Convergence Theorems of Estimation of Distribution Algorithms

Heinz Mühlenbein heinz.muehlenbeinr@online.de
Fraunhofer Institut IAIS, Schloss Birlinghoven, 53754 Sankt Augustin, Germany

Abstract

Estimation of Distribution Algorithms (EDAs) have been proposed as an extension of genetic algorithms. We assume that the function to be optimized is additively decomposed (ADF). The interaction graph of the ADF is used to create exact or approximate factorizations of the Boltzmann distribution. Convergence of the algorithm MN-GIBBS is proven. MN-GIBBS uses a Markov network easily derived from the ADF and Gibbs sampling. The Factorized Distribution Algorithm (FDA) uses a less general representation , a Bayesian network and probabilistic logic sampling (PLS). We shortly describe the algorithm LFDA which learns a Bayesian network from data. The relation between the network computed by LFDA and the optimal network used by FDA is investigated. Convergence of FDA to the optima is shown for finite samples if the factorization fulfills the running intersection property. The sample size is bounded by $O(nm \ln nm)$ where n is the size of the problem and m the number of sub-functions. For the proof results from statistical learning theory and Probably Approximately Correct (PAC) learning are used. Numerical experiments show that even for difficult test functions a sample size which scales linearly with n is often sufficient. We also show that a good approximation of the true distribution is not necessary, it suffices to use a factorization where the global optima have a large enough probability. This explains the success of EDAs in practical applications.

Keywords

Genetic algorithms, Bayesian networks, Markov networks, Boltzmann distribution, Gibbs sampling, PAC learning

1 Introduction

The *Estimation of Distribution Algorithm* (EDA) family of population based optimization algorithms was introduced by Mühlenbein and Paaß (1996) as an an extension of genetic algorithms. They address the problem that the search distributions implicitly generated by genetic algorithms through recombination and crossover do not exploit the correlation of the variables in samples of high fitness values. Therefore genetic algorithms have difficulties in solving these problems.

EDAs use 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 used and points with good fitness are selected either to estimate a search distribution or to be used for crossover and mutation.

Today two major branches of EDAs can be distinguished. In the first branch a factorization of the distribution is computed from the mathematical expression of the function to be optimized, in the second one the factorization is computed from the

©2009 by the Massachusetts Institute of Technology

correlations of the variables in samples of points with high fitness (learning). Most researchers concentrate on learning. But here it is difficult to prove convergence to the optima. Therefore the papers mainly contain experimental results with some theoretical extrapolations. This is unfortunate, because for EDAs using optimal graphical models lots of theoretical results are available. In any case, a comparison between the results of EDAs using learning with the corresponding algorithm using an optimal graphical model should always be made.

In this paper we first analyze EDAs where the graphical model is derived from the structure of the function. Two different graphical representations are investigated, *Markov networks* and the more restricted class of *Bayesian networks.* For both representations algorithms are presented, the Markov network algorithm MN-GIBBS and the Factorized Distribution Algorithm FDA. We then discuss algorithms learning the graphical model.

The outline of the paper is as follows. In section 2 we introduce the Boltzmann distribution. Then factorizations of the distribution are discussed. For additively decomposed functions (ADFs) a factor graph (Kschischang et al., 2001) can easily be computed. The factor graph defines a Markov network. The algorithm MN-GIBBS is investigated which uses *Gibbs sampling* to generate samples of the distribution. A general convergence theorem is proven for MN-GIBBS. Gibbs sampling is computationally expensive, so in section 3 the well-known FDA is shortly reviewed. FDA uses Bayesian networks as graphical models which can be sampled by *probabilistic logic sampling*. In section 4 we describe some algorithms which learn a graphical models from a sample of promising points.

In section 5 we derive an upper bound on the sample size needed for FDA to converge to the global optima. The bound is derived using the theory developed in *Probably Approximately Correct* (PAC) learning (Kearns and Vazirani, 1994).

The similarity between the factorizations computed by FDA and the learning algorithm LFDA is numerically investigated in section 6. We discuss what characterizes a good factorization for EDAs. Numerical results are presented in section 7.

A good introduction to early EDA research can be found in the book of Larrañaga and Lozano (2002). This paper concentrates on convergence theorems for EDAs. It also covers important design issues for EDAs. From theoretical considerations the algorithm MN-GIBBS is highly recommended.

There exist a huge literature about Markov networks and PAC learning. We assume that the reader is familiar with these subjects. This paper is intended to provide understanding and insight, therefore many examples are included. Proofs are omitted if they are only technical or can be found in easily accessible papers.

2 Convergence theory for infinite samples

We will use in this paper the following notation. Capital letters denote variables, lower cases instances of variables. If the distinction between variables and instances is not necessary, we will use lower case letters. Vectors are denoted by $\mathbf{x}$, a single variable by x_i. We consider discrete variables only.
Let a function $f : \mathcal{X} \to \mathrm{I\!R}$ be given. We consider the discrete 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 $\beta \geq 0$ the* Boltzmann distribution[1] *of a function $f(\mathbf{x})$ is defined as*

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

where $Z_f(\beta)$ is the partition function.

The Boltzmann distribution concentrates with increasing β around the global optima of the function. Obviously, the distribution converges for $\beta \to \infty$ to a distribution where only the optima have a probability greater than 0 (Mühlenbein and Mahnig, 2002b). Therefore, if it were possible to sample efficiently from this distribution for arbitrary β, optimization would be an easy task. But the computation of the partition function usually needs an exponential effort for a problem of n variables. We have therefore proposed an algorithm which incrementally computes the Boltzmann distribution from empirical data using Boltzmann selection.

Definition 2 *Given a distribution p and a selection parameter $\Delta\beta$,* Boltzmann selection *calculates the distribution for selecting points according to*

$$p^s(\mathbf{x}) = \frac{p(\mathbf{x}) e^{\Delta\beta f(\mathbf{x})}}{\sum_y p(\mathbf{y}) e^{\Delta\beta f(\mathbf{y})}} \tag{3}$$

The following theorem has been proven by Mühlenbein and Mahnig (2003).

Theorem 1 *If $p_\beta(\mathbf{x})$ is a Boltzmann distribution, then $p^s(\mathbf{x})$ is a Boltzmann distribution with inverse temperature $\beta(t+1) = \beta(t) + \Delta\beta(t)$.*

The following algorithm is called *BEDA*, the Boltzmann Estimated Distribution Algorithm.

BEDA

- **STEP 0:** $t \Leftarrow 0$. Generate N points according to the uniform distribution ($\beta(0) = 0$).
- **STEP 1:** With a given $\Delta\beta(t) > 0$ do Boltzmann selection giving the distribution $p^s(\mathbf{x}^t)$.
- **STEP 2:** Generate N new points according to the distribution $p(\mathbf{x}^{t+1}) = p^s(\mathbf{x}^t)$.
- **STEP 3:** If termination criteria fulfilled, STOP.
- **STEP 4:** $t \Leftarrow t + 1$. GOTO **STEP 1**.

The following convergence theorem has been proven by Mühlenbein et al. (1999).

Theorem 2 *For $\sum_t \Delta\beta(t) \to \infty$ and infinite populations BEDA converges to a distribution where only the global optima have a probability greater than zero.*

BEDA is only a conceptional algorithm, because the calculation of the distribution $p^s(\mathbf{x}^t)$ requires a sum over exponentially many terms. In order to compute the distribution more efficiently, it has to be factorized. This can be done if the fitness function is additively decomposed.

[1] 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 3 *Let $S_1, \ldots, S_m$, $S_i \subseteq \{1, \ldots, n\}$ be index sets. Let f_i be functions depending only on the variables of S_i. We denote these variables $\mathbf{x}_{s_i}$. S_i is called the scope of f_i. Then*

$$f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x}_{s_i}) \tag{4}$$

is an additive decomposition *of the fitness function (ADF).*

Definition 4 *Given an ADF, the interaction graph G_{ADF}*[2] *is defined as follows: The vertices represent the variables of the ADF . Two vertices are connected by an edge iff the corresponding variables are contained in the same sub-function.*

Remark: The class ADF covers all possible functions. In order to obtain algorithms of polynomial complexity we will later restrict the class. We will assume that the size of the scopes is bounded by a constant independently of n.

2.1 A convergence theorem for factor graphs

In this section we investigate how to efficiently compute and sample the Boltzmann distribution for ADFs. The idea is to factorize the distribution into a product of conditional marginal distributions. The most natural graphical representations of the structure of ADFs are *factor graphs* (Kschischang et al., 2001).

Definition 5 *A factor graph $\mathcal{FG}$ is a graph with two kinds of vertices, the set of factors $\{F_j\}_{j=1}^m$ with scopes $\{S_j\}_{j=1}^m$, and the set of random variables $\mathcal{X} = \{X_1, \ldots, X_n\}$. Each variable is connected to those factors where it is contained in the scope. The Gibbs distribution of $\mathcal{FG}$ is defined as*

$$p_G(\mathbf{x}) = \frac{1}{Z} \prod_{j=1}^{m} F_j(\mathbf{x}_{s_j}) \tag{5}$$

The Gibbs distribution defines a Markov network or Markov random field *on the variables (Pearl, 1988).*

The Boltzmann distribution of an ADF can easily be written as a Gibbs distribution.

$$p_\beta(\mathbf{x}) = \frac{1}{Z_f(\beta)} e^{\beta \sum_{j=1}^m f_j(\mathbf{x}_{s_j})} = \frac{1}{Z_f(\beta)} \prod_{i=1}^{m} e^{\beta f_j(\mathbf{x}_{s_j})} \tag{6}$$

Now set $F_j(\mathbf{x}_{s_j}) = e^{\beta f_j(\mathbf{x}_{s_j})}$ and a Gibbs distribution is obtained which is identical to the Boltzmann distribution. Therefore the BEDA convergence theorem remains valid for factor graphs.

Because the computation of Z is exponential in n, the above factorization is no improvement at first sight. But sampling from a Gibbs distribution can be done using local computations only. The method is called *random Gibbs sampling* (Geman and Geman, 1984; Pearl, 1988).[3] In order to understand Gibbs sampling, the concept *Markov blanket* is needed, the minimal set of variables that separate all variables of a given set D from the other variables in the graph.

Definition 6 (Markov Blanket (Pearl, 1988)) *Let a set of scopes S_j be given. The Markov blanket of a set of variables $D \subseteq \mathcal{X}$ is defined as*

$$MB(D) = \bigcup \{S_j : S_j \in S, S_j \cap D \neq \emptyset\} \setminus D \tag{7}$$

[2] Xiang et al. (1997) call it a decomposable Markov graph.

[3] Pearl (1988) calls it stochastic simulation.

Random Gibbs Sampling

- **STEP 0:** Generate $\mathbf{x}^0 = (x_1^0, \dots, x_n^0)$ randomly. Set $t = 0$.
- **STEP 1:** Choose an index i randomly.
- **STEP 2:** Sample x_i^{t+1} using $p(x_i|MB(X_i))$
- **STEP 3:** Set $\mathbf{x}^{t+1} = (x_1^t, \dots, x_i^{t+1}, \dots x_n^t)$.
- **STEP 4:** t:= t+1; If $t \leq Max$ goto **STEP 1**.

Example 1:

$$f(\mathbf{x}) = f_1(x_1, x_2) + f_2(x_2, x_3) + f_3(x_3, x_4) + f_4(x_4, x_1) \tag{8}$$

The dependencies of G_{ADF} form a loop. We have

$$S_1 = \{X_1, X_2\}, S_2 = \{X_2, X_3\}, S_3 = \{X_3, X_4\}, S_4 = \{X_4, X_1\}$$

Sequential Gibbs sampling would proceed as follows

$$p(x_1^t|x_2^{t-1}, x_4^{t-1}), p(x_2^t|x_1^t, x_3^{t-1}), p(x_3^t|x_2^t, x_4^{t-1}), p(x_4^t|x_3^t, x_1^t)$$

Gibbs sampling uses only local computations. If the size of the Markov blankets is bounded polynomially the computational complexity of one iteration is polynomially bounded. The first samples (until $t = t_0$) are usually thrown away. Instead of updating a single variable, it is possible to update a set of variables. This is called *blocked Gibbs sampling.* Given a set D the Gibbs sampling formula has to be changed in STEP 2:

- **STEP 2:** Sample $\mathbf{x}_D^{t+1}$ using $p(\mathbf{x}_D|MB(D))$

We now prove that Gibbs sampling converges to the true distribution.

Theorem 3 (Convergence) *Let $\mathcal{FG}$ be the factor graph corresponding to the given ADF. Then Gibbs sampling converges to the true Gibbs distribution if all conditionals are greater than zero.*

Proof: The proof is based on two basic theorems in the field of Markov chains and Markov random fields. We recall

Definition 7 *The stochastic process $X^{(i)} = (x_1, \dots, x_n)$ is called a Markov chain if*

$$p(X^{(i)}|X^{(i-1)}, \dots, X^{(0)}) = T(X^{(i)}|X^{(i-1)}) \tag{9}$$

T is called the transition matrix. The transition to $X^{(i)}$ depends only on the states of $X^{(i-1)}$. The interested reader is referred to Andrieux et al. (2003); Geman and Geman (1984); Gilks et al. (1996). Obviously Gibbs sampling defines a Markov chain. The transition matrix is given by the products of the local transitions used for Gibbs sampling. The following theorem is the foundation of Markov chains (Andrieux et al., 2003).

Theorem 4 *A Markov chain converges to a stationary distribution $\pi^*(x)$ if the chain has the two properties*

1. *Irreducibility*
2. *Aperiodicity*

Irreducibility means that there is a positive probability of visiting all states. Gibbs sampling fulfills the assumptions of the above theorem if $p(\mathbf{x}) > 0$ for all $\mathbf{x}$. Furthermore Gibbs sampling is aperiodic.

We now come to the difficult part of the proof. We know that Gibbs sampling converges to a stationary distribution, but not the structure of this distribution. This problem is solved by the next theorem. We state the theorem in the notation of Geman and Geman (1984).

Theorem 5 *Let $\mathcal{G}$ be the neighborhood system defined by the factor graph of the ADF. Perform Gibbs sampling using this neighborhood. Then X is a Markov random field with respect to $\mathcal{G}$ if and only if the stationary distribution $\pi^*(\mathbf{x})$ is the Gibbs distribution with respect to $\mathcal{G}$.*

In our application the Markov random field is defined by the factors and the corresponding Gibbs distribution. Therefore the stationary distribution of Gibbs sampling is this Gibbs distribution. $\diamond$

We can now define the Markov network algorithm (**MN-GIBBS**).

MN-GIBBS

- **STEP 0:** Compute the Markov network from the factor graph of the ADF.
- **STEP 1:** $t \Leftarrow 0$. Generate N points according to the uniform distribution with $\beta(0) = 0$.
- **STEP 2:** With a given $\Delta\beta(t) > 0$ do Boltzmann selection.
- **STEP 3:** Compute the conditional probabilities $p(x_i|MB(X_i) \setminus X_i)$ using the selected points.
- **STEP 4:** Generate a new population N^t according to Gibbs sampling.
- **STEP 5:** If termination criteria are met, STOP.
- **STEP 6:** Add the best point of the previous generation to the generated points (elitist).
- **STEP 6:** Set $t \Leftarrow t + 1$. Goto **STEP 2**.

From theorems 2 and 5 follows the next theorem.

Theorem 6 *For $\sum_t \Delta\beta(t) \to \infty$ and infinite populations the algorithm MN-GIBBS converges to a distribution where only the global optima have a probability greater than zero.*

The computational complexity for one iteration step of Gibbs sampling is bounded by $O(N)$ if the scope of the Markov blanket is bounded by a constant independent of n. But convergence to the true distribution has been proven only for $t \to \infty$. We will give polynomial complexity bounds for certain functions later. For the general case the bounds are exponential. Despite this fact Gibbs sampling is very popular in many scientific disciplines.

Blocked Gibbs sampling converges faster than single variable update, but the convergence might still be very slow. In particular samples with high probability $p(\mathbf{x})$ might be generated only after a large number of steps. This problem is addressed by *importance sampling*. Efficient sampling of Markov networks is still an active research area. There is a trade-off between computational effort and quality of the sample. A recent survey of different variants of Gibbs sampling has been published by Guo and Hsu (2002).

2.2 Implementations of MN-Gibbs

An early implementation using a restricted class of Markov networks is reported in Santana (2003). The algorithm called *MN-EDA^f* in Santana (2005) is an implementation of MN-GIBBS. The author is mainly interested in learning the Markov network from data, so *MN-EDA^f* is not thoroughly investigated. It is used for comparisons only. In the paper also different variants of Gibbs sampling are numerically investigated. An interesting variant is to start sampling not randomly, but at points with high fitness values.

The algorithm IS-DEUM by Shakya (2006) is a substantial modified Markov network algorithm. The author does not use the ADF for constructing the Markov network, but computes a restricted model U of the fitness function from the factor equation $-\ln f(\mathbf{x}) = U(\mathbf{x})$ using N samples. The author considers linear and quadratic U. The coefficients can be obtained by solving a linear equation. Depending on the relationship between N and the number of coefficients M of the model, the system will be under-, over-, or precisely-specified. For the solution of the equation Singular Value Decomposition is used.

Let M be the number of coefficients of the model. Then the computational complexity is $O(M^2N)$ if $N < M$ and $O(MN^2)$ if $N > M$. Thus the computation of a quadratic model is already very expensive. The model is computed only once. The global optima of the function are computed by Gibbs sampling, where the temperature $T = 1/\beta$ is continously decreased till it is zero. The algorithm spends most of the computation time within this modified Gibbs sampling. This is a strong indication that the algorithm is more a simulated annealing algorithm (Kirkpatrick et al., 1983) than an EDA.

A full MN-GIBBS implementation seems to be the algorithm $MARLEDA^{+model}$ from Alden (2007). The full Markov network model is derived from the ADF. The conditional probabilities are computed from the selected set of points. Then Gibbs sampling is used to create a new population. The author does not investigate this algorithm, because he concentrates on the difficult task of learning the Markov model from data. $MARLEDA^{+model}$ is used for numerical comparisons only.

2.3 Regional graphs and energy minimization

Santana (2005) and Mühlenbein and Höns (2005) introduced *regional graphs* for EDAs. Regional graphs are closely related to Markov random fields and the Gibbs distribution. A definition of regional graphs is outside the scope of this paper. Santana used the Kikuchi factorization and computed its parameters in each generation from the selected samples of high fitness. The new population is generated by Gibbs sampling.

Mühlenbein and Höns (2006) used the original idea of Kikuchi and computed the parameters of the Kikuchi factorization by minimizing Gibbs energy using the subfunctions of the ADF. This is done only once for a chosen β of the Gibbs distribution. Sampling is done with *Probabilistic Logic Sampling* (PLS) using a simplified factorization. PLS is explained in the next section.

3 A convergence theorem for the factorized distribution algorithm

The most popular graphical model used in EDAs is the Bayesian network (BN) (Pearl, 1988). A BN represents a factorized distribution in the following form

$$\tilde{p}(\mathbf{x}) = \prod_{i=1}^{n} p(x_i|\mathbf{pa}_i), \quad \mathbf{pa}_i \subseteq \{X1, X_2, \ldots, X_{i-1}\} \tag{10}$$

where $\mathbf{pa}_i$ are called the parents of x_i, $X_0 = \emptyset$. [4]
Note that any distribution can be written in the form of a Bayesian network because of

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)\cdots p(x_n|x_1,\ldots,x_{n-1}) \tag{11}$$

But this factorization uses conditional distributions of size $O(n)$, thus sampling from the distribution is exponential in n. Therefore we are looking for factorizations where the size of the marginals is bounded independently of n.
In order to compute a BN of an ADF, we can use the following simple algorithm.

Definition 8 *Given $S_1,\ldots,S_m$, we define the sets D_i, B_i and C_i for $i = 1,\ldots,m$:*

$$D_i := \bigcup_{j=1}^{i} S_j, \qquad B_i := S_i \setminus D_{i-1}, \qquad C_i := S_i \cap D_{i-1} \tag{12}$$

We require that $(B_i \neq \emptyset,\ p(\mathbf{x}_{B_i}|\mathbf{x}_{C_i}) > 0\ : i = 1,\ldots,m)$, $D_m = \{1,\ldots,n\}$ and set $D_0 = \emptyset$. In the theory of decomposable graphs, D_i are called histories, *B_i* residuals *and C_i* separators *(Lauritzen, 1996). A FDA factorization of the ADF is defined by*

$$\tilde{p}(\mathbf{x}) = \prod_{i=1}^{m} p(\mathbf{x}_{B_i}|\mathbf{x}_{C_i}) \tag{13}$$

The set $\{B_i, C_i\}$ is a *clique*[5]. A necessary condition that the FDA factorization describes a distribution is obviously that the given marginals and conditional distributions are consistent.

Definition 9 *A set of marginal distributions $p(\mathbf{x}_{B_i}, \mathbf{x}_{C_i})$ is called* consistent *if the marginal distributions fulfill the laws of probability, e.g.*

$$\sum_{\mathbf{x}_{B_i},\mathbf{x}_{C_i}} p(\mathbf{x}_{B_i}, \mathbf{x}_{C_i}) = 1 \tag{14}$$

$$\sum_{\mathbf{x}_{B_i}} p(\mathbf{x}_{B_i}, \mathbf{x}_{C_i}) = p(\mathbf{x}_{C_i}) \tag{15}$$

Any FDA factorization can easily be transformed into a BN. We just give a simple example

$$\begin{aligned}\tilde{p}(x) &= p(x_1,x_2)p(x_3,x_4,x_5|x_1,x_2)\\ &= p(x_1)p(x_2|x_1)p(x_3|x_1,x_2)p(x_4|x_1,x_,x_3)p(x_5|x_1,x_2,x_3,x_4)\end{aligned}$$

A FDA factorization can easily be used for sampling. The simplest sampling method is called *probabilistic logic sampling* (PLS) introduced by Henrion (1988). It works as follows:

Probabilistic Logic Sampling

- **STEP 1:** For $t = 1$ to N; For $i = 1$ to m
- **STEP 2:** Sample $\mathbf{x}^t_{B_i}$ from $p(\mathbf{x}_{B_i}|\mathbf{x}^t_{C_i})$

[4] In machine learning a Bayesian network is a directed graph because the dependencies are causal interpreted. For EDA applications this restriction is not necessary, because only the factorization is needed. The factorization defines an ordering of the network.

[5] A clique (Pearl, 1988) is a set of vertices V such that for every two vertices in V, there exists an edge connecting it. This is equivalent to saying that the subgraph induced by V is a complete graph.

In general, PLS does not generate the true distribution. This has been proven by Höns (2006). The proof is lengthy and very technical. It is therefore omitted.

Proposition 1 *Let a consistent set of marginal distributions $p(\mathbf{x}_{B_i}, \mathbf{x}_{C_i})$ be given. Then using PLS we have*

$$\tilde{p}(\mathbf{x}_{B_i}|\mathbf{x}_{C_i}) = p(\mathbf{x}_{B_i}|\mathbf{x}_{C_i}), \;\; i = 1, \ldots m \tag{16}$$

whereas in general

$$\tilde{p}(\mathbf{x}_{B_i}, \mathbf{x}_{C_i}) \neq p(\mathbf{x}_{B_i}, \mathbf{x}_{C_i}), \;\; i = 1, \ldots m \tag{17}$$

The inequality (17) is often overlooked. It means that probabilistic logic sampling does not reproduce the given marginals for general FDA factorizations, despite the conditionals are reproduced. Thus *PLS might not generate the true distribution.* The class of FDA factorizations has to be constrained further. This problem is investigated next.

3.1 Exact FDA Factorizations

The following theorem was proven by Mühlenbein et al. (1999) for the Boltzmann distribution.

Theorem 7 (Factorization Theorem) *Let $f(\mathbf{x}) = \sum_{i=1}^{m} f_{s_i}(\mathbf{x})$ be an ADF. Compute a FDA factorization. If*

$$\forall i \geq 2 \; \exists j < i \;\; \textit{such that} \; C_i \subseteq S_j \tag{18}$$

then

$$p_\beta(\mathbf{x}) = \prod_{i=1}^{m} p_\beta(\mathbf{x}_{B_i}|\mathbf{x}_{C_i}) = \frac{\prod_{i=1}^{m} p_\beta(\mathbf{x}_{B_i}, \mathbf{x}_{C_i})}{\prod_{i=2}^{m} p_\beta(\mathbf{x}_{C_i})} \tag{19}$$

Definition 10 *The constraint defined by equation* (18) *is called the* running intersection property (*RIP*) *(Lauritzen, 1996). The factorization is* polynomially bounded *(PBF) if the size of the cliques $\{B_i, C_i\}$ is bounded by a constant independent of n.*

The theorem is not restricted to the Boltzmann distribution, but is valid for all applications involving the computation of a sum-product (Kschischang et al., 2001). Note that exact factorizations are not unique. In fact we have

Corollary 1 *Any FDA factorization which fulfills the RIP and contains the interaction graph G_{ADF} as a subgraph generates the true distribution using PLS.*

The corollary follows from the observation that we can join two or more sub-functions, resulting in an ADF with larger sets $\tilde{S}_i$. From a numerical point of view, the best factorizations would fulfill the conditions of the theorem and have the smallest cliques.

Let us discuss a simple example, the function defined in (8).

$$\begin{aligned} f(\mathbf{x}) &= f_1(x_1, x_2) + f_2(x_2, x_3) + f_3(x_3, x_4) + f_4(x_4, x_1) \\ \tilde{p}_1(\mathbf{x}) &= p(x_1, x_2)p(x_3|x_2)p(x_4|x_3) \end{aligned}$$

The factorization leaves out the dependency between X_4 and X_1. This problem can be solved by joining sub-functions $f_3(x_3, x_4)$ and $f_4(x_4, x_1)$. This leads to the factorization

$$\tilde{p}_2(\mathbf{x}) = p(x_1, x_2)p(x_3|x_2)p(x_4|x_3, x_1)$$

This factorization contains all edges of G_{ADF} but violates the RIP because X_1 and X_3 are not contained in a common clique. Sampling with PLS might not reproduce the distribution p. Therefore we join sub-functions f_1 and f_2. This gives a factorization fulfilling the RIP

$$\tilde{p}_3(\mathbf{x}) = p(x_1, x_2)p(x_3|x_2, x_1)p(x_4|x_3, x_1)$$

Sampling $\tilde{p}_3(\mathbf{x})$ using PLS will generate the distribution p.

In order to obtain a factorization fulfilling the RIP all combinations of joining sub-functions have to be tested. This is prohibitive for an arbitrary ADF. Actually, it turns out that the computation of an exact factorization is done better by investigating the corresponding interaction graph G_{ADF}. A well-known algorithm computes *junction trees* (Jensen and Jensen, 1994). It obtains an exact factorization of reasonable clique sizes and fulfilling the *RIP*, if possible. A short description of the algorithm can be found in Mühlenbein and Höns (2005). The largest clique of the junction tree gives the largest marginal of the factorization and determines the numerical complexity. Computing the network with minimal largest clique size is NP-hard (Cooper, 1990).

The space complexity of exact FDA factorizations has been investigated by Gao and Culberson (2005). For many interesting problems like functions defined on grids of dimension two and higher exact factorizations have clique sizes of $O(n)$, thus they are not bounded polynomially. For these functions one has to use approximate factorizations. A good heuristic should minimize the size of the cliques but simultaneously use all dependencies in G_{ADF}.

3.2 The Factorized Distribution Algorithm FDA

The FDA factorization is the heart of the factorized distribution algorithm.

FDA

- **STEP 0:** Set $t \Leftarrow 0$. Generate N points randomly.
- **STEP 1:** Selection of points with high fitness.
- **STEP 2:** Compute the conditional probabilities $p^s(\mathbf{x}^t_{B_i}|\mathbf{x}^t_{C_i})$ using the selected points.
- **STEP 3:** Generate a new population according to $p(\mathbf{x}^{t+1}) = \prod_{i=1}^{m} p^s(\mathbf{x}^t_{B_i}|\mathbf{x}^t_{C_i})$
- **STEP 4:** If termination criteria are met, STOP.
- **STEP 5:** Add the best point(s) of the previous generation to the generated points (elitist).
- **STEP 6:** Set $t \Leftarrow t + 1$. Go to **STEP 1**.

In FDA we have not implemented a junction tree algorithm, but a very fast algorithm called the *sub-function merger algorithm*. It works as follows. Each new variable is included in a set together with the previous variables on which it depends. However, if another variable depends on a superset of variables, the two sets are merged. After completing the merge phase, the algorithm calculates $\tilde{C}_j$, $\tilde{B}_j$ and $\tilde{D}_j$ analogous to the construction given by (12). This sub-function merger algorithm might compute too large cliques. Therefore a cut parameter k is used which bounds the clique size. If the clique size becomes larger than k our implementation will randomly leave out arcs from G_{ADF}. The interested reader is referred to Mühlenbein and Höns (2005). Thus the FDA algorithm tries to cover all interactions of G_{ADF}, but does not care about the RIP.

The next theorem follows from the Factorization Theorem and the convergence theorem of BEDA.

Theorem 8 *Run FDA with Boltzmann selection. If the FDA factorization covers all dependencies of the network G_{ADF} and fulfills the RIP, then FDA with PLS will converge to the optima.*

We will later prove that the theorem remains true if a polynomially bounded sample size N is used. In Mahnig and Mühlenbein (2001) an adaptive annealing schedule SDS for Boltzmann selection has been derived and analyzed theoretically. Convergence to the global optima for FDA with other selection methods has been proven by Zhang (2004); Zhang and Mühlenbein (2004).

Note that the theorem gives *sufficient conditions* for convergence to the optima. The conditions are not necessary for convergence. We only need to use a factorization where the *probabilities of the global optima* are high and sampling generates points of high probability frequently. We explain the problem with a simple example.

Example 2:

$$f(\mathbf{x}) = \sum_{i=1}^{n} x_i + \prod_{i=1}^{n} x_i \tag{20}$$

The exact factorization is the joint distribution $p(x_1, \dots, x_n)$. The FDA factorization for $f(\mathbf{x}) = \sum_{i=1}^{n} x_i$ is

$$p(\mathbf{x}) = \prod_{i=1}^{n} p(x_i) \tag{21}$$

FDA will easily find the optimum using this factorization, because $\sum x_i$ and $\prod x_i$ have the same optimum. The factorization is also a good approximation of the true distribution, because $\prod x_i = 0$ for $x \neq (1, \dots, 1)$. Thus instead of using the exact distribution, it is possible to use a *simpler distribution which has the same global optima as the original distribution*.
Now we change the function.

$$f(\mathbf{x}) = \sum_{i=1}^{n} (1 - x_i) + (n+1) \cdot \prod_{i=1}^{n} x_i \tag{22}$$

The exact factorization is again the joint distribution. But now the optimum of the sum is $\mathbf{x} = (0, \dots, 0)$, the optimum of the product is $\mathbf{x} = (1, \dots, 1)$ which is the global optimum with a function value of $n + 1$. The factorization (21) is again a good approximation of the exact distribution because the product contributes to the fitness function only at $\mathbf{x} = (1, \dots, 1)$. But using this factorization, any kind of selection will drive $p(x_i)$ to 0. Thus FDA will converge to the second best maximum.The next function is

$$f(\mathbf{x}) = \sum_{i=1,4,\dots}^{3m-2} x_i * x_{i+1} * x_{i+2} \tag{23}$$

The exact factorization is

$$p(\mathbf{x}) = \prod_{i=1,4,\dots}^{3m-2} p(x_i, x_{i+1}, x_{i+2}) \tag{24}$$

Here the simple factorization (21) is a bad approximation of the exact factorization, but EDA algorithms using this factorization will easily find the optimum. Selection will increase the number of strings with a large number of 1's. Therefore $p(x_i)$ will converge to 1, ultimately generating the global optimum.

FDA has experimentally proven to be very successful on a number of functions where standard genetic algorithms fail to find the global optimum. For recent surveys and a more detailed description of the algorithm, the reader is referred to Mühlenbein and Mahnig (2002a, 2003); Mühlenbein and Höns (2005, 2006).
Next we shortly describe algorithms which learn the structure of the network from data.

4 Learning a Factorization from Data

If the explicit mathematical definition of the function is not known, its structure has to be learned from the data. For detecting dependencies different measures can be used, like Pearson's χ^2 test, the conditional entropy or the maximum likelihood (Mühlenbein and Höns, 2006).

4.1 Learning a Bayesian network

We consider the class of Bayesian networks defined in equation (10). The *conditional entropy* between two variables is defined as

$$H(X,Y) = -\sum_{x,y} p(x,y)\log p(x|y) \tag{25}$$

Let B denote a Bayesian network, D the given data set and $N = |D|$ its size. Then the conditional entropy $H(B,D)$ of the network structure B and data D is given by

$$H(B,D) = -\sum_{i=1}^{n}\sum_{\mathbf{pa}_i}\sum_{x_i}\frac{m(x_i,\mathbf{pa}_i)}{N}\log\frac{m(x_i,\mathbf{pa}_i)}{m(\mathbf{pa}_i)} \tag{26}$$

Here $m(x_i,\mathbf{pa}_i)$ denotes the number of occurrences of x_i given configuration $\mathbf{pa}_i$. $m(\mathbf{pa}_i) = \sum_{x_i} m(x_i,\mathbf{pa}_i)$. If $\mathbf{pa}_i = \emptyset$, then $m(x_i,\emptyset)$ is set to the number of occurrences of x_i in D.

In order to find a good network structure with small cliques, just minimizing the conditional entropy is not enough. Networks with a larger set of parameters obviously fit the distribution better. In order to give more weight to sparse Bayesian networks, a weight factor α is used which *penalizes the size of the network*. To score the networks we use the *Bayesian Information Criterion* proposed by Schwarz (1978)

$$BIC(\alpha) = -N\cdot H(B,D) - \alpha PA\cdot\log(N) \tag{27}$$

PA is the total number of probabilities to compute. The term $PA\cdot\log(N)$ models the computational cost of computing the probabilities. Note that $-H(B,D)$ is multiplied by the number of samples. The larger the sample size, the more weight is given to fitting the empirical distribution. Under certain assumptions, Schwarz computed $\alpha = 0.5$ as the optimal weight. A more detailed derivation of BIC using the maximum entropy principle and the log-likelihood principle can be found in (Mühlenbein and Höns, 2006).

To compute a network B^* which maximizes BIC requires a search in the space of all BNs with bounded number of parents. It has been proven that the computation of the best BN is *NP-hard* (Cooper, 1990; Chickering et al., 2004).

We use the following greedy algorithm instead. This simple learning method has been first proposed by Heckerman et al. (1995). It starts with an arc-less network. At each step it adds the edge which gives the maximum increase of BIC(α). k_{max} is the maximum number of incoming edges allowed.

$$\mathbf{BN}(\alpha, \mathbf{k_{max}})$$

- **STEP 0:** Start with an arc-less network.
- **STEP 1:** Add the arc (x_i, x_j) which gives the maximum increase of BIC(α) if $|pa_j| \le k_{max}$ and adding the arc does not introduce a cycle.
- **STEP 2:** Stop if no arc is found.

Checking whether an arc would introduce a cycle can easily be done by maintaining for each node a list of parents and ancestors, i.e. parents of parents etc. $(x_i \to x_j)$ introduces a cycle if x_j is ancestor of x_i. Because of the additivity of BIC only the term has to be recomputed where the edge is added. Thus the computational complexity of the learning algorithm is bounded by $O(Nn^2)$.

Our algorithm LFDA uses the above learning method. LFDA is very similar to FDA. The only difference is that in STEP 2 of FDA the Bayesian network is computed anew from the set of selected points. As selection converges to a small set of points with good fitness, the learned networks get more and more sparse. If selection converges to a single point, the network will be arc-less.

This learning method is also used by the Bayesian Optimization Algorithm BOA from Pelikan and Goldberg (2000, 2002). A different variant of Bayesian learning is implemented in the Estimation of Bayesian Networks algorithm (EBNA) (Etxeberria and Larrañaga, 1999). A general overview of learning graphical models can be found in the book edited by Jordan (1999). A recent numerical evaluation of the efficiency of popular learning algorithms has been done by Tsamardinos et al. (2006).

4.2 Learning of Markov networks and factor graphs

Learning of factor graphs is easier than learning of Bayesian networks because they match the ADF structure. Abbeel et al. (2006) present a polynomial learning algorithm. Like LFDA it uses the conditional entropy to find the best network in a restricted class of bounded Markov networks.

Theorem 9 (Computational complexity of learning Abbeel et al. (2006) Theorem 14) *Let $\gamma > 0$ be the minimum of the conditionals $p(X_i|\mathcal{X} \setminus X_i)$ be independent of n. Let k be the maximum number of variables per factor; let b be the maximum number of variables per Markov blanket. Then the running time rt of the learning algorithm is*

$$rt \in O\left(Nkb(k+b)n^{k+b}\right) \tag{28}$$

The learning algorithm tests all combinations of sets of variables and corresponding Markov blankets. Therefore we have an exponential dependence on the maximum scope size k and the maximum Markov blanket size b, the dominating term is n^{k+b}. For a large number of variables and $k \ge 3$ this learning algorithm is computational too expensive. The polynomial bound can be made smaller by implementing a more sophisticated learning algorithm. To my knowledge the learning algorithm has not yet been implemented.

A much simpler learning algorithm is used in MARLEDA (Alden, 2007). It uses Pearson's χ^2 test to compute the confidence level of the dependencies between variables. Using the dependencies a Markov network is constructed. The construction procedure is very simple, so it does not guarantee that the correct factor graph is obtained.

MN-EDA of Santana (2005) also uses the χ^2 test to detect dependent variables. It creates not a full Markov network, but Kikuchi approximations. The main computational cost of learning arises in performing the independence tests. It is upper bounded by $O(Nn^3)$.

5 Convergence of EDAs with Finite Samples

The convergence theorems presented so far are valid for infinite populations only. We now investigate convergence for finite populations. EDAs are probabilistic algorithms. For finite populations convergence to the global optima can only be probabilistic. In this section we will use the (ϵ, δ) convergence concept first applied in *Probably Approximately Correct* (PAC) learning (Kearns and Vazirani, 1994). It is defined as follows:

Definition 11 *Let $\epsilon > 0,\ \delta > 0$. Let p be the true distribution, $\tilde{p}$ an approximation. Then we speak of (ϵ, δ) convergence if*

$$prob\left(error(p, \tilde{p}) \leq \epsilon\right) \geq 1 - \delta \tag{29}$$

$error$ denotes any distance measure.

To get a feeling for sample complexity bounds, consider the following problem: suppose we have a coin whose heads probability p we wish to determine. Letting 0 denote tails and 1 denote heads we obtain a sample $D_N = \{x^1, \ldots, x^N\}$. The maximum likelihood approximation gives $\hat{p} = 1/N \sum_i \mathbf{x}^i$. The question then arises: how large must N be for $|\hat{p} - p| < \epsilon$ with probability at least $1 - \delta$. An application of Chernoff-type bounds (Kearns and Vazirani, 1994) shows that $N = \frac{1}{2\epsilon^2} ln \frac{2}{\delta}$ will suffice. This *upper bound* is distribution free and makes no assumption about the value of p.

In this case a lower bound can also be obtained. We set $p = 0.5$. By bounding the relevant binomial coefficients, we obtain inequalities for the probability that the estimate is more than ϵ away from p. One obtains $N \geq \frac{1}{5\epsilon^2} ln \frac{2}{\delta}$. Thus in this simple example the upper and lower bound differ only by a small multiplicative constant.

5.1 Sample complexity for FDA

Let a Bayesian network be given. Let $D_N = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ be the empirical sample where $\mathbf{x}^i = (x_1^i, \ldots, x_n^i)$. It is easy to show that the maximum likelihood approximation of the true probabilities are the long-run frequencies of the sample. The error between two distributions is often measured by the Kullback-Leibler divergence.

Definition 12 *The Kullback-Leibler (KLD) divergence between two distributions is defined by*

$$D(p||q) = \sum_{\mathbf{x}} p(\mathbf{x}) \ln \frac{p(\mathbf{x})}{q(\mathbf{x})} \tag{30}$$

Note that the divergence is not symmetric!

There exist a number of papers deriving bounds on the sample size for PAC convergence. These bounds are valid for the approximation of a single distribution (Dasgupta, 1997). But in EDAs a new distribution is computed at each generation. This complicates the application of PAC learning considerably. The next theorem gives our main result. Let $|p - \hat{p}|_1 = \sum_x |p(\mathbf{x}) - \hat{p}(\mathbf{x})|$.

Theorem 10 *Let f be an ADF function with m sub-functions of at most k binary variables. Let the FDA factorization*

$$p(\mathbf{x}) = \prod_{j=1}^{m} p(\mathbf{x}_{b_j}|\mathbf{x}_{c_j})$$

fulfill the RIP. Run FDA with Boltzmann selection and PLS. Let $p_g(\mathbf{x})$ be the true Boltzmann distribution at generation g with Boltzmann factor $\beta_g = \sum_{i=1}^{g} \Delta\beta_i$, $\hat{p}_g(\mathbf{x})$ be the empirical distribution. Let $N \ll 2^n$. Let Boltzmann selection select at least $N/4$ different points in each generation. Let any $0 < \epsilon < 1$, $0 < \delta < 1$ be given. Then in order that

$$|p_g - \hat{p}_g|_1^2 \leq g\epsilon \tag{31}$$

to hold with probability at least $1 - \delta$ it suffices that the sample size N fulfills

$$N \geq \frac{8 \ln 2}{\epsilon} m \left(2^k \ln 2 + \ln \frac{mg}{\delta}\right) \tag{32}$$

The proof can be found in the appendix.

The assumption that Boltzmann selection selects at least $N/4$ different points is fulfilled almost surely as long as $p_g(\mathbf{x}_{opt}) \leq 1/N$ [6] and selection is not too strong. FDA checks if this condition is fulfilled. If not, it increases the sample size or stops the algorithm after two generations.

Next we eliminate the dependency on the number of generations g. Thierens and Goldberg (1994) have proven that for truncation selection the number of generations until the population is fixed is bounded by n. Mahnig and Mühlenbein (2001) have shown that the Boltzmann selection scheme *SDS* is asymptotical equal to truncation selection. Therefore the bound can be used for *SDS* also. To be on the save side we set $g = 2n$. Setting $\epsilon' = \epsilon/(2n)$ in theorem 10 we obtain the corollary.

Corollary 2 *Let the assumptions of theorem 10 be fulfilled. Let any $0 < \epsilon < 1$, $0 < \delta < 1$ be given. Let $g = 2n$. Then for*

$$|p_g(\mathbf{x}_{opt}) - \hat{p}_g(\mathbf{x}_{opt})|^2 \leq \epsilon \tag{33}$$

to hold with probability $1 - \delta$ it suffices that N fulfills

$$N \geq \frac{16 \ln 2}{\epsilon} nm(2^k \ln 2 + \ln \frac{2nm}{\delta}) \tag{34}$$

For convergence to the optima $p_g(\mathbf{x}_{opt})$ has to be larger than $1/N$. This can always be achieved because $p_g(\mathbf{x})$ is a Boltzmann distribution. We shortly discuss this problem. We will require that $p_\beta(\mathbf{x}_{opt}) = e^{-1}$.

Example 3: $f1(\mathbf{x}) = \sum_{i=1}^{n} x_i$
We easily compute

$$p_\beta(\mathbf{x}_{opt}) = \frac{e^{n\beta}}{(1 + e^\beta)^n} = \frac{1}{(1 + e^{-\beta})^n} \tag{35}$$

We set $\beta = \ln n$ and obtain

$$p_\beta(\mathbf{x}_{opt}) \to e^{-1} \qquad n \to \infty$$

[6] If $p_g(\mathbf{x}_{opt}) > 1/N$ then the optimum will be found with high probability in this or the next generation.

We now change the scaling of the function: $f2(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} x_i$
Here the difference between the optimum and the second best value is $1/n$. One computes

$$p_\beta(\mathbf{x}_{opt}) = \frac{1}{(1+e^{-\beta/n})^n} \tag{36}$$

If we set $\beta = n \ln n$ we obtain as before $p_\beta(\mathbf{x}_{opt}) \to e^{-1}$.

Table 1 shows the numerical results.

$f1$	n	β_{theo}	$\hat{p}(x(opt)$	β_{FDA}	g
$f1$	50	3.9	0.32	3.7	12
$f2$	50	195	0.29	190	12
$f1$	100	4.6	0.28	4.6	19
$f2$	100	460	0.35	430	18
$f1$	200	5.3	0.27	5.1	28
$f2$	200	1059	0.31	1010	28
$f1$	400	6.0	0.35	5.9	42
$f2$	400	2396	0.32	2360	43

Table 1: Comparison of theoretical β and β obtained by FDA. (t) denotes truncation selection

For both functions the Boltzmann selection schedule *SDS* reaches the theoretical β in almost the same number of generations, despite the large difference of the values. Boltzmann selection is invariant to a scaling of the function. In both cases the number of generations to reach the required value of β scales like $O(\sqrt{n})$. This has been proven for this class of functions by Mühlenbein and Schlierkamp-Voosen (1993).

Summary: We have proven convergence to the optima in polynomial time for FDA using PLS if the network fulfills the running intersection property. Numerical experiments indicate that he sample size bound of $N \in O(nm \ln nm)$ is too large. It arose because we had to estimate g successive generations of distributions. But the strongest assumption is the RIP. This assumption is necessary because otherwise the optima of the empirical distribution might not be the optima of the Boltzmann distribution.

5.2 Computational complexity of EDAs using Markov networks

A recent complexity analysis for Markov networks has been done by Abbeel et al. (2006). They assume that an unknown process has produced the samples according to the unknown Markov random field. Then the sample complexity is polynomially bounded if the scope of the Markov blankets are bounded independently of n.

But EDAs have to sample the Markov random field. Here often Gibbs sampling is used. Gibbs sampling is an iterative process, the number of steps needed for convergence is unknown in general. For special functions in statistical physics and image restauration the computational complexity of Gibbs sampling has been intensively studied. The following results have been reported by Gibbs (2000).

Let a Markov chain with probability transition matrix P and stationary distribution pi be given. Let $\mathbf{x}^0$ be the initial configuration.

Definition 13 *The total variation distance at step t is*

$$D_{x^0}(t) = \frac{1}{2} \sum_x |P^t(\mathbf{x}^0, \mathbf{x}) - \pi(\mathbf{x})| \tag{37}$$

The convergence time of the Markov chain used by the Gibbs sampler is defined as

$$\tau(\epsilon) = \max_{x^0} \min\{t : D_{x^0}(t) \le \epsilon \ \ t' \ge t\} \tag{38}$$

$P^t(\mathbf{x}^0, \mathbf{x})$ denotes the probability that the Markov chain with initial state $\mathbf{x}^0$ is in state $\mathbf{x}$ at iteration t.
For the ferromagnetic Ising model

$$\begin{aligned} H(s) &= -\sum_{i,j} s_i s_j - h \sum_i s_i \\ p_\beta(s) &= e^{\beta H(s)} \end{aligned}$$

Gibbs (2000) summarizes the results[7]. The convergence rate for one-dimensional problems is $(O(n \ln(n))$ where n denotes the number of points. In dimensions higher than one, this result holds for $h = 0$ for all values below a critical value at which a phase transition occurs[8]. For the Ising model with an external field $(h > 0)$ the convergence rate can be shown to be $O(n \ln(n))$ for all β in two dimensions and for small β in higher dimensions.

The disappointing result for $h = 0$ can easily be explained. This problem has two optima, $\mathbf{s}_+ = (+1, +1, \ldots, +1)$ and $\mathbf{s}_- = (-1, -1, \ldots, -1)$. If β gets large, Gibbs sampling needs a long time to traverse between $\mathbf{s}_+$ and s_-. But in general the complexity results are encouraging. Frigessi et al. (1997) conjecture that Gibbs sampling from a Markov random field that does not undergo a phase transition has polynomial complexity.

5.3 Sample complexity for learning the ADF structure by probing

A learning method not based on statistical independence tests has been investigated by Heckendorn and Wright (2004). Earlier work has been reported by Munetomo and Goldberg (1999). The method computes the structure of the function by computing its Walsh coefficients.

Theorem 11 (Heckendorn and Wright (2004)) *Assume a class of ADF functions where each sub-function has at most k variables. Let $\delta > 0$ be a constant. Then the number of function evaluations required by the DETECT-LINKAGE learning algorithm of order 2 to detect the scope of all sub-functions with probability at least $1 - \delta$ is bounded by*

$$N \in O\left(2^k n^2 \ln n \ln(1 - \delta^{\frac{1}{2}})\right) \tag{39}$$

Wright and Pulavarty (2005) propose to use FDA to compute a factorization of the ADF. If the ADF structure is too complex the algorithm MN-GIBBS should be used instead. For *separable* ADFs a smaller bound can be shown Streeter (2003). For separable functions an upper bound of $N \in O(m \ln m)$ where $n = m * k$ has been reported for the extended compact genetic algorithm $ECGA$ by Harik et al. (2006). But their error measure is weaker. It is assumed that the probability of a sub-function being not correct is $\delta = 1/m$.

[7] In physics one minimizes H.

[8] For two dimensional problems we have $\beta_c \approx 0.44$.

6 The Connection between FDA and LFDA

Learning can have an advantage compared to a fixed FDA factorization. It can use the actual data to find a good model. But what is a good model for EDAs ? We have shown in the previous sections that for provable convergence a model should match the structure of G_{ADF}. Can learning methods detect this structure in principle? The answer is yes. The reason is the following conjecture (Mühlenbein and Höns, 2005). In order to state the conjecture Shannon's *mutual information* $I(X,Y)$ is needed (Pearl, 1988). It is a nonnegative quantity and is equal to 0 iff X and Y are mutually independent. It is closely related to the conditional entropy.

$$I(X;Y) = H(X|Y) - H(X) \tag{40}$$

Conjecture: *Let the empirical distribution $\hat{p}(\mathbf{x})$ be generated by selection from an ADF . Then for large sample sizes N created by selection the mutual information is the largest between those variables which are contained in a common sub-function. Therefore it is possible to detect the interaction graph G_{ADF} from data.*

We have numerically investigated the conjecture and found no violation so far. Thus detecting the dependencies is not the problem of learning, but to compute a good Bayesian network given this information. Computing the best BN is NP-hard. Therefore most learning algorithms use a simple greedy search. Thus the quality of the learned BNs depends mainly on the learning algorithm. We will test the LFDA learning algorithm numerically using four test functions of increasing complexity.

F1: The trap function $f_{Trap(k,m)}$
Let $k \geq 3$, let m be the number of sub-functions.

$$f_{Trap(k,m)} = \sum_{j=0,k,\ldots,k\cdot(m-1)} f_{trap(k)}(x_j, x_{j+1}, \ldots, x_{j+k-1}) \tag{41}$$

$f_{Trap(k,m)}$ is a separable function and therefore easy to optimize. $f_{trap(k)}$ is a deceptive function (Deb and Goldberg, 1993). $f_{trap(3)}$ is shown in table 2.

F2: The function $f_{Iso(n)}$
Let $n \geq 4$ be even, let $m = n/2$ be the number of sub-functions.

$$f_{Iso(n)} = \sum_{j=0,2,\ldots,n-4} f_1(x_j, x_{j+1}, x_{j+2}) + f_2(x_{n-2}, x_{n-1}, x_n) \tag{42}$$

The functions f_1 and f_2 are defined in table 2. f_1 has the maximum at $x = (0,0,0)$, whereas f_2 has the maximum at $x = (1,1,1)$. The global optimum is $\mathbf{x} = (1,1,\ldots,1)$ with a function value of $m^2 - m + 1$. It is very isolated. The second optimum has a value of $m^2 - m$ and is at $\mathbf{x} = (0,0,\ldots,0)$. Its attractor region is much larger, therefore the function is very difficult to optimize.

F3: The 2D grid Ising spin glass

$$f_{Ising} = -\sum_{i,j} J_{i,j} s_i \cdot s_j - \sum_i h_i s_i \tag{43}$$

j are the four neighbors of i in the 2D grid. The couplings $J_{i,j}$ are randomly drawn from a Gaussian distribution. h is the external magnetic field. The spins s_i have values

u	0	1	2	3
$f_{trap(3)}$	2	1	0	3
f_1	m	0	0	m-1
f_2	0	0	0	m

Table 2: Definition of the sub-functions of $f_{trap(3)}$ and $f_{Iso(n)}$; m denotes the number of sub-functions, and u the number of bits on.

in $\{-1, 1\}$. We will use $h_i = 0, \ \forall i$. This function is symmetric, therefore it has two global optima, which are the binary complement from each other.

F4: Kauffman's $n : k$ function.
Here we have n sub-functions, each with k variables. For each variable $k-1$ variables are randomly chosen, defining the scope S_i of sub-function f_i. The function values of f_i are uniformly distributed in $[0, 1]$

$$K_{n;k} = \sum_{i=1}^{n} f_i(\mathbf{x}_{S_i}) \tag{44}$$

The interconnection structures of the four functions are representative for a large class of functions. f_{Trap} is a separable function. f_{Iso} is defined on the real axis with an overlap of one variable, the Ising model is defined on a 2D grid, and Kauffman's function has a random interconnection structure.

The optimization of $K_{n;3}$ is NP-hard (Wright et al., 2000). The computational complexity of the Ising spin glass model depends on the structure and is discussed in the appendix.

6.1 Investigation of the learned Bayesian networks

The *SDS* selection method does not work efficiently with LFDA. The selection is too weak the first generations, important dependencies are not detected. Therefore we use LFDA with truncation selection with standard parameter $\tau = 0.3$. BOA uses tournament selection with 50% replacement. Both algorithms compute the network anew in each generation. Table 3 shows numerical results for problem sizes of $n = 49$ or $n = 50$. We want to show a tendency, therefore the table just summarizes a single network, usually taken after the first four generations. The learning algorithms does not penalize RIP violations, so there are lots of RIP violations.

Table 3 shows the results for two penalty factors α, namely the standard setting $\alpha = 0.5$ and $\alpha = 0.1$. BOA is run with the standard setting. The factorizations computed with $\alpha = 0.5$ and $\alpha = 0.1$ are very different. We show a typical part of both factorizations for f_{Trap} using a population of $N = 4000$. The number of parents is restricted to $k = 4$. The learning algorithm computed

$$\begin{aligned} p(\mathbf{x}) &= p(x_5)p(x_7|x_5)p(x_6|x_5, x_7, x_8)p(x_8|x_5, x_7, x_9)\cdots \quad \alpha = 0.5 \\ p(\mathbf{x}) &= p(x_0|x_2, x_3, x_{35}, x_{45})p(x_1|p(x_0, x_2, x_3, x_{44})\cdots \quad \alpha = 0.1 \end{aligned}$$

The standard penalty factor $\alpha = 0.5$ gives a sparse network, where 80 edges of a total of 87 are correct. In contrast, the small weight $\alpha = 0.1$ gives a dense network where each variable has the maximum number of allowed parents, namely 4. The total number of edges is 182.

problem	N	α	k	(c/t/m) edges	RIP viol.	best
$f_{Trap(5,10)}$	1000	0.5	4	20/45/80	7	no
	4000	0.5	4	80/87/20	0	yes
	2500	0.1	4	93/182/7	40	yes
(*BOA*)	3000		4	94/190/6	44	yes
	4000	0.1	2	70/95/30	17	yes
$F_{Iso(48)}$	1000	0.1	2	54/92/18	29	no
(*BOA*)	1500		2	52/95/20	?	yes
	3000	0.5	2	63/78/9	6	yes
	2000	0.1	2	60/94/12	24	yes
$Ising$	500	0.5	4	32/59/52	7	yes
	500	0.1	4	15/167/69	43	no
	4000	0.5	4	56/73/28	16	yes
	4000	0.1	4	61/174/23	45	yes
$K_{50;3}$	1000	0.5	4	43/63/75	16	no
	1000	0.1	4	57/179/61	44	yes
	10000	0.5	4	88/95/30	23	yes
	10000	0.1	4	100/177/18	77	yes

Table 3: Typical LFDA network: α structure penalty, k number of parents, graph:(correct(c)/total(t)/missed(m) edges), RIP violations, best: optimum found;

For f_{Trap} about $80\% - 90\%$ of the edges have to be correctly identified to find the optimum. Shown is also a run with only 2 allowed parents per variable. With $\alpha = 0.1$ the optimum is found with high probability. This means that the high order dependencies are not necessary for finding the optimum of this function. The results for f_{Iso} are similar. If the population size is increased to $N = 3000$ samples, the number of missed edges is reduced to 9.

For the $7 \cdot 7$ Ising problem, LFDA finds the optimum with a population size of 500. Only 32 edges of the Bayesian network are contained in G_{ADF}, 52 edges are missing. Here the run with $\alpha = 0.1$ does not find the optimum. A larger population size reduces the number of missed edges. With a population size of 4000 the learning algorithm computes a network with 70% correct edges. For Kauffman's function the run with $\alpha = 0.1$ finds the optimum with $N = 1000$, but the standard setting fails. Only 50% of the edges are contained in G_{ADF}. Even for $N = 10000$ and $\alpha = 0.1$ there are 18 missed edges. Here the limitations of the learning algorithm show up.

The results of BOA are comparable to running LFDA with $\alpha = 0.1$ (see the analysis by Lima et al. (2007); Yu (2006)). Recently an investigation of the networks computed by the successor program hBOA appeared in Hauschild et al. (2007). hBOA generates sparser networks than BOA. The results of hBOA are more similar to using LFDA with $\alpha = 0.5$.

Summary: Increasing the sample size reduces the number of missed edges (edges contained in G_{ADF} but not in the Bayesian network). LFDA might converge to the global optima with small sample sizes using Bayesian networks with a large number of missed edges. But this behavior depends on the problem to be optimized.

7 Numerical results

The computation time of learning in LFDA is bounded by $O(N * n^3)$. The implementation is not optimized because many different algorithms are implemented using the same data structures. Therefore the computation time gets very large for large problems[9]. For BOA the computational complexity of learning the Bayesian network is $O(N * n^2)$. Therefore we use BOA for larger problems.

For comparison with other papers we compute the probability of finding the optimum, mathematically $prob(\textit{finding the optimum}) > 1 - \delta$[10]. This criterion is the interesting one for optimization. The bound on the sample size N^* is estimated by computing the smallest sample size where the probability for finding the optimum is at least $1 - \delta$. For small problems 1000 runs have been made, 100 runs for problems with $n > 200$.

problem	n	N^*	$gen.$	FE	δ
$f_{Trap(5,n/5)}$	30	1800	4.3	9500	0.04
	60	4800	7.6	40200	0.03
	120	10000	11.9	118000	0.04
(*)	180	17000	17.4	300000	0.00
BOA	120	6000	24	78000	0.01
BOA(*)	240	14000	34	252000	0.00
BOA(*)	480	34000	49	884000	0.00
$f_{Iso(n)}$	31	1000	4.1	4100	0.06
	59	2500	7.1	19400	0.05
	121	5000	11.4	62800	0.09
BOA	121	7500	20.0	82500	0.00
BOA	201	15000	28.0	240000	0.00
BOA(*)	401	120000	53	3300000	0.00

Table 4: Numerically determined sample size bound N^*, truncation selection $\tau = 0.3$, for LFDA $\alpha = 0.5$, statistics from 100 runs, (*) single runs

For the function f_{Trap} both LFDA and BOA converge to the optimum. The sample size is bounded by $O(n \ln n)$. The scaling was tested for BOA till $n = 480$. For the function f_{Iso} we expect difficulties. This is indeed the case. Till $n = 201$ the sample size scales again like $O(n \ln n)$. But for $n = 401$ *BOA* did not find the global optimum for $N = 90000$. It succeeded with $N = 120000$. Thus we have a eight times larger sample size if the problem size is doubled (from $n = 201$ to $n = 401$). This shows that it is dangerous to extrapolate from small problem sizes to large ones. Nevertheless it is impressive that *BOA* finds the global optimum for $n = 401$ at all.

The numerical results for MARLEDA are disappointing compared to LFDA and BOA. For the function $f_{Trap(5,60)}$ with 300 variables MARLEDA did not find the global optimum, but instead computed optima nearby the trap optimum (Alden, 2007). The problem is obviously the implemented learning algorithm.

In table 5 the results of FDA are presented. We have shown that for functions fulfilling the assumptions of the factorization theorem with bounded clique size k a

[9] For large bi-partitioning problems LFDA has been optimized and runs in $O(N * n)$ time using sophisticated hashing techniques and restricting the possible edges to edges which are contained in the graph (Mühlenbein and Mahnig, 2002a).

[10] According to PAC convergence the criterion should be changed to $prob(|\mathbf{x}_{app} - \mathbf{x}_{opt}| < \epsilon) > 1 - \delta$

sample size scaling with at most $O(nm \ln nm)$ is sufficient to obtain convergence to the global optima.

problem	n	N^*	$gen.$	FE	δ
$f_{Trap(5,n/5)}$	50	375	9.9	3700	0.05
	100	500	16.1	8070	0.04
	200	800	25.3	20200	0.03
	400	1200	38.2	45800	0.04
(t)	200	800	16.3	13000	0.05
(t)	300	1300	18.4	23920	0.00
M^+	300	1400	15.0	21000	0.00
$f_{Iso(n)}$	49	440	10.1	4440	0.07
	99	940	16.1	15100	0.07
	201	1900	24.9	47400	0.07
	401	3800	36.2	138000	0.05
(t)	201	90000	12.8	1350000	0.60

Table 5: Numerically determined sample size bound N^*; (t) denotes truncation selection. M^+ is $MARLEDA^{+model}$ (Alden, 2007).

The sample size scales less than linear for the decomposable function f_{Trap}. The number of function evaluations scales about $O(m \ln m)$ where m is the number of subfunctions. This seems to be the best possible scaling for a random heuristic (Streeter, 2003).

For the function f_{Iso} the sample size scales about $O(n)$. For this function the *SDS* selection method is essential. Truncation selection for $n = 201$ needs a population size of $N = 90000$ compared to $N = 1900$ with *SDS*! For all other problems truncation selection is numerically more efficient (see also the discussion of the importance of selection in Hauschild et al. (2007)). In any case, the numerical determined sample size is $N \in O(n)$ at most.

Results are also shown for $MARLEDA^{+model}$. The author did not compute N^*. Reported is only the result for $f_{Trap(5,60)}$. The numerical results are almost identical to the results of FDA using truncation selection. This is to be expected, because both algorithms use an exact model.

In table 6 we present results of the Ising spin glass. For this problem our convergence results can not applied. For each problem size we investigated 10 randomly generated problems. The FDA factorization uses cliques of size 5 (Mühlenbein and Höns, 2006). This factorization covers all interactions, but violates the RIP. LFDA computes Bayesian networks with at most 4 parents. The table shows the results of two instances giving the best (1) and the worst results (2). We also report the average of the best solution found. For $n = 100$ there is no difference in the performance between the different problems. For $n = 225$ we generated a problem instance (2) where FDA always converged to a local optimum. The value of the local optimum differs from the global optimum at the fifth decimal place. Increasing the pop-size from $N = 4000$ to $N = 70000$ did not improve the results. Interestingly LFDA found the optimum in 3 out of 10 runs. For $n = 400$ the runtime for LFDA gets large. Here only the results for the problem giving the worst result with FDA is shown.

We have included a result from MARLEDA, despite it is a single run for a Ising model restricted to integer couplings ($J_{ij} \in \{-1, +1\}$). Because the result is so good, we hope to convince the researchers to do more experiments. Santana (2005) also reports

n	$Alg.$	$Pr.$	N	$gen.$	δ	average	best
100	FDA	1	1000	13.5	0.2	73.890	73.977(*)
100	LFDA	1	1500	14.1	0.1	73.947	73.977(*)
100	FDA	2	3500	14.5	0.2	73.342	73.359(*)
100	LFDA	2	3000	13.1	0.1	73.353	73.359(*)
225	FDA	1	3000	24.5	0.2	168.494	168.540(*)
225	LFDA	1	3000	25.0	1.0	167.449	167.757
225	FDA	2	4000	23.5	1.0	164.441	164.441
225	FDA	2	70000	22.5	1.0	164.441	164.441
225	LFDA	2	4000	25.5	0.7	164.304	164.473(*)
400	FDA	1	5000	33.1	0.8	298.874	300.504(*)
400	LFDA	1	5000	35.9	0.9	297.872	300.504(*)
400	M^+	2	900	20.0	-	-	(*)

Table 6: Best(1) and worst results(2) out of 10 randomly generated spin glass problems; (*) global optimum; M^+ is $MARLEDA^{+model}$; results from (Alden, 2007)

results for his MN-GIBBS implementation *MN-EDA*^{f} for very small Ising problems (up to $n = 64$). He observes that *MN-EDA*^{f} is by far the best algorithm among the algorithm he compared.

For BOA and hBOA the scaling of the sample size and the number of function evaluations has been intensively investigated by Pelikan and his coworkers (Pelikan and Goldberg, 2006; Pelikan and Hartmann, 2006; Hauschild et al., 2007). But even in very good experimental work sometimes unjustified conclusions are made. We just cite: "hBOA is able to solve 2D Ising spin glasses in polynomial time"(p. 257 Hauschild et al. (2007)). The statement should be more precise: hBOA is able to solve all problems tested in polynomial time. Later we read (p.259): "While for 2D Ising spin glasses it is unclear what is an ideal probabilistic model, the probability models are shown to correspond closely to the structure of the underlying problem." The empirical observation supports our result. We have shown in this paper that for provable convergence the ideal probabilistic model corresponds to the structure of the ADF.

8 Conclusion and Outlook

The family of EDAs are well founded in statistical learning theory. Using known results from Bayesian networks and Markov networks we have been able to bound the sample size needed for PAC convergence to the global optima. The numerical experiments confirm the theoretical results.

Numerically the best results are obtained by using EDAs which use optimal networks derived from the ADF. Especially efficient is here the algorithm FDA using the sampling method PLS with an exact factorization fulfilling the RIP. For complex problems the algorithm MN-GIBBS using Markov networks seems most promising. The numerical efficiency of MN-GIBBS can be substantially increased if state-of-the-art Gibbs samplers are used. Here lots of experiments are needed.

The results of algorithms learning Bayesian networks from data are astonishingly good. But they cannot match the performance of algorithms using optimal networks. They have problems with artificial functions like f_{Iso} For this class of functions the learned network has to contain a substantial fraction of the edges of the interaction graph G_{ADF}. But for many practical problems the number of correct edges can be small in order to find the optimum.

A promising new development are programs which compute the structure of the fitness function from data by probing. A good candidate is the algorithm of Wright and Pulavarty (2005). The computational complexity of the learning algorithm is bounded by $\Omega(n^2 \ln n)$. This is the lowest bound reported for an algorithm computing the correct structure of the fitness function. This learning algorithm can be used as a preprocessing step for FDA or MN-GIBBS. An EDA implementation is urgently needed.

All EDA algorithms can run with *local hill climbing* algorithms. This is very successful in combinatorial optimization using genetic algorithms (Mühlenbein, 1991). For EDAs the importance of local hill climbing was demonstrated by Mühlenbein and Mahnig (2002a); Pelikan and Hartmann (2006). Convergence theorems for this class of algorithms are currently under investigation.

The interested reader can download our free EDA software by contacting me.

A Proof of theorem 10

The proof is based on three lemmas. The first lemma is a fundamental theorem of PAC learning (Kearns and Vazirani (1994)).

Lemma 1 *Let p be a distribution over $\{0,1\}^k$. Let $\{\mathbf{x}^i\}_{i=1}^N$ be i.i.d. samples from p, let $\hat{p}$ be the empirical distribution. Let any $0 < \epsilon < 1$,$0 < \delta < 1$ be given. Then for*

$$D(p||\hat{p}) \leq \epsilon$$

to hold with probability $1 - \delta$ it suffices that

$$N \geq \frac{1}{\epsilon}(\ln|H| + \ln\frac{1}{\delta}) \tag{45}$$

where $|H| = 2^{2^k}$ is the size of the hypothesis space.

Lemma 2 *Let any $0 < \epsilon < 1$, $0 < \delta < 1$ be given. Let*

$$p(\mathbf{x}) = \prod_{j=1}^{m} p(\mathbf{x}_{b_j}|\mathbf{x}_{c_j})$$

fulfill the RIP. Let $k = \max_j val(X_{b_j}, X_{c_j})$. Let $\{\mathbf{x}^i\}_{i=1}^N$ be samples generated from p. Compute $\hat{p}(\mathbf{x}_{b_j}|\mathbf{x}_{c_j}$ using the samples. Let

$$\hat{p}(\mathbf{x}) = \prod_{j=1}^{m} \hat{p}(\mathbf{x}_{b_j}|\mathbf{x}_{c_j})$$

Then for

$$D(p||\hat{p}) \leq \epsilon$$

to hold with probability $1 - \delta$ it suffices that

$$N \geq \frac{m}{\epsilon}\left(2^k \ln 2 + \ln\frac{m}{\delta}\right) \tag{46}$$

Proof: We have

$$\begin{aligned}
D(p||\hat{p}) &= \sum_x \prod_{j=1}^m p(\mathbf{x}_{b_j}|\mathbf{x}_{c_j}) \left(\sum_{j=1}^m \ln p(\mathbf{x}_{b_j}|\mathbf{x}_{c_j}) - \sum_{j=1}^m \ln \hat{p}(\mathbf{x}_{b_j}|\mathbf{x}_{c_j}) \right) \\
&= \sum_{j=1}^m \sum_{x_{b_j}, x_{c_j}} p(\mathbf{x}_{b_j}, \mathbf{x}_{c_j}) \left(\ln p(\mathbf{x}_{b_j}, \mathbf{x}_{c_j}) - \ln \hat{p}(\mathbf{x}_{b_j}, \mathbf{x}_{c_j}) \right) \\
&- \sum_{j=1}^m \sum_{x_{c_j}} p(\mathbf{x}_{c_j}) \left(\ln p(\mathbf{x}_{c_j}) - \ln \hat{p}(\mathbf{x}_{c_j}) \right) \\
&= \sum_{j=1}^m D\left(p(X_{b_j}, X_{c_j}) || \hat{p}(X_{b_j}, X_{c_j})\right) - \sum_{j=1}^m D(p(X_{c_j})||\hat{p}(X_{c_j})) \\
&\leq \sum_{j=1}^m D\left(p(X_{b_j}, X_{c_j}) || \hat{p}(X_{b_j}, X_{c_j})\right)
\end{aligned}$$

because $D(p||\hat{p}) \geq 0$. We next apply the Union bound and lemma 1 and obtain

$$D(p||\hat{p}) \leq m\epsilon'$$

with probability at least $1 - m\delta'$ if N fulfills (46). We set $\epsilon = m\epsilon'$ and $\delta = m\delta'$ and the lemma is proven. ◇

FDA does not sample from the true Boltzmann distribution, but it uses Boltzmann selection instead. For later use we change the error function to the L_1 Norm $|p - \hat{p}|_1 = \sum_x |p(\mathbf{x}) - \hat{p}(\mathbf{x})|$.

Lemma 3 *Let the assumptions of lemma 2 be fulfilled. Sample $\{\mathbf{x}^i\}_{i=1}^N$ points using the Boltzmann distribution p_β. Let $\Delta\beta$ be given. From the sample select N points according to the probabilities (Boltzmann selection)*

$$p_\beta(\mathbf{x}^i) = \frac{e^{\Delta\beta f(\mathbf{x}^i)}}{\sum_{i=1}^N p_\beta(\mathbf{x}^i) e^{\Delta\beta f(\mathbf{x}^i)}}$$

Let there at least $N/4$ different points be selected. Compute the empirical distribution

$$\hat{p}(\mathbf{x}) = \prod_{j=1}^m \hat{p}(\mathbf{x}_{b_j}|x_{c_j})$$

Then for

$$|p_{\beta+\Delta\beta} - \hat{p}|_1^2 = \leq \epsilon \tag{47}$$

to hold with probability $1 - \delta$ it suffices that

$$N \geq \frac{8 \ln 2}{\epsilon} m \left(2^k \ln 2 + \ln \frac{m}{\delta} \right) \tag{48}$$

Proof: Because Boltzmann selection selects at least $N/4$ different samples, this is equivalent to sample $N/4$ points from the true distribution $p_{\beta+\Delta\beta}$. We apply lemma 2 using $N/4$ and obtain

$$D(p||\hat{p}) \leq \epsilon'$$

with probability at least $1 - \delta$ if

$$N \geq \frac{4m}{\epsilon'} \left(2^k \ln 2 + \ln \frac{m}{\delta} \right) \tag{49}$$

Next we use the bound ((Cover and Thomas, 1989), lemma 12.6.1)

$$|p - \hat{p}|_1^2 \leq 2 \ln 2 D(p||\hat{p}) \tag{50}$$

We set $\epsilon = 2 \ln 2 \epsilon\prime$ and the lemma is proven. $\diamond$

We are now ready to prove the theorem.

Proof of theorem 10: The first population is generated using the uniform random distribution. Using lemma 3 this gives an error of ϵ. Using the lemma iteratively, each new generation adds an error of ϵ in the worst case. The Union bound gives

$$|p_g - \hat{p}_g|_1^2 \leq g\epsilon \tag{51}$$

with probability at least $1 - g\delta'$. Setting $\delta = g\delta'$ gives the bound 32. $\diamond$

B The computational complexity of the 2-D Ising model

Barahona (1982) has shown the following surprising result: the 2D Ising model (i.e. $J_{ij} \in \{-1, 0, +1\}$) without external magnetic field ($h_i = 0$) can be solved in polynomial time, whereas the problem with external magnetic field is *NP-hard*. In fact, Barahona (1982) proved the following theorem

Theorem 12 *Given a planar graph $G = (V, E)$ with all its vertices of degree three to find the minimum value of*

$$H = \sum_{(i,j) \in E} s_i s_j + \sum_{i \in V} s_i \tag{52}$$

is NP-hard.

Note that each spin has only three neighbors. For the Gaussian Ising spin glass the question of computational complexity is still open.

This result poses a challenge to our major convergence result for FDA formulated in theorem 8. Any *exact* factorization of the 2D Ising models contains a clique of size $O(\sqrt{n})$ (Gao and Culberson, 2005), independent of an external magnetic field. This means that FDA needs an *exponential* effort for provable convergence to the global optima. Since any exact factorization has to contain G_{ADF} FDA cannot distinguish between different classes of Ising models having the same interaction graph.

References

Abbeel, P., Koller, D., and Ng, A. (2006). Learning factor graphs in polynomial time & sample complexity. *Journ. Machine Learning Research*, 7:1743–1780.

Alden, M. E. (2007). *MARLEDA : Effective Distribution Estimation through Random Fields*. PhD thesis, University of Texas at Austin, Austin, USA.

Andrieux, C., de Freitas, N., Doucet, A., and Jordan, M. (2003). An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43.

Barahona, F. (1982). On the computational complexity of the Ising spin glass models. *J. Phys. A: Math. Gen.*, 15:3241–3253.

Chickering, D., Heckerman, D., and Meek, C. (2004). Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, 5:1287–1330.

Cooper, G. (1990). The computational complexity of probabilistic inference using belief networks. *Artificial Intelligence*, 42:393–405.

Cover, T. M. and Thomas, J. (1989). *Elements of Information Theory*. Wiley, New York.

Dasgupta, S. (1997). The sample complexity of learning fixed structure Bayesian networks. *Machine Learning*, 29(2-3):165–180.

Deb, K. and Goldberg, D. E. (1993). Analyzing deception in trap functions. In Whitley, L. D., editor, *Foundations of Genetic Algorithms*, pages 93–108, San Mateo. Morgan-Kaufman.

Etxeberria, R. and Larrañaga, P. (1999). Global optimization using Bayesian networks. In Ochoa, A., Soto, M. R., and Santana, R., editors, *Proceedings of the Second Symposium on Artificial Intelligence (CIMAF-99)*, pages 151–173, Havana, Cuba.

Frigessi, A., Martinelli, F., and Stadler, J. (1997). Computational complexity of Markov chain Monte Carlo methods for finite Markov random fields. *Biometrika*, 84:1–18.

Gao, Y. and Culberson, J. (2005). Space complexity of estimation of distribution algorithms. *Evolutionary Computation*, 13(1):125–143.

Geman, D. and Geman, S. (1984). Stochastic relaxation, Gibbs sampling, and the restauration of images. *IEEE Transaction on Pattern Recognition and Machine Intelligence*, 6:721–741.

Gibbs, A. (2000). Bounding the convergence time of the Gibbs sampler in Bayesian image restoration. *Biometrika*, 87:749–766.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996). *Markov Chain Monte Carlo in Practice*. Chapman & Hall, London.

Guo, H. and Hsu, W. (2002). A survey of algorithms for real-time Bayesian network inference. In *KDD-02/UAI-02 Workshop on Real-Time Decision Support*, http://citeseer.ist.psu.edu/Guo2survey.html.

Harik, G., Lobo, F., and K, S. (2006). Linkage learning via probabilistc modeling in the extended compact genetic algorithm (ECGA). In Pelikan, M., Sastry, K., and Cantu-Paz, E., editors, *Scalable Optimization via Probabilistic Modeling*, Studies in Computational Intelligence, pages 39–61. Springer, Berlin.

Hauschild, M., Pelikan, M., Lima, C., and Sastry, K. (2007). Analyzing probabilistic models in hierarchical BOA on trap and spin glasses. In *Proceedings of GECCO'07*, pages 523–530.

Heckendorn, R. B. and Wright, A. H. (2004). Efficient linkage discovery by limited probing. *Evolutionary Computation*, 12:517–545.

Heckerman, D., Geiger, D., and Chickering, D. (1995). Learning Bayesian networks:The combination of knowledge and and statistical data. *Machine Learning*, 20:197–243.

Henrion, M. (1988). Propagating uncertainty in Bayesian networks by Probabilistic Logic Sampling. In Lemmar, J. and Kanal, L., editors, *Uncertainty in Artificial Intelligence*, pages 149–181, New York. Elsevier.

Höns, R. (2006). *Estimation of Distribution Algorithms and Minimum Relative Entropy*. PhD thesis, University of Bonn, Bonn, Germany.

Jensen, F. V. and Jensen, F. (1994). Optimal junction trees. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 360–366, Seattle.

Jordan, M. I., editor (1999). *Learning in Graphical Models*. MIT Press, Cambridge.

Kearns, M. and Vazirani, U. (1994). *An Introduction to Computational Learning Theory*. MIT Press, Cambridge.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science*, 220:671–680.

Kschischang, F. R., Frey, B. J., and A.Loeliger, H. (2001). Factor graphs and the sum-product algorithm. *IEEE Transactions on Information Theory*, 47(2):498–519.

Larrañaga, P. and Lozano, J. A., editors (2002). *Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation*. Kluwer Academic Publishers, Boston/Dordrecht/London.

Lauritzen, S. L. (1996). *Graphical Models*. Clarendon Press, Oxford.

Lima, C. F., Pelikan, M., Goldberg, D. E., Lobo, F. G., Sastry, K., and Hauschild, M. (2007). Influence of selection and replacement strategies on linkage learning in BOA. IlliGAL Report No. 2007013, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL.

Mahnig, T. and Mühlenbein, H. (2001). A new adaptive Boltzmann selection schedule SDS. In *Proceedings Congress on Evolutionary Computation 2001*, pages 121–128. IEEE.

Mühlenbein, H. (1991). Evolution in time and space - the parallel genetic algorithm. In Rawlins, G., editor, *Foundations of Genetic Algorithms*, pages 316–337, San Mateo. Morgan-Kaufman.

Mühlenbein, H. and Höns, R. (2005). The estimation of distributions and the minimum relative entropy principle. *Evolutionary Computation*, 13(1):1–27.

Mühlenbein, H. and Höns, R. (2006). The factorized distribution algorithm and the minimum relative entropy principle. In Pelikan, M., Sastry, K., and Cantu-Paz, E., editors, *Scalable Optimization via Probabilistic Modeling*, Studies in Computational Inteligence, pages 11–37. Springer, Berlin.

Mühlenbein, H. and Mahnig, T. (2002a). Evolutionary optimization and the estimation of search distributions with applications to graph bipartitioning. *Journal of Approximate Reasoning*, 31(3):157–192.

Mühlenbein, H. and Mahnig, T. (2002b). Mathematical analysis of evolutionary algorithms. In Ribeiro, C. C. and Hansen, P., editors, *Essays and Surveys in Metaheuristics*, Operations Research/Computer Science Interface Series, pages 525–556. Kluwer Academic Publisher, Norwell.

Mühlenbein, H. and Mahnig, T. (2003). Evolutionary algorithms and the Boltzmann distribution. In DeJong, K., Poli, R., and Rowe, J. C., editors, *Foundations of Genetic Algorithms 7*, pages 525–556. Morgan Kaufmann Publishers, San Francisco.

Mühlenbein, H., Mahnig, T., and Ochoa, A. (1999). Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247.

Mühlenbein, H. and Paaß, G. (1996). From recombination of genes to the estimation of distributions I. binary parameters. In Voigt, H.-M., Ebeling, W., Rechenberg, I., and Schwefel, H.-P., editors, *Lecture Notes in Computer Science 1141: Parallel Problem Solving from Nature - PPSN IV)*, pages 178–187, Berlin. Springer-Verlag.

Mühlenbein, H. and Schlierkamp-Voosen, D. (1993). Predictive models for the breeder genetic algorithm. *Evolutionary Computation*, 1(1):25–49.

Munetomo, M. and Goldberg, D. (1999). Linkage identification by non-monotonicity detection for overlapping functions. *Evolutionary Computation*, 7:377–398.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems*. Morgan-Kaufmann, San Mateo.

Pelikan, M. and Goldberg, D. (2006). Hierarchical Bayesian optimization algorithm. In Pelikan, M., Sastry, K., and Cantu-Paz, E., editors, *Scalable Optimization via Probabilistic Modeling*, Studies in Computational Intelligence, pages 63–90. Springer, Berlin.

Pelikan, M. and Goldberg, D. E. (2000). Linkage problem, distribution estimation, and Bayesian networks. *Evolutionary Computation*, 8:311–340.

Pelikan, M. and Goldberg, D. E. (2002). A survey of optimization by building and using probabilistic models. *Computational Optimization and Applications*, 1:5–20.

Pelikan, M. and Hartmann, A. (2006). Searching for ground states of Ising spin glasses with hierarchical BOA and Cluster Exact Approximation. In Pelikan, M., Sastry, K., and Cantu-Paz, E., editors, *Scalable Optimization via Probabilistic Modeling*, Studies in Computational Intelligence, pages 315–332. Springer, Berlin.

Santana, R. (2003). A Markov network based factorized distribution algorithm for optimization. In *Proceedings of the 14th European Conference on Machine Learning)*, volume 2837 of *Lecture Notes in Artificial Intelligence*, pages 337–348, Berlin. Springer-Verlag.

Santana, R. (2005). Estimation of distribution algorithms with Kikuchi approximations. *Evolutionary Computation*, 13(1):67–97.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics*, 7:461–464.

Shakya, S. K. (2006). *DEUME : A framework for an Estimation of Distribution Algorithm based on Markov Random Fields*. PhD thesis, Robert Gordon University, Aberdeen, Scotland.

Streeter, M. J. (2003). Upper bounds on the time and space complexity of optimizing additively separable functions. In K. Deb et al., editor, *Proceedings of GECCO-2003*, Lecture Notes in Computer Science 3103, pages 186–197. Springer Press.

Thierens, D. and Goldberg, D. (1994). Convergence models of genetic algorithm selection schemes. In Davidor, Y. and Schwefel, H.-P., editors, *Parallel Problem Solving from Nature - PPSN III)*, Lecture Notes in Computer Science 866, pages 116–121, Berlin. Springer-Verlag.

Tsamardinos, I., Brown, L., and Alifernis, C. (2006). The max-min hill-climbing Bayesian network structure algorithm. *Machine Learning*, 65(1):31–78.

Wright, A. H. and Pulavarty, S. S. (2005). Estimation of distribution algorithm based on linkage discovery and factorization. In H.G. Beyer et al., editor, *Proceedings of GECCO-2005*, pages 695–703. ACM Press.

Wright, A. H., Thompson, R. K., and Zhang, J. (2000). Stochastic relaxation, Gibbs sampling, and the restauration of images. *IEEE Transaction on Evolutionary Computation*, 6:373–379.

Xiang, Y., Wong, S. K. M., and Cercone, N. (1997). A ‘microscopic’ study of minimum entropy search in learning decomposable Markov networks. *Machine Learning*, 26:65–92.

Yu, T.-L. (2006). *A matrix approach for finding extrema: problems with modularity, hierarchy and overlap*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana, Illinois.

Zhang, Q. (2004). On the convergence of a factorized distribution algorithm with truncation selection. *Complexity*, 9(4):17–23.

Zhang, Q. and Mühlenbein, H. (2004). On the convergence of a class of estimation distribution algorithms. *IEEE Trans. on Evolutionary Computation*, 8(2):17–23.