Invited paper

Convergence Theory and Applications of the Factorized Distribution Algorithm

Heinz Mühlenbein and Thilo Mahnig

RWCP[1] Theoretical Foundation GMD Laboratory, GMD - Forschungszentrum Informationstechnik, St. Augustin

The paper investigates the optimization of additively decomposable functions (ADF) by a new evolutionary algorithm called Factorized Distribution Algorithm (FDA). FDA is based on a factorization of the distribution to generate search points. First separable ADFs are considered. These are mapped to generalized linear functions with metavariables defined for multiple alleles. The mapping transforms FDA into an Univariate Marginal Frequency Algorithm (UMDA). For UMDA the exact equation for the response to selection is computed under the assumption of proportionate selection. For truncation selection an approximate equation for the time to convergence is used, derived from an analysis of the *OneMax* function. FDA is also numerically investigated for non separable functions. The time to convergence is very similar to separable ADFs. FDA outperforms the genetic algorithm with recombination of strings by far.

Keywords: response to selection, Fisher's Theorem, additively decomposed functions, genetic algorithm, factorized distribution

1. Introduction

A genetic algorithm is a population-based search method. A set of points is generated, promising points are selected and new points are generated using the genetic operators recombination/crossover and mutation. The simple genetic algorithm (Goldberg, 1989) selects promising points according to

$$p^s(\mathbf{y}, t) = p(\mathbf{y}, t)\frac{g(\mathbf{y})}{\bar{g}(t)}. \qquad (1)$$

Here $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ denotes a vector of discrete random variables (genotype), $p(\mathbf{y}, t)$ is the distribution of $\mathbf{y}$ at generation t and $\bar{g}(t) = \sum p(\mathbf{y}, t)g(\mathbf{y})$ is the average fitness of the population. For simplicity we assume binary variables $y_i \in \{0, 1\}$. The above selection scheme is called *proportionate*, because above average genotypes increase in proportion to their fitness values. Selection should guide the creation of new points, therefore one would like to generate new points according to

$$p(\mathbf{y}, t+1) = p^s(\mathbf{y}, t). \qquad (2)$$

A discrete density is defined by 2^n parameters. This means that a straightforward implementation of this equation is computationally prohibitive. Therefore the central question of a genetic algorithm, as well as for any population-based algorithm, can be formulated as follows: If and how can Equation 2 be approximated with substantially less than exponential computational effort? Instead of extending the genetic algorithm, we take a new approach based on probability theory. We try to approximate the equation by explicit aggregation.

A possible structure for aggregation is a schema (Holland, 1992). We just give an example for a schema. Extending the usual notation

$$H(y_i, y_k) = (*, \ldots, *, y_i, *, \ldots, *, y_k, *, \ldots *) \qquad (3)$$

defines a schema where the values of y_i and y_k are held fixed, the other variables are free. If we sum Equation 1 over all $\mathbf{y}$ which are members of schema H, we obtain the probability of H after

[1]Real World Computing Partnership

selection

$$p^s(H(y_i, y_k), t) = \sum_{\mathbf{y} \in H} \frac{p(\mathbf{y}, t) g(\mathbf{y})}{\bar{g}(t)}. \quad (4)$$

In probability terms $p^s(H(y_i, y_k), t)$ is a marginal distribution which we abbreviate by $p^s(y_i, y_k, t)$. In fact, all schemata define corresponding marginal distributions. We can now state our question in terms of probability theory: *Does a set of marginal distributions exist which gives a good approximation to Equation 2 and which can be computed in polynomial time?*

The simplest choice are first order schemata or univariate marginal distributions. They are used by the *Univariate Marginal Distribution Algorithm* (UMDA) (Mühlenbein, 1998). Here new points are generated according to

$$p(\mathbf{y}, t+1) = \prod_{i=1}^{n} p^s(y_i, t). \quad (5)$$

$p^s(y_i, t)$ is the density of a first order schema defined by the gene at loci i. If the distribution $p^s(\mathbf{y}, t)$ in Equation 1 is complex, first order schemata give a bad approximation. It is often claimed by using the Schema Theorem (Goldberg, 1989) that creating new points by Mendelian recombination is a good approximation to Equation 2. But Mühlenbein (1998) has given numerical and theoretical evidence that Mendelian recombination behaves similarly to UMDA, i.e., it approximates more Equation 5 than Equation 2.

In order to get a better approximation for complex distributions higher order schemata have to be used. But which schemata should be used? We illustrate the problem with an example. If three loci ($n = 3$) are given and second order schemata are used, then in general

$$p(\mathbf{y}, t) \neq p(y_1, y_2, t) p(y_1, y_3, t) p(y_2, y_3, t).$$

In fact, there exists no closed expression which gives $p(\mathbf{y})$ as a function of bivariate distributions. This problem is discussed in detail by Mühlenbein et al. (1998).

In this paper we consider fitness functions which allow a factorization of $p^s(\mathbf{y}, t)$ into a product of marginal distributions. An example are *additively decomposable functions* (ADF)

$$g(\mathbf{y}) = \sum_{j=1}^{l} g_j(\mathbf{y}_{S_j}), \quad (6)$$

where $S_j \subset \{1, \ldots, n\}$ and $\bigcup S_j = \{1, \ldots, n\}$.

The outline of the paper is as follows. First we investigate separable functions, i.e. $S_i \cap S_j = \emptyset$.In Section 2 we transform the given function into a function with metavariables defined for multiple alleles. For this problem an exact equation of the response to selection is computed. From this equation Fisher's (1958) fundamental theorem of natural selection follows for generalized linear functions. In Section 4 a weak form of Fisher's theorem is proven for arbitrary fitness functions. Then we examine the relation between the structure of the response equation and the structure of the fitness function. The results are used to compute an approximate solution for linear fitness functions. In Section 7 we investigate a special function where the defining sets overlap in one variable. Here conditional distributions have to be used.

The last two sections deal with truncation selection. Here only approximate solutions are obtained. We define and evaluate the *Factorized Distribution Algorithm* (FDA) with numerical examples. FDA is an extension of UMDA using a given factorization.

2. Mapping to Metavariables with Multiple Alleles

We assume that the probability $p^s(\mathbf{y}, t)$ can be expressed as a product

$$p^s(\mathbf{y}, t) = \prod_{i \in I} p(\mathbf{y}_{S_i}, t) \quad (7)$$

where $I = \{1, \ldots, l\}$ and the index sets S_i are *disjoint* and

$$\bigcup_{i \in I} S_i = \{1, \ldots, n\}.$$

We combine all y_j with $j \in S_i$ into one metavariable x_i with more than two alleles. Let

$$\begin{aligned} \Lambda_i &:= \{1, \ldots, 2^{|S_i|} - 1\} \\ \bar{\Lambda}_i &:= \Lambda_i \cup \{0\}. \end{aligned} \quad (8)$$

Then the given fitness function $g(\mathbf{y})$ can be formulated as a fitness function $f(\mathbf{x})$ with metavariables x_i with values from $\bar{\Lambda}_i$ by a one to one mapping $\Psi_i : \{0, 1\}^{|S_i|} \rightarrow \bar{\Lambda}_i$. We demonstrate the mapping with some examples.

Example 1: Let $n = 2$. Then

$$g(\mathbf{y}) = g_1(y_1, y_2) = \alpha_0 + \alpha_1 y_1 + \alpha_2 y_2 + \alpha_{12} y_1 y_2. \tag{9}$$

We define a metavariable $x_1 = \Psi_1(y_1, y_2) = y_1 + 2 \cdot y_2$ with $x_1 \in \{0, 1, 2, 3\}$. The function is transformed to

$$f_1(x_1) = c_0 + \sum_{j=1}^{3} c_1^j \nu_1^j$$

with

$$\nu_1^j = \begin{cases} 1 & x_1 = j \\ 0 & \text{otherwise} \end{cases}$$

$$c_0 = \alpha_0, \qquad c_1^1 = \alpha_0 + \alpha_1,$$
$$c_1^2 = \alpha_0 + \alpha_2, \quad c_1^3 = \alpha_0 + \alpha_1 + \alpha_2 + \alpha_{12}.$$

Example 2: Let $n = 2m$ and

$$g(\mathbf{y}) = \sum_{i=1}^{m} g_i(y_{2i-1}, y_{2i}).$$

With metavariables $x_i = \Psi_i(y_{2i-1}, y_{2i}) = y_{2i-1} + 2y_{2i}$ we obtain a *generalized linear function*

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i)$$

with $f_i(x_i) = g_i(y_{2i-1}, y_{2i})$.

Example 3: Now higher order interactions are considered. Let $n = 2m$ and

$$g(\mathbf{y}) = \sum_{i=1}^{m} g_i(y_{2i-1}, y_{2i}) + \sum_{i<j} g_{ij}(y_{2i-1}, y_{2i}, y_{2j-1}, y_{2j}).$$

We first choose $x_i = \Psi_i(y_{2i-1}, y_{2i}) = y_{2i-1} + 2y_{2i}$ and get

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i) + \sum_{i<j} f_{ij}(x_i, x_j) \qquad \text{with} \quad x_i \in \{0, \ldots, 3\}$$

where $f_i(x_i) = g_i(y_{2i-1}, y_{2i})$ and $f_{ij}(x_i, x_j) = g_{ij}(y_{2i-1}, y_{2i}, y_{2j-1}, y_{2j})$.

We can also use metavariables with a larger alphabet, e.g. combine four bits into a metavariable. We assume that m is even and the index sets (i, j) with $g_{ij} \neq 0$ are *disjoint*. For simplicity we assume a chain, i.e. $g_{i,i+1} \neq 0$. We set $x_i = \Psi_i(y_{4i-3}, y_{4i-2}, y_{4i-1}, y_{4i}) = y_{4i-3} + 2y_{4i-2} + 4y_{4i-1} + 8y_{4i}$ and get

$$f(\mathbf{x}) = \sum_{i=1}^{n/4} f_i(x_i) \quad \text{with} \quad x_i \in \{0, \ldots, 15\}$$

with suitable f_i. This mapping transforms g into a generalized linear function.

Example 4: Now consider the inverse mapping. Let a function $f(\mathbf{x})$ with $x_i \in \{0, 1, 2, 3\}$, $i \in \{1, \ldots, m\}$ be given. Then set $n = 2m$ and

$$y_{2i-1} = (x_i \,\text{MOD}\, 2) \qquad y_{2i} = (x_i \,\text{DIV}\, 2).$$

In this way, we have

$$\begin{aligned} \Psi_i(y_{2i-1}, y_{2i}) &= y_{2i-1} + 2y_{2i} \\ \Psi_i^{-1}(x_i) &= (x_i \,\text{MOD}\, 2, x_i \,\text{DIV}\, 2) \\ f(\mathbf{x}) &= f(x_1, \ldots, x_m) \\ &= g(y_1, y_2, \ldots, y_{2m-1}, y_{2m}) \\ &= g(\Psi_1^{-1}(x_1), \ldots, \Psi_m^{-1}(x_m)) \end{aligned}$$

by defining

$$\begin{aligned} g(\mathbf{y}) &= g(y_1, y_2, \ldots, y_{2m-1}, y_{2m}) \\ &= f((y_1 + 2y_2), \ldots, (y_{2m-1} + 2y_{2m})) \\ &= f(\Psi_1(y_1, y_2), \ldots, \Psi_m(y_{2m-1}, y_{2m})). \end{aligned}$$

If the metavariables are defined by Equation 8, then the given fitness function is transformed into a generalized linear function. This may lead to a large set $\bar{\Lambda}_i$. For computational reasons it might be necessary to partition the set S_i and use a separate metavariable for each partition. This was done in example 3. In this case the transformed function might be non linear. Therefore in the next section we compute the equation for the response to selection (Mühlenbein, 1988) for the general case.

3. The Exact Equation for the Response

In the following we will call x_i a variable, keeping in mind that its domain of definition is discrete. Then $\mathbf{x} = (x_1, \ldots, x_{|I|})$ is a vector of discrete random variables. Furthermore

$$p(\mathbf{x}, t) = \prod_{i \in I} p(x_i, t), \tag{10}$$

$$\bar{f}(t) = \sum_{\mathbf{x}} \prod_{i \in I} p(x_i, t) f(\mathbf{x}). \tag{11}$$

Let v_i be a shorthand notation for $x_i = v_i$. v_i will be an element of Λ_i or $\bar{\Lambda}_i$. For the next proof we will consider $p(v_i, t) =: p[v_i]$ to be variables of $\bar{f}(t)$. Thus $p[v_i]$ denotes a *variable* of the function $\bar{f}(t)$, whereas $p(x_i = v_i, t)$ denotes the *value* of the corresponding marginal probability distribution. Then $\bar{f}(t)$ is a polynomial in $p[v_i]$. In order to distinguish the interpretations we also write

$$W(\mathbf{p}) := \bar{f}(t).$$

Note that $p(x_i = 0)$ is not a free parameter, because we have the additional constraint

$$p(x_i = 0, t) = 1 - \sum_{v_i \in \Lambda_i} p(v_i, t). \tag{12}$$

We will later derivate $W = \bar{f}(t)$ with respect to $p[v_i]$. For this operation additional definitions are necessary.

Definition 1. *Let J be a subset of I, $J \subseteq I$. Let $v_J \in \Lambda_J$, where Λ_J denotes the product space $\prod_{j \in J} \Lambda_j$.*

$$\varepsilon_J := (-1)^{|J|} \tag{13}$$

$$p(x_J, t) := \prod_{j \in J} p(x_j, t) \tag{14}$$

$$\sum_{\mathbf{x}|x_J = v_J} F(\mathbf{x}) \quad \text{sum over all } x \text{ with } x_j, j \in J \text{ fixed} \tag{15}$$

$$T_J\mathbf{x} := (\hat{x}_1, \ldots, \hat{x}_{|I|}) \quad \text{with}$$

$$\hat{x}_i := \begin{cases} x_i & i \notin J \\ 0 & i \in J \end{cases} \tag{16}$$

$$\partial_{v_J} W := \left(\prod_{j \in J} \frac{\partial}{\partial p[v_j]} \right) W \tag{17}$$

where the $\frac{\partial}{\partial p[v_j]}$ are multiplied formally.

Following the proof in (Mühlenbein, 1998) for binary genes we will compute the exact equation for the response to selection. We will need the expressions

$$\bar{f}(v_i, t) := \sum_{\mathbf{x}|x_i = v_i} f(\mathbf{x}) \prod_{j \neq i} p(x_j, t) \tag{18}$$

$$F(v_i, t) := \bar{f}(v_i, t) - \bar{f}(t). \tag{19}$$

Then

$$V_1 = \sum_{i \in I} \sum_{v_i \in \bar{\Lambda}_i} p(v_i, t) F(v_i, t)^2 \tag{20}$$

defines the *additive genetic variance* (Mühlenbein, 1998). By appropriate summation of Equation 1 and using Equation 2 we obtain difference equations for the univariate marginal frequencies

$$p(v_i, t+1) = p(v_i, t) + p(v_i, t) \cdot \frac{F(v_i, t)}{\bar{f}(t)}, \quad v_i \in \bar{\Lambda}_i. \tag{21}$$

Now we are able to define UMDA.

Definition 2. *The Univariate Marginal Distribution* **UMDA** *algorithm generates new points according to*

$$p(\mathbf{x}, t+1) = \prod_{i \in I} p(x_i, t+1). \tag{22}$$

In the original space UMDA generates new points according to

$$p(\mathbf{y}, t+1) = \prod_{i \in I} p(\mathbf{y}_{S_i}, t+1). \tag{23}$$

$p(y_{S_i}, t+1)$ can be derived from $p(x_i, t+1)$ by using the mapping Ψ_i^{-1}. Therefore UMDA defines a *Factorized Distribution Algorithm* in the original space. If the size of the index sets $|S_i|$ is bounded by a fixed constant k, then only $l * 2^k = O(n)$ marginal distributions $p(y_{S_i}, t)$ have to be computed. This is an enormous reduction compared to the $2^n - 1$ needed without a factorization.

Theorem 1. *For* **UMDA** *with multiple alleles and proportionate selection, the response to selection* $R(t) = \bar{f}(t+1) - \bar{f}(t)$ *is given by*

$$R(t) = \frac{V_1}{W} + \frac{1}{2} \sum_{i,j \in I} \sum_{\substack{v_i \in \Lambda_i \\ w_j \in \Lambda_j}} \frac{p(v_i,t)F(v_i,t)p(w_j,t)F(w_j,t)}{W^2} \cdot \frac{\partial^2 W}{\partial p[v_i] \partial p[w_j]} + \frac{1}{3! \cdot W^3} \sum_{|J|=3} \sum_{v_J \in \Lambda_J} \sum_{j \in J} \Big(\prod p(v_j,t)F(v_j,t) \Big) \partial_{v_J} W + \ldots \tag{24}$$

Proof: We start with a multidimensional Taylor expansion of $R(t) = \bar{f}(t+1) - \bar{f}(t)$, where t is considered fixed and $p[v_i]$ are variables. With the notation defined before, we have

$$R(t) = \sum_{k \geq 1} \frac{1}{k!} \left(\sum_{i \in I} \sum_{v_i \in \Lambda_i} \Delta p(v_i,t) \frac{\partial}{\partial p[v_i]} \right)^k W$$

with

$$\Delta p(v_i,t) = p(v_i,t+1) - p(v_i,t).$$

From (21) it follows that

$$\Delta p(v_i,t) = p(v_i,t) \cdot \frac{F(v_i,t)}{W}.$$

Because of the structure of W (Equation 11) we have $\frac{\partial}{\partial p[v_i]} \frac{\partial}{\partial p[w_i]} W = 0$. Therefore we have already proven that part of the expression where derivatives of order 2 or higher are involved. We now calculate the first derivatives of W. Choose $i \in I$. We will use the shorthand notation $0_i := (x_i = 0)$. The parameter t will be omitted in the following.

$$\begin{aligned} W &= \sum_{v_i \in \bar{\Lambda}_i} \sum_{\mathbf{x}|x_i=v_i} f(\mathbf{x}) p(v_i) \prod_{j \neq i} p(x_j) \\ &= \sum_{v_i \in \Lambda_i} p(v_i) \sum_{\mathbf{x}|x_i=v_i} f(\mathbf{x}) \prod_{j \neq i} p(x_j) + \\ &\quad \sum_{\mathbf{x}|x_i=0} f(\mathbf{x}) \Big[1 - \sum_{v_i \in \Lambda_i} p(v_i) \Big] \prod_{j \neq i} p(x_j) \\ &= \sum_{v_i \in \Lambda_i} p(v_i) \Big(\sum_{\mathbf{x}|x_i=v_i} f(\mathbf{x}) \prod_{j \neq i} p(x_j) - \\ &\quad \sum_{\mathbf{x}|x_i=0} f(\mathbf{x}) \prod_{j \neq i} p(x_j) \Big) + \sum_{\mathbf{x}|x_i=0} f(\mathbf{x}) \prod_{j \neq i} p(x_j) \\ &= \sum_{v_i \in \Lambda_i} p(v_i) \left(\bar{f}(v_i) - \bar{f}(0_i) \right) + \bar{f}(0_i). \end{aligned} \tag{25}$$

We now interpret $p(v_i)$ as the variable $p[v_i]$. As the term in the parentheses is independent of $p[v_i]$, we get

$$\frac{\partial W}{\partial p[v_i]} = \bar{f}(v_i) - \bar{f}(0_i) = F(v_i) - F(0_i). \tag{26}$$

The first term of R is thus

$$\begin{aligned} &\sum_{i \in I} \sum_{v_i \in \Lambda_i} \Delta p(v_i) \frac{\partial W}{\partial p[v_i]} \\ &= \frac{1}{W} \sum_{i \in I} \sum_{v_i \in \Lambda_i} p(v_i) F(v_i) \left[F(v_i) - F(0_i) \right]. \end{aligned} \tag{27}$$

From (25) it follows that

$$\begin{aligned} W &= \sum_{v_i \in \Lambda_i} p(v_i) \Big(F(v_i) - F(0_i) \Big) + F(0_i) + W \\ &\Rightarrow \sum_{v_i \in \Lambda_i} p(v_i) F(v_i) = - \Big(1 - \sum_{v_i \in \Lambda_i} p(v_i) \Big) F(0_i) \\ &= -p(0_i) F(0_i). \end{aligned}$$

Therefore we obtain

$$\begin{aligned} &\sum_{i \in I} \sum_{v_i \in \Lambda_i} \Delta p(v_i) \frac{\partial W}{\partial p(v_i)} \\ &= \sum_{i \in I} \sum_{v_i \in \Lambda_i} p(v_i) F^2(v_i) - \sum_{i \in I} F(0_i) \sum_{v_i \in \Lambda_i} p(v_i) F(v_i) \\ &= \sum_{i \in I} \sum_{v_i \in \bar{\Lambda}_i} p(v_i) F^2(v_i). \end{aligned}$$

□

Theorem 1 gives the response to selection for arbitrary fitness functions. Separable fitness functions can be transformed into generalized linear fitness functions. For these function Fisher's theorem (1958) is valid.

Corollary 1: *For a generalized linear function*

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(x_i) \quad x_i \in \bar{\Lambda}_i$$

the response for UMDA with proportionate selection is given by

$$R(t) = \frac{V_1}{W}. \tag{28}$$

Proof: We have

$$\frac{\partial W}{\partial p[v_i]} = \bar{f}(v_i) - \bar{f}(0_i).$$

Now

$$\begin{aligned}\bar{f}(v_i) &= \sum_{k=1}^{m} \sum_{\mathbf{X}|x_i=v_i} \left[f_k(x_k) \prod_{j\neq i} p(x_j, t) \right] \\ &= f_i(v_i) + \sum_{k\neq i} \sum_{\mathbf{X}|x_i=v_i} f_k(x_k) \prod_{j\neq i} p(x_j, t).\end{aligned}$$

For $v_i = 0$ this gives

$$\bar{f}(0_i) = f_i(v_i) + \sum_{k\neq i} \sum_{\mathbf{X}|x_i=v_i} f_k(x_k) \prod_{j\neq i} p(x_j, t).$$

Combining the equations we obtain

$$\frac{\partial W}{\partial p[v_i]} = f_i(v_i) - f_i(0).$$

Thus the first derivatives of W are independent of p. Therefore all higher derivatives will be zero. This gives $R(t) = \frac{V_1}{W}$. □

For the sake of completeness, we will show that for non linear fitness functions higher order derivatives of W do not vanish in general.

Corollary 2: *For fitness functions with second order interactions like*

$$f(\mathbf{x}) = \sum_{k\in G} g_k(x_{k-1}, x_k)$$

with $G = \{2, 4, 6, \ldots, m\}$, *second order derivatives of* W *will not vanish.*

Proof: We have for i even

$$\begin{aligned}\bar{f}(v_i) &= \sum_{x|x_i=v_i} \left(\sum_{k\in G} g_k(x_{k-1}, x_k) \right) \prod_{j\neq i} p(x_j, t) \\ &= \sum_{v_{i-1}\in\bar{\Lambda}_{i-1}} g_i(v_{i-1}, v_i) \sum_{\substack{x|x_i=v_i \\ x_{i-1}=v_{i-1}}} \prod_{j\neq i} p(x_j, t) \\ &\quad + \sum_{\substack{k\in G \\ k\neq i}} \sum_{x|x_i=v_i} g_k(x_{k-1}, x_k) \prod_{j\neq i} p(x_j, t).\end{aligned}$$

This yields

$$\begin{aligned}\frac{\partial W}{\partial p[v_i]} &= \bar{f}(v_i) - \bar{f}(0_i) \\ &= \sum_{v_{i-1}\in\bar{\Lambda}_{i-1}} [g_i(v_{i-1}, v_i) - g_i(v_{i-1}, 0_i)] p(v_{i-1}, t)\end{aligned}$$

and we have

$$\begin{aligned}&\frac{\partial^2 W}{\partial p[v_i]\partial p[v_{i-1}]} \\ &= g_i(v_{i-1}, v_i) - g_i(v_{i-1}, 0) - g_i(0, v_i) + g_i(0, 0).\end{aligned}$$

Therefore the second order derivatives will **not** vanish in general. □

4. A Weak Version of Fisher's Theorem

Fisher's theorem is only valid for generalized linear functions. For arbitrary functions we can only show that the *response* $R(t)$ *is always greater or equal to zero* if new points are generated according to Equation 21. The proof is based on the general form of the inequality by Baum and Eagon (1967), written here in our notation.

Theorem 2. (Baum, Eagon) *Let* $\bar{W}(p) = \bar{W}(\{p_{ij}\})$ *be a polynomial with nonnegative coefficients homogeneous of degree* n *in its variables* p_{ij}, $i \in I$, $j \in \bar{\Lambda}_i$. *Let* $\mathbf{p} = \{p_{ij}\}$ *be any point in the domain* $D : p_{ij} \geq 0$, $\sum_j p_{ij} = 1$. *Let* $\mathbf{p}'$ *denote the point given by the coordinates*

$$p'_{ij} = \frac{p_{ij} \frac{\partial \bar{W}}{\partial p_{ij}}|_{\mathbf{p}}}{\sum_{k\in\bar{\Lambda}_i} p_{ik} \frac{\partial \bar{W}}{\partial p_{ik}}|_{\mathbf{p}}}. \tag{29}$$

Then $\bar{W}(\mathbf{p}') > \bar{W}(\mathbf{p})$ *unless* $\mathbf{p}' = \mathbf{p}$.

Theorem 3. *For* **UMDA** *and proportionate selection we have* $R(t) > 0$ *unless all marginal frequencies remain the same.*

Proof: We consider $W = \bar{f}(t)$ to be a function of the variables $p[v_i] = p(v_i)$ with $v_i \in \bar{\Lambda}_i$. $p(0_i)$ is considered a variable. We define $p_{ij} = p(v_i)$ with $v_i = j$. Then the constraint $\sum_j p_{ij} = 1$ is obviously fulfilled. Next we have to show that Equation (29) is identical to Equation (21) for the univariate marginal frequencies.

We easily obtain

$$\frac{\partial \bar{W}}{\partial p_{ij}}|_{\mathbf{p}} = \bar{f}(x_i = j, t)$$

where $\bar{f}(x_i = j, t)$ is defined in Equation 18. Furthermore

$$\sum_{k \in \bar{\Lambda}_i} p_{ik} \bar{f}(x_i = k, t) = \bar{W}$$

is valid. Therefore the marginal frequencies of UMDA fulfill the assumptions of Theorem 2.

□

Thus UMDA has a numerically very useful property that the average fitness of the population always increases. This property can be used to terminate the algorithm. In the next section we investigate the relation, which exists between the vanishing of all higher order derivatives of W and the structure of f.

5. Relation between f and W

In the following, the parameter t will be considered fixed and is omitted in the formulas.

Theorem 4. *If for all* $J \subseteq I$, $|J| \geq k \geq 2$ *and for all* $v_J \in \Lambda_J : \partial_{v_J} W = 0$, *then for all subsets* K *with* $|K| \leq k$ *there exist functions* $f_K : \bar{\Lambda}_K \to R$ *with* $f_K(0) = 0$ *for* $K \neq 0$ *and* $f_K(\mathbf{x}) = 0$ *for* $x \notin \Lambda_K$ *and*

$$f(\mathbf{x}) = \sum_{|K| \leq k} f_K(x_K). \tag{30}$$

The proof is by induction over $|J|$. It is lengthy and will be omitted. The theorem states that the number of interacting variables is at most k if all order k and higher derivatives of W vanish. We just give two examples.

Corollary: *Assume that all derivatives of* W *except the first vanish, then*

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i), \tag{31}$$

that is, f *is a generalized linear function.*

For binary variables this was already conjectured in (Mühlenbein, 1998).

Let $\bar{\Lambda}_i = \{0, 1\}$. Assume that all derivatives except the first and second vanish, then f is of the form

$$f(\mathbf{x}) = c_0 + \sum_{i=1}^{n} c_i x_i + \sum_{i<j} c_{ij} x_i x_j. \tag{32}$$

In the next section we will approximately solve the difference equations (21) for linear functions.

6. Linear Fitness Functions

Now we will use the theory to obtain the response equations and the change in allele frequencies for *linear fitness functions* with an arbitrary alphabet. Let an arbitrary linear function be given

$$f(\mathbf{x}) = \alpha_0 + \sum_{i \in I} \alpha_i x_i \qquad \text{with } x_i \in \bar{\Lambda}_i. \tag{33}$$

Then

$$\bar{f}(t) = \alpha_0 + \sum_{i \in I} \sum_{v_i \in \Lambda_i} \alpha_i v_i p(v_i, t). \tag{34}$$

The marginal frequencies after one step of proportionate selection are given by Equation 21:

$$\begin{aligned} p(v_i, t+1) &= p(v_i, t) \cdot \frac{\bar{f}(v_i, t)}{\bar{f}(t)} \\ &= p(v_i, t) + p(v_i, t) \cdot \frac{F(v_i, t)}{\bar{f}(t)} \end{aligned} \tag{35}$$

where $\bar{f}(v_i)$ is now given by

$$\bar{f}(v_i, t) = \alpha_0 + \sum_{\mathbf{x}|x_i = v_i} \Big(\prod_{k \neq i} p(x_k, t) \Big) \sum_{j \in I} \alpha_j x_j \tag{36}$$

Theorem 5. *For linear functions marginal distributions and the response to selection are given for UMDA with proportionate selection by*

$$p(v_i, t+1) = p(v_i, t) + p(v_i, t)\frac{\alpha_i v_i - \sum_{w_i \in \Lambda_i} \alpha_i w_i p(w_i, t)}{\bar{f}(t)} \quad (37)$$

$$R(t) = \frac{V_1(t)}{\bar{f}(t)} = \frac{1}{\bar{f}(t)} \sum_{i \in I} \sum_{v_i \in \Lambda_i} \alpha_i^2 v_i^2 \cdot p(v_i, t)(1 - p(v_i, t))^2. \quad (38)$$

Proof: We sort Equation 36 according to multinomials $p(v_J, t)$ for t fixed and $J \subseteq I \setminus i$ and $v_J \in \Lambda_J$, that is, we substitute $p(0_i, t) = 1 - \sum_{v_i \in \Lambda_i} p(v_i, t)$. In order to obtain a given multinomial $p(v_J, t)$, we see that all x_k with $k \notin J$ have to be zero, whereas the x_j with $j \in J$ might be zero or v_j. For $x_j = 0$ a change in the sign of the product is involved. This leads to

$$\bar{f}(v_i, t) = \alpha_0 + \sum_{J \subseteq I \setminus i} \sum_{v_J \in \Lambda_J} p(v_J, t) \cdot \underbrace{\sum_{J' \subseteq J} \varepsilon_{J \setminus J'} \left(\alpha_i v_i + \sum_{k \in J'} \alpha_k v_k \right)}_{H(v_J)}. \quad (39)$$

It is evident that $\sum_{J' \subseteq J} \varepsilon_{J'} \alpha_i v_i = 0 \quad \forall \emptyset \subset J' \subseteq I$.

For $|J| > 1$, each term $\alpha_k v_k$ in $H(v_J)$ occurs exactly $2^{|J|-1}$ times with alternating signs, such that

$$|J| > 1 \implies H(v_J) = 0.$$

For $|J| = 1$ we obtain $H(v_j) = +\alpha_j v_j$, and $H(\emptyset) = \alpha_i v_i$. This yields

$$\bar{f}(v_i, t) = \alpha_0 + \alpha_i v_i + \sum_{j \neq i} \sum_{v_j \in \Lambda_j} \alpha_j v_j p(v_j, t).$$

As derivatives of W of order 2 or higher vanish, we finally get the following equations:

$$p(v_i, t+1) = p(v_i, t) \cdot \frac{\alpha_0 + \alpha_i v_i + \sum_{j \neq i} \sum_{w_j \in \Lambda_j} \alpha_j w_j p(w_j, t)}{\bar{f}(t)} \quad (40)$$

$$= p(v_i, t) + p(v_i, t)\frac{\alpha_i v_i - \sum_{w_i \in \Lambda_i} \alpha_i w_i p(w_i, t)}{\bar{f}(t)}. \quad (41)$$

□

The difference equations 37 can be easily computed numerically. An analytical solution is nevertheless complicated. In the following we will derive an approximate solution for a special case.

Theorem 6. *For $\bar{\Lambda}_i = \{0, 1\}$, $\alpha_0 = 0$, $\alpha_i > 0$ and $p(x_i, 0) = \frac{1}{2}$ with $p_i(t) := p(x_i = 1, t)$, we can approximate the marginal frequency by*

$$p_i(t) \simeq 1 - q_0 a_i^t \quad (42)$$

$$a_i = \frac{\alpha - \alpha_i \left(1 + \frac{\alpha^2 - \alpha_i^2}{\alpha^2 + \beta}\right)}{\alpha - \alpha_i} \quad (43)$$

$$q_i = \frac{1 - \frac{\alpha_i}{\alpha}}{2a_i} \quad (44)$$

with

$$\alpha := \sum_j \alpha_j \qquad \beta := \sum_j \alpha_j^2.$$

Proof: From Equation 37 we obtain

$$p_i(t+1) = p_i(t)\left(1 + \alpha_i \frac{1 - p_i(t)}{\bar{f}(t)}\right).$$

We calculate the frequencies for two generations:

$$p_i(1) = \tfrac{1}{2}\left[1 + \alpha_i \frac{1}{\alpha}\right] = \tfrac{1}{2}\left(1 + \tfrac{\alpha_i}{\alpha}\right) = 1 - \tfrac{1}{2}\left(1 - \tfrac{\alpha_i}{\alpha}\right)$$

$$p_i(2) = \tfrac{1}{2}\left(1 + \tfrac{\alpha_i}{\alpha}\right)\left[1 + \alpha_i \frac{\frac{1}{2}(1 - \frac{\alpha_i}{\alpha})}{\frac{1}{2}\sum_j \alpha_j (1 + \frac{\alpha_j}{\alpha})}\right] = \tfrac{1}{2}\left(1 + \tfrac{\alpha_i}{\alpha}\right)\left[1 + \alpha_i \frac{\alpha - \alpha_i}{\alpha^2 + \beta}\right]$$

$$= \tfrac{1}{2} + \tfrac{1}{2}\tfrac{\alpha_i}{\alpha} + \tfrac{1}{2}\tfrac{\alpha_i}{\alpha}\frac{\alpha^2 - \alpha_i^2}{\alpha^2 + \beta} = 1 - \tfrac{1}{2}\left[1 - \tfrac{\alpha_i}{\alpha}\left(1 + \frac{\alpha^2 - \alpha_i^2}{\alpha^2 + \beta}\right)\right].$$

Using the *ansatz* $p_i(t) = 1 - q_i a_i^t$ we obtain the equations

$$q_i a_i = \tfrac{1}{2}\left(1 - \tfrac{\alpha_i}{\alpha}\right)$$
$$q_i a_i^2 = \tfrac{1}{2}\left[1 - \tfrac{\alpha_i}{\alpha}\left(1 + \frac{\alpha^2 - \alpha_i^2}{\alpha^2 + \beta}\right)\right].$$

Dividing the second equation by the first one we get a_i and then q_i. □

The *OneMax* function is defined by $\alpha_i = 1$, $i = 1, \ldots, n$. In this case we obtain

$$a_i = 1 - \frac{1}{n}$$
$$p(t) = 1 - \frac{1}{2}(1 - \frac{1}{n})^t.$$

This is the *exact* solution of Equation 37. For *OneMax* our approximation is exact.

Next we consider a more difficult function. Let $n = 10$, $\bar{\Lambda}_i = \{0, 1\}$, $\alpha_0 = 0$, $\alpha_i = 2^{i-1}$. Thus the fitness function is

$$f(x_1, \ldots, x_5) = \sum_i 2^{i-1} x_i \quad \text{or} \quad f(x) = x. \tag{45}$$

From Theorem 6 we compute the numerical values for $a_8, \ldots, a_{10}$ and $q_8, \ldots, q_{10}$:

a_{10}	=	0.437	q_{10}	=	0.571
a_9	=	0.765	q_9	=	0.490
a_8	=	0.894	q_8	=	0.489

We numerically compare the approximate values with the exact values. The result is shown in Figure 1. The frequencies $(1 - p_i)$ are plotted on a logarithmic scale, which leads to almost linear graphs as predicted.

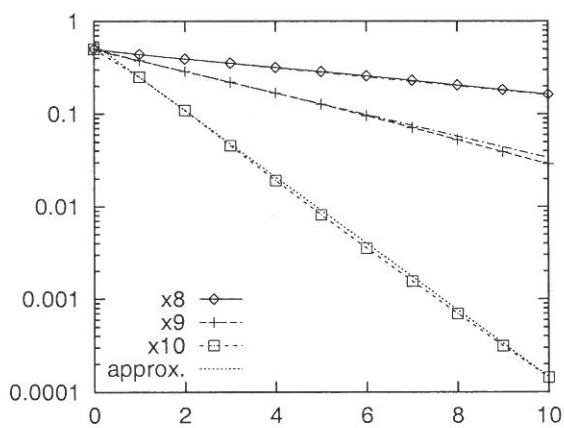

Fig. 1. Approximated vs. exact allele frequencies

7. Conditional Probability Distributions

If the index sets S_i overlap, then the analysis becomes much more difficult. The probability distribution is no longer the product of independent distributions. We have to introduce conditional probabilities.

In order to describe the problems associated with this factorization, we take as a simple example the index sets $\{1, 2\}$ and $\{2, 3\}$. The probability distribution factors into

$$p(x_1, x_2, x_3, t) = p(x_1, x_2, t)p(x_3|x_2, t). \tag{46}$$

As the factors and the product must be probability distributions, we have the constraints

$$p(0, 0, t) = 1 - p(0, 1, t) - p(1, 0, t) - p(1, 1, t) \tag{47}$$
$$\forall x_2 : p(0|x_2, t) = 1 - p(1|x_2, t). \tag{48}$$

p is determined by 5 parameters. A general distribution with 3 variables has $2^3 - 1$ parameters. So we have a slight reduction in the number of free parameters. Our goal is to calculate the response. We will omit the parameter t when the value is clear from the context.

Using Equations 1 and 2 we obtain

$$p(x_1, x_2, x_3, t+1) = p(x_1, x_2, x_3, t)\frac{f(x_1, x_2, x_3)}{\bar{f}(t)}. \tag{49}$$

These are eight equations. By suitable aggregation we can derive the difference equations for the five distributions defining the probability p. We formulate the result as a theorem.

Theorem 7. *The response to selection for a fitness function with sets* $\{1, 2\}$ *and* $\{2, 3\}$ *giving the factorization* $p = p(x_1, x_2, t)p(x_3|x_2, t)$ *is obtained by*

$$R(t) = \frac{V_c}{W} + \frac{1}{W^2}\sum_{v,w} p(v, t)F(v)p(w, t)F(w)\frac{\partial^2 W}{\partial v \partial w}$$

with

$$V_c = \sum_{x_1,x_2} p(x_1, x_2, t) F(x_1, x_2)^2 + \sum_{x_2,x_3} p(x_3|x_2, t) \frac{\bar{f}(x_2)}{W} F(x_3|x_2)^2$$

where v and w have to be chosen consistently for x_2 and be actual parameters, that is $v \in \{(0,1),(1,0),(1,1)\}$ and $w \in \{(1|0),(1|1)\}$.

Proof: Summing equations 49 and 46 for $x_3 = 0$ and $x_3 = 1$ we get

$$\begin{aligned} p(x_1, x_2, t+1) &= p(x_1, x_2, t) \frac{\bar{f}(x_1, x_2)}{W} \\ \bar{f}(x_1, x_2, t) &= \sum_{x_3} f(\mathbf{x}) p(x_3|x_2, t) \\ F(x_1, x_2, t) &:= \bar{f}(x_1, x_2, t) - W. \end{aligned} \quad (50)$$

This leads to

$$\begin{aligned} \Delta p(x_1, x_2) &= p(x_1, x_2, t+1) - p(x_1, x_2, t) \\ &= p(x_1, x_2, t) \frac{F(x_1, x_2, t)}{W}. \end{aligned} \quad (51)$$

The same procedure is done by inserting $x_1 = 0$ and $x_1 = 1$ yielding

$$\begin{aligned} &p(x_2, t+1) p(x_3|x_2, t+1) \\ &= p(x_3|x_2, t) \frac{1}{W} \Big[p(0, x_2, t) f(0, x_2, x_3) + \\ &\qquad p(1, x_2, t) f(1, x_2, x_3) \Big]. \end{aligned}$$

where $p(x_2, t)$ denotes the univariate marginal frequency of x_2. Using the above formulas, we get

$$\begin{aligned} p(x_2, t+1) &= \frac{\bar{f}(x_2, t)}{W} \\ \bar{f}(x_2, t) &:= \sum_{x_1,x_3} f(\mathbf{x}) p(\mathbf{x}, t). \end{aligned}$$

Obviously, $\bar{f}(x_2 = 0) + \bar{f}(x_2 = 1) = \bar{f}(t)$. Setting

$$\bar{f}(x_3|x_2, t) := \sum_{x_1} f(\mathbf{x}) p(x_1, x_2, t) \quad \text{and} \quad (52)$$

$$F(x_3|x_2, t) := \bar{f}(x_3|x_2, t) \frac{W}{\bar{f}(x_2, t)} - W \quad (53)$$

we have

$$p(x_3|x_2, t+1) = p(x_3|x_2, t) \frac{\bar{f}(x_3|x_2, t)}{W} \frac{W}{\bar{f}(x_2, t)}. \quad (54)$$

This leads to the difference equations

$$\begin{aligned} \Delta p(1|x_2) &= p(1|x_2, t+1) - p(1|x_2, t) \\ &= p(1|x_2, t) \frac{F(1|x_2, t)}{W}. \end{aligned}$$

It should be noted that this result is formally similar to the non-conditional one. But the term $F(x_3|x_2, t)$ is much more complex than the usual term $F(x_3, t)$. A tedious calculation finally gives an expression of the first derivatives of W and therefore the conjecture. □

Formally we have succeeded to obtain the exact response equation for a simple example with overlapping sets. The analysis can be extended to a chain of overlapping variables. But the interpretation and the calculation of the conditional terms is difficult. Moreover, if the sets S_i overlap in a more irregular manner, it can be very difficult to obtain a factorization of the probability at all. We will therefore investigate these problems numerically by simulation and by approximations derived from our theory.

8. The Factorized Distribution Algorithm

FDA is based on a factorization of the distribution $p(y, t)$. In its most general form we assume that

$$p(\mathbf{y}, t) = \prod_{i \in I} p(\mathbf{y}_{N_i}|\mathbf{y}_{R_i}, t) \quad N_i, R_i \subset \{1, \ldots, n\} \quad (55)$$

where $N_i \cup R_i = S_i$. We will not mathematically discuss the factorization problem here. This is done in (Mühlenbein et al., 1998). But the reader should get an idea, how to obtain the factorization from the examples we will discuss in detail. If the sets S_i are disjoint, then $N_i = S_i$ and $R_i = \emptyset$. The transformed fitness function is a generalized linear function.

Proportionate selection is well suited for mathematical analysis, but it is not suited for a practical optimization algorithm. It selects too

weakly in the final stages of the algorithm (Mühlenbein, 1998). Therefore FDA is normally used with truncation selection.

$$\text{FDA}_r$$

- **STEP 0:** Set $t \Leftarrow 1$. Generate $(1-r) * N \gg 0$ points randomly and $r * N$ points according to Equation 56.
- **STEP 1:** Selection (e.g. proportionate or truncation -select τN, $\tau < 1$ points) .
- **STEP 2:** Compute the conditional probabilities $p^s(\mathbf{y}_{N_i}|\mathbf{y}_{R_i}, t)$ with the selected points.
- **STEP 3:** Generate a new population according to $p(\mathbf{y}, t+1) = \prod_{i \in I} p^s(\mathbf{y}_{N_i}|\mathbf{y}_{R_i}, t)$
- **STEP 4:** If termination criteria is met, FINISH.
- **STEP 5:** Add the best point of the previous generation to the generated points (elitist).
- **STEP 6:** Set $t \Leftarrow t+1$. Go to STEP 2.

FDA is obviously an extension of UMDA, working in the original space. For separable functions both algorithms generate the same search points. FDA combines mutation and recombination of genetic algorithms into one operator using probability distributions. The factorization of the probability defined by Equation 56 can also be used at the initialisation. For faster convergence, a proportion of $r * N$ individuals can be generated with a local approximation of the conditional marginal distributions. This is done as follows:

$$p(\mathbf{y}_{N_i}|\mathbf{y}_{R_i}, t) = \frac{b^{g_i(\mathbf{y}_{N_i}, \mathbf{y}_{R_i})}}{\sum_{\mathbf{z}_{N_i}} b^{g_i(\mathbf{z}_{N_i}, \mathbf{y}_{R_i})}} \qquad (56)$$

with arbitrary b. The larger b, the "steeper" the distribution, whereas $b = 1$ yields a uniform distribution. b should be chosen so small that all $p(\mathbf{y}_{N_i}|\mathbf{y}_{R_i}, 0) > 0$.

FDA has a micro and a macro structure. If the subsets S_i are disjoint, the fitness function is just a generalized linear function with macro variables $\mathbf{X}_i$ which can be defined as the integer representation of S_i.

For proportionate selection we have analytically derived exact difference equations for the marginal distributions. For truncation selection we have not yet been able to compute exact difference equations. Therefore we will show the similarity between UMDA and FDA in the next section mainly by simulation.

9. Convergence Results for Truncation Selection

We have previously analyzed UMDA for the simple fitness function *OneMax* in detail. *OneMax* just counts the number of bits in the given string. This approximation will be of relevance for our FDA. Therefore we recall the theoretical results (Mühlenbein et al. 1994, Mühlenbein, 1998).

Assuming for all loci identical univariate marginal distributions for the initial population ($p_i(x_i = 1, t = 0) = p_0$), the difference equation for $p(t)$ can be obtained from the equation for the response to selection $R(t) = np(t+1) - np(t) \approx I * V(t)^{1/2}$. $V(t)$ denotes the variance of the population and I denotes the *selection intensity*. For *OneMax* we have $V(t) = np(t)(1-p(t))$. Therefore we obtain the difference equation for $p(t)$

$$p(t+1) = p(t) + \frac{I}{n}\sqrt{np(t)(1-p(t))}. \qquad (57)$$

From the difference equation a differential equation can be derived. This has the solution

$$p(t) = 0.5\left(1 + sin\left(\frac{I}{\sqrt{n}}t + arcsin(2p_0 - 1)\right)\right). \qquad (58)$$

This equation completely describes the dynamics of UMDA for *OneMax*.

For the sake of completeness we show for selected truncation thresholds the corresponding selection intensities (Mühlenbein, 1998) in the following table.

τ	0.75	0.5	0.25	0.125	0.04
I	0.42	0.8	1.27	1.65	2.14

Table 1. Selection intensity

Next we compute GEN_e, the number of generations until convergence. Convergence is achieved when the average fitness of the population is equal to the best fitness. GEN_e is obtained by setting $p(t_e) = 1$.

$$GEN_e = \left(\frac{\pi}{2} - arcsin(2p_0 - 1)\right)\frac{\sqrt{n}}{I}. \quad (59)$$

GEN_e depends on the size of the problem, the size of the population N and the truncation threshold τ. By looking at our results for proportionate selection, we expect that FDA and UMDA behave very similarly, also for truncation selection. Therefore we conjecture:

- GEN_e will be proportionate to $\sqrt{n}$.
- GEN_e will be inversely proportionate to I.

The second conjecture is a statistical property and has been confirmed in (Mühlenbein, 1998). The first conjecture is valid if the population size N is larger than a critical population size N^*, defined as the minimal population size needed to find the optimum with high probability, e.g. 99%. The determination of the critical population size N^* is very difficult. We have not yet succeeded with an analytical formula.

10. Numerical Results

We will first test the conjecture concerning the number of generations until equilibrium. For this we will use the following three functions

$$F1(X) = \sum_{i=1}^{n} x_i \qquad x_i \in \{0, 1\}$$

$$F2(X) = \sum_{i=1}^{l} f_2(x_i) \quad x_i \in \{0, 1, ..., 7\}$$

$$F3(X) = \sum_{i=1}^{l} f_3(x_i) \quad x_i \in \{0, 1, ..., 7\}.$$

For the function $F2$ we set $f_2(x_i)$ to the values of a *OneMax* function of order three. F_2 is thus identical to F_1. For F_3 we set $f_3(7) = 10$ and all other values to zero.

Given our theory, we expect the following results. GEN_e should be equal for $F1$ and $F2$.

n	$F1$	$F2$	F3	$GEN_e(59)$
30	7.0	7.0	6.2	7.2
60	10.0	10.0	9.0	10.1
90	12.2	12.3	11.0	12.4
120	14.2	14.4	12.9	14.4
GA_T	18.8	18.8	21.3	
150	16.0	16.3	14.1	16.0
180	17.2	17.8	15.9	17.5

Table 2. Generations until convergence, truncation threshold 0.3

GEN_e should be smaller for $F3$ because here FDA has only two main alternatives — $x_i = 7$ and all the rest.

The results from Table 2 confirm our prediction. Note how precisely Equation 59 predicts GEN_e obtained from actual simulation with FDA. GA_T is a genetic algorithm with truncation selection and uniform crossover. It needs slightly more generations for *OneMax*, as already mentioned in (Mühlenbein et al., 1994). For the function F_3 the genetic algorithm needs almost twice as many generations as FDA, which has knowledge about the micro-structure of F_3.

We will now turn to non-separable fitness functions. The following test suite is used.

Let u denote the number of bits turned on in the sub-string. Sub-function f^3_{dec} is defined as follows:

$$f^3_{dec} = \begin{cases} 0.9 & \text{for } u = 0 \\ 0.8 & \text{for } u = 1 \\ 0.0 & \text{for } u = 2 \\ 1.0 & \text{for } u = 3. \end{cases}$$

f^3_{dec} is used to define the separable *deceptive* function of order three

$$F_{DEC}(\mathbf{y}) = \sum_{i=1}^{l} f_{dec}(\mathbf{y}_{S_i}).$$

The factorization of the distribution is obvious, because the function is separable.

The next function is composed of two subfunctions:

$$f^l_1 = \begin{cases} l & \text{for } u = 0 \\ l-1 & \text{for } u = 3 \\ 0 & \text{for } else. \end{cases}$$

Function f_2^l has only one non zero element, $f_2^l(1,1,1) = l$. These two functions are used to define the function F_{O-PEAK}

$$F_{O-PEAK}(\mathbf{y}) = \sum_{i=1}^{l-1} f_1^l(\mathbf{y}_{S_i}) + f_2^l(\mathbf{y}_{S_l})$$

where $S_i \cap S_{i+1} = x_{2i+1}$. This function is non separable, it has a chain like interaction structure. The variables x_{2i+1} are contained in two sets.

This function is very difficult to optimize. The global optimum is $\mathbf{x} = (1,1,\ldots,1)$ with $F_{O-PEAK} = l(l-1)+1$. This optimum is triggered by f_2^l. It is very isolated, the second best optimum is given by $\mathbf{x} = (0,0,\ldots,0)$ with a function value of $l(l-1)$.

For this function a factorization with $N_i = S_i \setminus S_{i-1}, S_0 = \emptyset$ and $R_{i+1} = x_{2i+1}, R_1 = \emptyset$ will be used.

For numerical comparison we also consider the separable function

$$F_{CHAIN}(\mathbf{y}) = \sum_{i=1}^{l-1} f_1^l(\mathbf{y}_{S_i}) + f_2(\mathbf{y}_{S_l}),$$

where $S_i \cap S_{i+1} = \emptyset$.

The last example is a two dimensional Ising system.

$$F_{ISING}(\mathbf{y}) = \sum_i \sum_{j \in N(i)} J_{ij} s_i s_j$$

with $J_{ij} \in \{-1,1\}$ and $s_i \in \{-1,1\}$. The sum is taken over the four spatial neighbors $N(i)$, but each J_{ij} is used only once. We have used the following factorization for an $11 * 11$ grid. The spins are sequentially numbered from 1 till 121

$$p(\mathbf{s}) = p(s_1, s_2, s_{12}, s_{13}) p(s_3, s_{14} | s_2, s_{12}) * \ldots * p(s_{121} | s_{109}, s_{110}, s_{120}).$$

All distributions use four variables. Each variable appears exactly once at the left side from the conditional sign |. All grid interactions are covered by the factorization. For this class of Ising models exact solutions can be computed. We have computed a fairly difficult free boundary problem.

Table 3 gives the numerical results. In order to shorten the computation time, the runs have been stopped after the first occurrence of the optimum, after all individuals are equal, or after a specified maximum number of generations. *GEN* gives the generation count when stopped.

The surprising result is that *GEN* mainly depends on n, despite the great differences of the fitness functions. There is even not a large difference between F_{ISING} and the separable F_{CHAIN}, if the same number of variables is considered. For separable functions an initialization according to the marginal probabilities speeds up the convergence. This can be seen with F_{DEC}. For all functions *Equation 59 obtained for OneMax is a very good prediction for* GEN_e.

F	n	l	*Alg.*	*GEN*	*popsize*	*best*
F_{DEC}	90	30	$FDA_{0.0}$	11	1000	30
F_{DEC}	90	30	$FDA_{0.5}$	7	1000	30
F_{DEC}	90	30	GA_T	32	5000	27.1*
F_{CHAIN}	60	20	$FDA_{0.5}$	5	1000	400
F_{CHAIN}	90	30	$FDA_{0.5}$	7	1000	900
F_{CHAIN}	120	40	$FDA_{0.5}$	9	1000	1600
F_{CHAIN}	90	30	GA_T	25	5000	900
F_{O-PEAK}	51	25	$FDA_{0.0}$	6	1000	601
F_{O-PEAK}	101	50	$FDA_{0.0}$	9	2000	2451
F_{O-PEAK}	151	75	$FDA_{0.0}$	13	5000	5511
F_{O-PEAK}	101	50	GA_T	30	5000	2450*
F_{ISING}	121		$FDA_{0.5}$	11	1000	178
F_{ISING}	121		$FDA_{0.5}$	11	2000	178
F_{ISING}	121		GA_T	40	5000	174*

Table 3. Numerical Results, truncation threshold 0.3

GEN remains constant if the population size is larger than the critical population size. This is shown with the last entries of the table concerning the ISING model. The different complexity of the optimization problem is only reflected in the population size! The more difficult the function, the greater the population size has to be. The ISING model needs a surprising small population size, whereas the critical population size is very large for F_{O-PEAK}. For this function the critical population size increases dramatically. This has to to be expected.

For comparison we also note the results of a genetic algorithm GA_T with uniform crossover and truncation selection. The genetic algorithm is not able to solve the separable function F_{DEC} and the very difficult function F_{O-PEAK}. The solution of the 2-dimensional Ising model seems surprisingly simple. Even GA_T found the optimum once in 10 runs.

In all cases FDA outperforms the genetic algorithm by far, in quality of solution obtained and/or in number of function evaluations needed to obtain the optimum.

11. Conclusion

The factorized Distribution Algorithm FDA is an extension of UMDA for non separable ADFs. For separable functions it behaves exactly like an UMDA for functions with multiple alleles. For proportionate selection we have derived difference equations for the marginal frequencies. For truncation selection we confirmed numerically that the number of generations until convergence can be estimated by an equation derived for the simple linear *OneMax* function. If a certain percentage of the initial population is generated by using the factorization, GEN_e becomes even smaller.

We have shown that FDA is also very efficient for non separable ADFs. Here the corresponding factorization is given by conditional probabilities. There is almost no difference in the convergence speed of optimizing separable or non separable functions. The difficulty of the optimization problem is only reflected in the population size needed to obtain the optimum.

We do not want to give the impression that FDA should always be preferred to UMDA or genetic algorithms. In principle, UMDA can be seen as the simplest FDA implementation. The set of functions which can be solved by UMDA is surprisingly large. FDA should only be used for functions with a complex gene interaction structure. Furthermore the success of FDA critically depends on a correct probability model. This problem and constraint optimization problems are discussed by Mühlenbein et al. (1998).

References

[1] BAUM, L.E. & EAGON, J.A., (1967). An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. *Bull. Am. Math. Soc.*, 73:pp360–363

[2] FISHER, R. A., (1958). *The Genetical Theory of Natural Selection*. New York:Dover.

[3] GOLDBERG, D.E., (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Reading: Addison-Wesley.

[4] HOLLAND, J., (1992). *Adaptation in Natural And Artificial Systems*. Cambridge:MIT Press.

[5] MÜHLENBEIN, H. & SCHLIERKAMP-VOOSEN, D., (1994). The science of breeding and its application to the breeder genetic algorithm. *Evolutionary Computation*, 1:pp. 335–360.

[6] MÜHLENBEIN, H., (1998). The Equation for Response to Selection and its Use for Prediction. *Evolutionary Computation*, 5:pp. 303–346.

[7] MÜHLENBEIN, H. & MAHNIG, TH. & OCHOA, A., (1998). Schemata, Distributions and Graphical Models in Evolutionary Optimization *submitted for publication*, http://set.gmd.de/AS/ga/publineu.html

Received: June, 1998
Accepted: December, 1998

Contact address:
Heinz Mühlenbein and Thilo Mahnig
RWCP Theoretical Foundation GMD Laboratory
GMD - Forschungszentrum Informationstechnik
53754 St. Augustin
e-mail: muehlenbein@gmd.de

DR. HEINZ MUEHLENBEIN is currently head of the research group Adaptive Systems at the GMD, St. Augustin, Germany. He has published research work in the areas of computer networks, parallel processing, evolutionary algorithms, neural networks and robotics. He is the European editor of the journal "Evolutionary Computation" and an editor of "Journal of Heuristics".

THILO MAHNIG received the diploma in mathematics from the University of Bonn in differential geometry in 1996. He is currently a PhD student at GMD - German National Research Center for Information Technology, St. Augustin. His research interest is in the theory of population-based optimization algorithms.