9.43	∞	63.99	0.10	0.33
15	23.4	∞	64.00	0.01	0.07

Table 1: Response to selection and inverse temperature for *OneMax* (64).

This problem is studied in detail in table 1. Here the different approximations for 64 bits and $c{=}1$ were studied. In the second and third column are the values of β according to the difference equation and the differential equation. The approximation is very good in the beginning, but in the end the two columns differ considerably.

The expected increase in average fitness is the first term in the Taylor series, given by equation (58). It is shown in the sixth column, whereas the fifth column shows the *actual* increase by taking the difference of adjacent values in column 4.

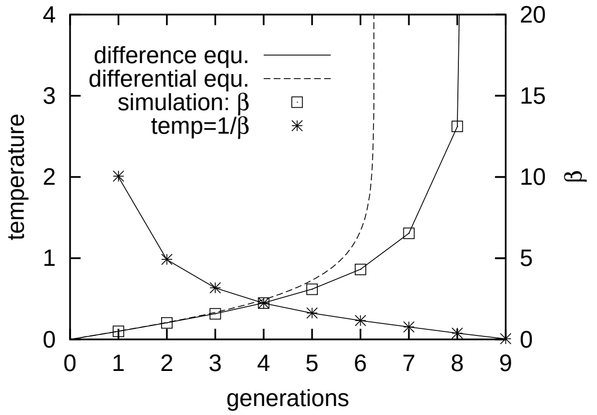

Figure 2: $\beta(t)$ for *OneMax*(16), simulation versus theory

Figure 2 shows the resulting annealing schedule for 16 bits. Plotted are the inverse temperature β as well as the temperature $1/\beta$. The theoretical solution according to (56) are shown with a simulation run. To compare we have also shown the differential equation (60).

One can see that β grows almost linearly in the beginning, while in the end the temperature goes to 0 linearly. According to lemma 2, for $\beta \geq 11.2$, half of the population should consist of the global optimum. In the simulation run, the algorithm terminates after reaching this value.

The results from the simulation remain very close to the theoretically predicted values.

6.2 Linear functions

For linear functions

$$\mathit{Linear}(x) = \sum_{i=1}^{n} \alpha_i x_i \tag{62}$$

the factorization of the Boltzmann distribution was calculated in equation (22). We can also calculate the partition function and get

$$Z_f(\beta) = \prod_{i=1}^{n} (1 + e^{\beta\alpha_i}) \tag{63}$$

and

$$p_i(\beta) := p_\beta(x_i = 1) = \frac{e^{\beta\alpha_i}}{1 + e^{\beta\alpha_i}}. \tag{64}$$

Furthermore, because of the nature of the distribution, the variance is just the sum of the variance of the factors and we have

$$\sigma_f^2(\beta) = \sum_{i=1}^{n} \frac{\alpha_i^2 e^{\beta\alpha_i}}{(1 + e^{\beta\alpha_i})^2} = \sum_{i=1}^{n} \alpha_i^2 p_i(\beta)\big(1 - p_i(\beta)\big) \tag{65}$$

and thus

$$\beta(t+1) = \beta(t) + \frac{c}{\sqrt{\sum_i \alpha_i^2 p_i(\beta)\big(1 - p_i(\beta)\big)}} \tag{66}$$

By differentiating (64) we get

$$\begin{aligned}\frac{dp_i(\beta)}{dt} &= \frac{\alpha_i e^{\beta\alpha_i}(1 + e^{\beta\alpha_i})\beta' - e^{\beta\alpha_i}\alpha_i e^{\beta\alpha_i}\beta'}{(1 + e^{\beta\alpha_i})^2} \\ &= p_i(\beta)\big(1 - p_i(\beta)\big)\alpha_i \tfrac{d\beta}{dt} \end{aligned} \tag{67}$$

Therefore we obtain the approximate differential equation

$$\frac{dp_i(\beta)}{dt} = c \cdot \frac{p_i(\beta)\big(1 - p_i(\beta)\big)\alpha_i}{\sqrt{\sum_i \alpha_i^2 p_i(\beta)\big(1 - p_i(\beta)\big)}} \tag{68}$$

Note that the solution of these differential equations remain the same if we multiply all α_i by a constant factor, as predicted.

For *OneMax* we have $\alpha_i = 1$. In this case all marginal frequencies are equal to p_β. We obtain the differential equation

$$\frac{dp_\beta}{dt} = c\sqrt{p_\beta(1 - p_\beta)/n} \tag{69}$$

The solution of this equation is given by (61).

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 depends critically on a good annealing schedule. Therefore 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.

Bibliography

[CF99] K. Chellapilla and D.B. Fogel. Fitness distributions in evolutionary computation: motivation and examples in the continuous domain. *BioSystems*, 54:15–29, 1999.

[dlMT93] M. de la Maza and B. Tidor. An analysis of selection procedures with particular attention paid to proportional and boltzmann selection. In S. Forrest, editor, *Proc. of the Fifth Int. Conf. on Genetic Algorithms*, pages 124–131, San Mateo, CA, 1993. Morgan Kaufman.

[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.

[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.

[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.