A Theoretical and Experimental Investigation of Estimation of Distribution Algorithms

Heinz Mühlenbein and Robin Höns

Fraunhofer Institute for Autonomous Intelligent Systems,
53754 Sankt Augustin, Germany,
{heinz.muehlenbein,robin.hoens}@ais.fraunhofer.de

Abstract. Estimation of Distribution Algorithms (EDA) have been proposed as an extension of genetic algorithms for optimization. In this paper the major design issues are presented within a general interdisciplinary framework. It is shown that EDA algorithms compute *maximum entropy* or *minimum relative entropy* approximations. A special structure learning algorithm *LFDA* is analyzed in detail. It is based on a finite minimum log-likelihood ratio principle. We investigate important parameters of the presented EDA algorithms by analyzing the performance on synthetic benchmark functions.

1 Introduction

The *Estimation of Distribution* (EDA) family of population based search algorithms was introduced in [12] as an extension of genetic algorithms.[1] The following observations lead to this proposal. First, genetic algorithms have difficulties to optimize deceptive and non-separable functions, and second, the search distributions implicitly generated by recombination and crossover can be extended to include the correlation of the variables in samples of high fitness values.

EDA uses probability distributions derived from the function to be optimized to generate search points instead of crossover and mutation as done by genetic algorithms. The other parts of the algorithms are identical. In both cases a population of points is generated and points with good fitness are selected either to estimate a search distribution or to be used for crossover and mutation.

In [12] the distribution has been estimated by computationally intensive Monte Carlo methods. The distribution was restricted to tree-like structures. It has been shown in [11] that simpler and more effective methods exist which use a general factorization of the distribution.

The family of EDA algorithms can be understood and further developed without the background of genetic algorithms. The problem to estimate empirical distributions has been investigated independently in several scientific disciplines. A discussion of the different approaches in statistics, belief networks, and statistical physics can be found in [5].

[1] In [12] they have been named *conditional distribution algorithms.*

Today two major branches of EDA can be distinguished. In the first branch the factorization of the distribution is computed from the structure of the function to be optimized, in the second one the structure is computed from the correlations of the data. The second branch has been derived from the theory of belief networks [2]. For large real life applications often a hybrid between these two approaches is most successful [8].

The paper is intended as a short introduction to the theory of EDA. We will only consider binary variables. It is not intended as a survey of ongoing research. Here an excellent overview is already available [3].

The outline of the paper is as follows. In section 2 the basic steps to derive the Factorized Distribution Algorithm are recapitulated. A factorization theorem will be discussed which uses the structure of the function to be optimized to factor the distribution. In section 3 the problem addressed by the factorization theorem is generalized. We introduce the principles *minimum relative entropy* and *minimum log-likelihood ratio*. In section 4 the learning of models from samples of high fitness values is described. Important parameters of the presented EDA algorithms are investigated in section 5.

2 Factorization of the Search Distribution

EDA has been derived from a search distribution point of view. We just recapitulate the major steps published in [11, 7–9].

Let a function $f : X \to \mathbb{R}_{\geq 0}$ be given. We consider the optimization problem

$$\mathbf{x}_{opt} = \operatorname{argmax} f(\mathbf{x}) \tag{1}$$

A good candidate for optimization using a search distribution is the Boltzmann distribution.

Definition 1 *For $\beta \geq 0$ define the* Boltzmann distribution[2] *of a function $f(\mathbf{x})$ as*

$$p_\beta(\mathbf{x}) := \frac{e^{\beta f(\mathbf{x})}}{\sum_y e^{\beta f(\mathbf{y})}} =: \frac{e^{\beta f(\mathbf{x})}}{Z_f(\beta)} \tag{2}$$

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

2.1 Factorization of the distribution

In this section an efficient numerical algorithm is derived if the fitness function is additively decomposed.

[2] The Boltzmann distribution is usually defined as $e^{-\frac{E(\mathbf{x})}{T}}/Z$. The term $E(x)$ is called the energy and $T = 1/\beta$ the temperature. We use the inverse temperature β instead of the temperature.

Definition 2 *Let $s_1, \ldots, s_m$ be index sets, $s_i \subseteq \{1, \ldots, n\}$. Let f_i be functions depending only on the variables x_j with $j \in s_i$. Then*

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x}_{s_i}) \tag{3}$$

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

We need the following sets:

Definition 3 *Given $s_1, \ldots, s_m$, we define for $i = 1, \ldots, m$ the sets d_i, b_i and c_i:*

$$d_i := \bigcup_{j=1}^{i} s_j, \qquad b_i := s_i \setminus d_{i-1}, \qquad c_i := s_i \cap d_{i-1} \tag{4}$$

We set $d_0 = \emptyset$.

From the additive decomposition of the function we can construct a *graphical model* by connecting those variables which are contained in the same subfunction. In the theory of decomposable graphs, d_i are called *histories*, b_i *residuals* and c_i *separators* [4]. In [11] we have proven the following theorem.

Theorem 4 (Factorization Theorem). *Let $p_\beta(\mathbf{x})$ be a Boltzmann distribution with*

$$p_\beta(\mathbf{x}) = \frac{e^{\beta f(\mathbf{x})}}{Z_f(\beta)} \tag{5}$$

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

$$b_i \neq \emptyset \quad \forall i = 1, \ldots, m; \quad d_m = \{x_1, \ldots, x_n\}, \tag{6}$$

$$\forall i \geq 2 \; \exists j < i \;\; \textit{such that } c_i \subseteq s_j \tag{7}$$

then

$$p_\beta(\mathbf{x}) = \prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i} | \mathbf{x}_{c_i}) = \frac{\prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})}{\prod_{i=2}^{m} p_\beta(\mathbf{x}_{c_i})} \tag{8}$$

Definition 5 *The constraint defined in* (7) *is called the* **running intersection property** *(RIP). The factorization is polynomially bounded (PBF) if the size of the sets $\{b_i, c_i\}$ is bounded by a constant independent of n.*

The algorithm *FDA* uses a factorization and estimates the unknown marginals from samples. For the class of PBFs fulfilling the *RIP* the algorithm will converge to the optimum, but convergence to the optimum will depend on the size of the sample. The necessary size of the sample is smaller if a number of steps with low selection is used instead of just one step using strong selection.

Algorithm 1: *FDA* – Factorized Distribution Algorithm

1. Calculate b_i and c_i from the decomposition of the function.
2. $t \Leftarrow 1$. Generate an initial population with N individuals from the uniform distribution.
3. **do** {
4. Select $M \leq N$ individuals
5. Estimate the conditional probabilities $p(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}, t)$ from the selected points.
6. Generate new points according to $p(\mathbf{x}, t+1) = \prod_{i=1}^{m} p(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}, t)$.
7. $t \Leftarrow t + 1$.
8. } **until** (stopping criterion reached)

FDA has experimentally proven to be very successful on a number of functions where standard genetic algorithms fail to find the global optimum. In [6] the scaling behavior for various test functions has been studied. For recent surveys the reader is referred to [8, 10].

Optimization problems which have a polynomially bounded factorization fulfilling *RIP* can provably be solved in polynomial time. This is a *sufficient condition*, not a *necessary condition*. Many problems do not admit a PBF fulfilling *RIP*, but an approximate factorization might still lead to the optimum.

Conjecture: *In the class of non-polynomially bounded problems there exist instances which can only be solved in exponential time. But the number of instances which can be solved polynomially seems to be very large.*

3 Minimizing the Kullback-Leibler divergence

The factorization theorem provides the marginals needed for an exact factorization. Thus the given distribution will be exactly reproduced. The estimation problem can be generalized to any *given* set of marginals.

Problem
Given a set of marginal distributions $p(\mathbf{x}_{s_i})$ from an unknown Boltzmann distribution, compute a distribution which satisfies the marginals.

Among the possible solutions of this problem, a common choice is the distribution which maximizes the entropy. Let us recall

Definition 6 *The entropy [1] of a distribution is defined by*

$$H(p) = -\sum_{x} p(\mathbf{x}) \ln(p(\mathbf{x})) \tag{9}$$

Maximum entropy principle (MaxEnt): *Find the maximum entropy distribution for $p(\mathbf{x})$ which satisfies the given marginals.*

The MaxEnt solution is unique if the given marginal distributions fulfill the requirements of probability theory [5]. If there exists some information about the

target distribution, then we might want to approximate the target distribution. This can be achieved as follows.

Definition 7 *The Kullback-Leibler divergence (KLD) between two distributions is defined by*

$$KLD(q||p) = \sum_{\mathbf{x}} q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} \tag{10}$$

Note that KLD is not symmetric! Thus we have two choices for minimization.

Minimum relative entropy principle (MinRel) *Given a set of consistent marginal distributions, find the distribution q which minimizes $KLD(q||p)$ to the target distribution $p(\mathbf{x})$.*

In [1] (p. 18) $KLD(p||q)$ is called the expected logarithm of the *likelihood ratio*. It is a measure of the inefficiency of assuming q when the true distribution is p. It is connected to the description length. If we knew p we could construct a code with average description length $H(p)$. If, instead, we used the code for distribution q, we would need $H(p) + KLD(p||q)$ bits on the average to describe the random variable. Thus the following principle is also justified:

Minimum expected log-likelihood ratio principle (MinLike) *Given a set of consistent marginal distributions, find the distribution q which minimizes $KLD(p||q)$ to the target distribution $p(\mathbf{x})$.*

If p is the uniform random distribution, then MinLike minimizes $\sum_x \ln q(\mathbf{x})$. This is not the entropy of $q(\mathbf{x})$. The MinLike principle will be used for structure learning of Bayesian networks.

4 Learning a Bayesian network from data

This section will be very brief, compared to the difficulty of the subject. An excellent in-depth discussion can be found in [3]. We will just motivate some of the major design decisions.

First we simplify the notation. Capital letters denote variables, lower case letters instances of the variables. Using simple rules of probability one can show that any factorization can be written as a Bayesian network

$$q(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\pi_i) \tag{11}$$

Π_i are called the parents of X_i. If the running intersection property is fulfilled, the Bayesian network is *singly connected*. If the number of the parents $|\Pi_i|$ is bounded by a constant independent from n, we say the Bayesian network is polynomially bounded (PBB).

Both the MaxEnt and the MinRel principle assume that a fixed set of marginal distributions is given. But if the data is provided by a numerical sample, we can choose which marginal distributions should be used in order to obtain a Bayesian network which reproduces the data accurately.

Thus we have to deal now with the problem *how to choose the appropriate marginal distributions.* This problem can be solved in the following way. Let Q be the set of all distributions $q(\mathbf{x})$ for the Bayesian networks considered. We introduce the average of $\ln q$ over the true distribution p

$$E(\ln q) = \sum_{\mathbf{x}} p(\mathbf{x}) \ln q(\mathbf{x}) \tag{12}$$

We have

$$E(\ln q) = -H(p) - KLD(p||q) \tag{13}$$

Remark: The minimization of $KLD(p||q)$ in Q is equivalent to maximization of $E(\ln q)$.

Theorem 8. *For the distribution* $q(x) = \prod_{i=1}^{n} q(x_i|\pi_i)$ *we have*

$$E(\ln q) = \sum_{i=1}^{n} \sum_{x_i, \pi_i} p(x_i, \pi_i) \ln q(x_i|\pi_i) \tag{14}$$

Proof.

$$\begin{aligned} \sum_{\mathbf{x}} p(\mathbf{x}) \ln q(\mathbf{x}) &= \sum_{\mathbf{x}} p(\mathbf{x}) \sum_{i=1}^{n} \ln q(x_i|\pi_i) \\ &= \sum_{i=1}^{n} \sum_{x_i, \pi_i} p(x_i, \pi_i) \ln q(x_i|\pi_i) \end{aligned}$$

Equation (14) can be approximated using a finite sample. We introduce the following notation. Let N denote the size of the sample $\mathcal{X}$. Let $N(x_i, \pi_i)$ denote the number of instances with $X_i = x_i$ and $\Pi_i = \pi_i$, where the states of Π_i are numbered $1 \leq \pi_i \leq 2^{|\Pi_i|}$. Let $N(\pi_i) = \sum_{x_i} N(x_i, \pi_i)$. We can now approximate

$$E(\ln q) \approx L(q|\mathcal{X}) := \sum_{i=1}^{n} \sum_{x_i, \pi_i} \frac{N(x_i, \pi_i)}{N} \ln \frac{N(x_i, \pi_i)}{N(\pi_i)} \tag{15}$$

Thus we have arrived at the following principle

Finite sample MaxLike principle (FinMaxLike)
Maximize in the class of Bayesian networks Q

$$\max_{q \in Q} L(q|\mathcal{X}) = \max_{q \in Q} \sum_{i=1}^{n} \sum_{x_i, \pi_i} \frac{N(x_i, \pi_i)}{N} \ln \frac{N(x_i, \pi_i)}{N(\pi_i)} \tag{16}$$

Remark: When an edge $i_1 \to i_2$ is added, $L(q|\mathcal{X})$ is increased by

$$I(X_{i_1}, X_{i_2}|\Pi_{i_2}) = \sum_{x_{i_1}, x_{i_2}, \pi_{i_2}} \frac{N(x_{i_1}, x_{i_2}, \pi_{i_2})}{N} \ln \frac{N(x_{i_1}, x_{i_2}, \pi_{i_2}) N(\pi_{i_2})}{N(x_{i_1}, \pi_{i_2}) N(x_{i_2}, \pi_{i_2})} \tag{17}$$

which is called the *conditional mutual information* of X_{i_1} and X_{i_2}, given Π_{i_2} [1]. Since this is always non-negative, FinMaxLike does not prefer exact models of small complexity (a small number of connections) compared to exact models of large complexity. Thus a criterion is urgently needed with the following property: It combines maximizing the log-likelihood with minimizing the complexity.

There have been many proposals for such a criterion. We just discuss the popular criterion derived by [13]. It has been also called the *Bayesian Information Criterion* [2].

Definition 9 *Let V be the number of free parameters in the marginal distributions of the graphical model q. Then the weighted BIC measure is defined by*

$$BIC_\alpha = N * L(q|\mathcal{X}) - \alpha \ln N * V \tag{18}$$

It has been shown that BIC is asymptotically equivalent to the minimum description length. [13] computed $\alpha = 0.5$ as the best weighting factor for $N \to \infty$.

The BIC criterion can be used to incrementally construct a Bayesian network starting from the empty network. In most programs a simple greedy hill climbing heuristic is used, which chooses the edge maximizing (18). This gives the algorithm *LFDA* (Learning Factorized Distribution Algorithm). For details, the reader is referred to [6, 3].

5 How to test EDA algorithms

EDA algorithms are complex stochastic programs. They have to be tested in a number of carefully selected steps. In our opinion most researchers developing EDAs have concentrated so far on the benchmark method to show the power of EDA algorithms. A popular benchmark or a difficult function is taken and the success of the optimization algorithm is shown. The success rate is the percentage of runs computing the optimum. There is no detailed discussion why the algorithms work. The internal behavior of the algorithm (e.g. which Bayesian network it has constructed etc.) is not reported. With the benchmark approach a generalization of the results is difficult.

We propose that EDA algorithms should be tested in carefully selected steps instead – starting from theoretically understood problems to more complex ones. Both *FDA* and *LFDA* depend on parameters. For *FDA* these are the size of the population N, the selection strength (for truncation selection this is the selection threshold $\tau = M/N$, where M is the number of selected individuals), and the Bayesian hyper-parameter ρ. *LFDA* has in addition the structure penalty factor α. We evaluate the algorithms and the parameters with a set of synthetic fitness functions. The first three functions are separable of order 5. That means that they consist of $m = n/5$ disjunct blocks of size 5. Thus we have

$$F_l(x) = \sum_{i=1}^{m} f_l(x_{5i-4}, \ldots, x_{5i}) \qquad (l = 1, 2, 3)$$

The first two sub-functions are defined as follows

$$f_1(x_1,x_2,x_3,x_4,x_5) = \begin{cases} 1 & \iff (x_1,x_2,x_3,x_4,x_5) = (1,1,1,1,1) \\ 0 & \text{otherwise} \end{cases} \tag{19}$$

$$f_2(x_1,x_2,x_3,x_4,x_5) = \begin{cases} 1 & \iff (x_1,x_2,x_3,x_4,x_5) = (1,1,1,1,1) \\ 0.9 & \iff (x_1,x_2,x_3,x_4,x_5) = (0,0,0,0,0) \\ 0 & \text{otherwise} \end{cases} \tag{20}$$

The third function is deceptive.

$$f_{\text{dec}}(x_1,x_2,x_3,x_4,x_5) = \begin{cases} 0.9 & \iff \sum x_i = 0 \\ 0.8 & \iff \sum x_i = 1 \\ 0.7 & \iff \sum x_i = 2 \\ 0.6 & \iff \sum x_i = 3 \\ 0.0 & \iff \sum x_i = 4 \\ 1.0 & \iff \sum x_i = 5 \end{cases} \tag{21}$$

The fourth and the fifth function are non-separable. They consist of m overlapping blocks of size 3 ($n = 2m+1$). We have a different function for the last block.

$$F_l(\mathbf{x}) := \sum_{i=1}^{m-1} f_l(x_{2i-1},x_{2i},x_{2i+1}) + g_l(x_{2m-1},x_{2m},x_{2m+1}) \qquad (l=4,5) \tag{22}$$

The fourth function *IsoPeak* is defined by

$$f_4(x,y,z) = \begin{cases} m & \iff (x,y,z) = (0,0,0) \\ m-1 & \iff (x,y,z) = (1,1,1) \\ 0 & \text{otherwise} \end{cases} \tag{23}$$

$$g_4(x,y,z) = \begin{cases} m & \iff (x,y,z) = (1,1,1) \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

The fifth function is the most difficult to optimize.

$$f_5(x,y,z) = \begin{cases} m & \iff (x,y,z) = (0,0,0) \\ m-3 & \iff (x,y,z) = (1,0,0) \text{ or } (0,1,0) \\ m-1 & \iff (x,y,z) = (1,1,1) \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

$$g_5(x,y,z) = \begin{cases} m & \iff (x,y,z) = (0,0,0) \\ 2m+5 & \iff (x,y,z) = (1,1,1) \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

Table 1. Population size for which the optimum is found with 100% in 20 runs. ρ denotes use of the Bayesian hyper-parameter (29). The selection threshold for truncation selection was set to $\tau = 0.3$, except * 0.1, † 0.7, ‡ 0.95. For missing values, the required population size is too large.

Alg	n	F_1	F_1 ρ	F_2	F_2 ρ	F_{dec}	F_{dec} ρ	F_4	F_4 ρ	F_5	F_5 ρ
FDA	25	200	50	200	50	400	50	250	1200	500	3600
FDA	50	200	50	200	50	600	100	700	4000*	3500	–
FDA	100	500	100	200	100	800	100	1500	–	15000‡	–
FDA	200	700	200	700	200	1200	300	3500†	–	–	–
LFDA	25	400	300	500	300	3500*	500*	900	2000	1500	6000
LFDA	50	800	400	700	500	18000*	17000*	5000*	12000*	50000	–
UMDA	25	70	150	500*	600*	–	–	–	–	–	–
UMDA	50	100	200	1600*	1300*	–	–	–	–	–	–
UMDA	100	200	330	4200*	3300*	–	–	–	–	–	–
UMDA	200	400	600	–	–	–	–	–	–	–	–

The function F_5 combines deception and isolation. The global optimum is at $\mathbf{x} = (1, 1, \ldots, 1)$, but the second largest values have a Hamming distance of $n-3$ or $n-4$ to the optimum. The third largest values have a distance of $n-5$ or $n-6$. Thus the global optimum is extremely isolated. This problem poses a challenge to any iterative optimization algorithm. The algorithms are attracted by the second largest optimum. But from this region there exists no path to the global optimum. All of the 0's have to be flipped together in order to jump to the global optimum. Any randomized local optimization procedure will need an exponential time to find the global optimum. Nevertheless, we know from the factorization theorem that *FDA* will compute the optimum in polynomial time.

In table 1 we show general results for some EDAs. *UMDA* uses only univariate marginals for the factorization. The *selection threshold* τ is defined by $M = \tau N$ where M is the number of selected points.

5.1 The Bayesian hyper-parameter ρ

Let the marginal distribution $p(\mathbf{x})$ to be estimated from a population of size M. Let $\mathbf{x} = (x_1, \ldots, x_k)$ and $N(\mathbf{x})$ the number of individuals with configuration $\mathbf{x}$. Then in the Bayesian framework

$$\hat{p}(\mathbf{x}) = \frac{N(\mathbf{x}) + \rho}{M + 2^k \rho} \tag{27}$$

is used with some $\rho > 0$. For $\rho = 0$ we have the conventional maximum likelihood estimator. The question is how to set ρ. We discuss the problem with a specific example – the factorization consists of m blocks of size k, and $m-1$ blocks are correct, in the last block all the bits in the whole population are incorrect. In order to obtain the optimum we would like to maximize the probability to flip

the incorrect block, while leaving all the others intact:

$$r(\rho) = \frac{\rho}{M + 2^k \rho} \left(\frac{M + \rho}{M + 2^k \rho} \right)^{m-1} \longrightarrow \max \tag{28}$$

Setting the derivative with respect to ρ equal to zero, we get

$$\rho_{\max} = \frac{M}{(2^k - 1)(m - 1) - 1} \tag{29}$$

The above value has been derived under severe assumptions. We see in Table 1 that it works fine for the separable functions F_1, F_2, F_{dec}, but not at all for the overlapping functions.

Nevertheless it gives some indication about the possible range. The derivation of a hyper-parameter for conditional distributions is much more difficult.

5.2 The penalty weight α

Schwarz [13] has computed an optimal penalty factor $\alpha = 0.5$ under severe assumptions. (One of the assumptions is $N \to \infty$.) Since we are using fairly small population sizes, we investigate the influence of α on the computed network in the neighborhood of $\alpha = 0.5$. In the first test we generate uniform random data. In this case the exact network has no edges at all. Table 2 shows empirical results. How is an optimal α defined? It is obvious that no edges will be generated for a large α. For very small α many edges will be generated. Thus we are looking for a value of α at the transition between these two regimes. Loosely speaking, we look for α_{tr} with $\#edges \leq 5$ for $\alpha > \alpha_{\text{tr}}$, $4 \leq \#edges \leq 10$ for $\alpha = \alpha_{\text{tr}}$ and $\#edges > 10$ for $\alpha < \alpha_{\text{tr}}$.

Table 2. Number of edges added by *LFDA* for a uniform random data set (average over ten runs).

α	n	N	$\#edges$	n	N	$\#edges$	n	N	$\#edges$
1.00	25	200	0.3	50	400	0.4	100	800	0.8
0.75	25	200	1.5	50	400	3.4	100	800	6.5
0.50	25	200	7.1	50	400	17.2	100	800	45.8
0.25	25	200	38.4	50	400	89.4	100	800	197.6
0.10	25	200	113.1	50	400	254.7	100	800	536.5
0.50	25	10000	0.5	50	10000	4.3	100	10000	10.9

The results of table 2 suggest that a value of $\alpha_{\text{tr}} = 0.75$ fulfills the requirements for reasonably large population sizes. For very large population sizes $\alpha_{\text{tr}} = 0.5$ might be indeed the best value.

Next we investigate populations generated by *LFDA*. We take the separable function F_{dec} as example. Instead of reporting the number of found edges, we

first investigate the detection of dependent variables after the first three selection steps. We tried two criteria, the (unconditional) mutual information (17), and the Chi-Square test. The results were very similar. In Table 3 we show for the first three generations how many of the twenty edges with the largest values of (17) are correct, i. e. in the same block.

Table 3. Number of correct edges within the twenty edges with biggest mutual information. Values for the first three generations, averaged over five runs.

		$\alpha = 0.25$				$\alpha = 0.5$			
n	N	$\tau = 0.1$			$\tau = 0.3$	$\tau = 0.1$			$\tau = 0.3$
25	1000	11.2	19.4	20.0	3.6 5.4 7.2	11.4	17.8	19.4	3.0 2.2 3.4
50	1000	1.6	1.8	4.0	1.8 1.2 1.6	2.6	3.2	7.0	1.8 1.0 1.8
50	5000	3.8	15.2	20.0	1.2 2.0 4.6	4.8	7.2	11.6	2.8 4.6 4.8

Table 4. Number of correct edges (total number of edges) in the graph for the first three generations of *LFDA*, running on F_{dec} with $n = 50$.

N	$\alpha = 0.25$, $\tau = 0.1$	$\alpha = 0.25$, $\tau = 0.3$	$\alpha = 0.5$, $\tau = 0.1$
1000	18 (135), 23 (134), 20 (112)	6 (87), 10 (104), 11 (110)	2 (38), 5 (57), 9 (62)
5000	25 (98), 57 (119), 63 (122)	13 (74), 21 (83), 26 (95)	10 (25), 15 (39), 21 (43)

In Table 4 we give for the first three generations the number of correct edges added to the graph. For F_{dep} with $n = 50$ there are 100 correct edges.

The results are very disappointing. A reasonable number of correct edges is computed for large population sizes only. But all structure learning algorithms depend on the correct computation of the dependent variables! It seems that for large population sizes a smaller value of α is favorable.

This problem needs further investigation. Maybe the introduction of local hill-climbing algorithms will bring improvements. These take the generated individuals and attempt to improve them by searching for better values in their neighborhood. Then the space of the local maxima is much smaller than the original search space. Thus structure learning algorithms should be able to compute the dependencies more easily. First results in [8] confirm this conjecture.

6 Conclusion and Outlook

In this paper we have investigated some EDA algorithms for optimization. The efficient estimation and sampling of distributions is a common problem in several scientific disciplines. The different approaches have been discussed in more detail in [5]. We have identified two principles used for the estimation – minimum relative entropy and minimum expected log-likelihood ratio. If p is the distribution to be estimated, then MinRel minimizes $KLD(q||p)$ whereas MinLike minimizes $KLD(p||q)$.

The structure learning algorithms have problems to detect the correct dependencies of the variables in reasonably large data sets. Thus the efficiency of such algorithms is low. Here the introduction of local hill-climbing processes promises improvements.
Our software can be downloaded from our web site
http://www.ais.fraunhofer.de/~muehlen/.

References

1. Th. M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley, New York, 1989.
2. M. I. Jordan. *Learning in Graphical Models.* MIT Press, Cambrigde, 1999.
3. P. Larrañaga and J.A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Optimization.* Kluwer Academic Press, Boston, 2001.
4. S. L. Lauritzen. *Graphical Models.* Clarendon Press, Oxford, 1996.
5. H. Mühlenbein and R. Höns. Estimation of distributions and maximum entropy. *Evolutionary Computation*, 2004.
6. H. Mühlenbein and Th. Mahnig. FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376, 1999.
7. 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, Berlin, 2000.
8. H. Mühlenbein and Th. Mahnig. Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning. *Journal of Approximate Reasoning*, 31(3):157–192, 2002.
9. H. Mühlenbein and Th. Mahnig. Mathematical analysis of evolutionary algorithms. In C. C. Ribeiro and P. Hansen, editors, *Essays and Surveys in Metaheuristics*, Operations Research/Computer Science Interface Series, pages 525–556. Kluwer Academic Publisher, Norwell, 2002.
10. H. Mühlenbein and Th. Mahnig. Evolutionary algorithms and the Boltzmann distribution. In K. De Jong, R. Poli, and J. C. Rowe, editors, *Foundations of Genetic Algorithms 7*, pages 525–556. Morgan Kaufmann Publishers, San Francisco, 2003.
11. H. Mühlenbein, Th. Mahnig, and A. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.
12. H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. binary parameters. In H.-M Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Lecture Notes in Computer Science 1141: Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, Berlin, 1996. Springer-Verlag.
13. G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 7:461–464, 1978.