# The Estimation of Distributions and the Minimum Relative Entropy Principle

**Heinz Mühlenbein** heinz.muehlenbein@ais.fraunhofer.de
Fraunhofer Institute for Autonomous Intelligent Systems
53754 Sankt Augustin, Germany

**Robin Höns** robin.hoens@ais.fraunhofer.de
Fraunhofer Institute for Autonomous Intelligent Systems
53754 Sankt Augustin, Germany

**Abstract**

Estimation of Distribution Algorithms (EDA) have been proposed as an extension of genetic algorithms. In this paper we explain the relationship of EDA to algorithms developed in statistics, artificial intelligence, and statistical physics. The major design issues are discussed within a general interdisciplinary framework. It is shown that *maximum entropy* approximations play a crucial role. All proposed algorithms try to minimize the Kullback-Leibler divergence $KLD$ between the unknown distribution $p(\mathbf{x})$ and a class $q(\mathbf{x})$ of approximations. However, the Kullback-Leibler divergence is not symmetric. Approximations which suppose that the function to be optimized is additively decomposed (ADF) minimize $KLD(q||p)$, the methods which learn the approximate model from data minimize $KLD(p||q)$. This minimization is identical to maximizing the *log-likelihood*. In the paper three classes of algorithms are discussed. *FDA* uses the ADF to compute an approximate factorization of the unknown distribution. The factors are marginal distributions, whose values are computed from samples. The second class is represented by the Bethe-Kikuchi approach which has recently been rediscovered in statistical physics. Here the values of the marginals are computed from a difficult constrained minimization problem. The third class learns the factorization from the data. We analyze our learning algorithm *LFDA* in detail. It is shown that learning is faced with two problems: first, to detect the important dependencies between the variables, and second, to create an acyclic Bayesian network of bounded clique size.

**Keywords**

Estimation of distributions, Boltzmann distribution, factorization of distributions, maximum entropy principle, minimum relative entropy, minimum log-likelihood ratio, Bayesian information criterion, Bethe approximation.

## 1 Introduction

The *Estimation of Distribution* (EDA) family of population based search algorithms was introduced by Mühlenbein and Paaß (1996) as an extension of genetic algorithms.[1] The following observations lead to this proposal. First, genetic algorithm 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.

[1] Mühlenbein and Paaß (1996) have named them *conditional distribution algorithms*.

 

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

In (Mühlenbein and Paaß, 1996) the distribution was estimated by computationally intensive Monte Carlo methods. The distribution was restricted to tree-like structures. It has been shown by Mühlenbein et al. (1999) 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 of estimating empirical distributions has been investigated independently in several scientific disciplines. In this paper we will show how results in statistics, belief networks and statistical physics can be used to understand and further develop EDA. In fact, an interdisciplinary research effort is well under way which cross-fertilizes the different disciplines.

Unfortunately each discipline uses a different language, has a slightly different application, and has developed different algorithms. In EDA we have to sample from a distribution, in belief networks one computes a single marginal distribution $p(\mathbf{y}|\mathbf{z})$ for new evidence $\mathbf{z}$, and statistical physicists want to compute the free energy of a Boltzmann distribution. Thus the algorithms developed for belief networks concentrate on computing a single marginal distribution, whereas for EDA we want to generate samples in areas of high fitness values. All disciplines face the problem to develop fast algorithms to compute marginal distributions. The foundation of the theory is the same for all disciplines. It is based on graphical models and their decomposition. We hope that the readers are interested in accompanying us on our journey through the different disciplines. We will leave out a discussion of the approaches in probabilistic logic to simplify the presentation.

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 (Jordan, 1999). For large real life applications often a hybrid between these two approaches is most successful (Mühlenbein and Mahnig, 2002a).

The paper is intended as a short introduction to the theory of EDA. It is not intended as a survey of ongoing research. Here an excellent overview is already available (Larrañaga and Lozano, 2002). For simplicity we will only consider discrete variables.

The outline of the paper is as follows. In section 2 the basic steps to derive the Factorized Distribution Algorithm *FDA* are recapitulated. A factorization theorem will be discussed which uses the structure of the function to be optimized to factor the distribution. In section 2.2 the junction tree algorithm is described which computes an exact factorization by decomposing graphical models. Unfortunately many important problems do not allow an exact factorization useful for numerical computations. In section 3 the estimation problem is generalized. Here the concept of *maximum entropy* distributions is explained.

In section 4 the methods developed in statistical physics are described. In this approach the marginals are not computed from data, but from the known expression of the function. In section 5 the learning of models from samples of high fitness values is described. Then we compare the different approaches presented using a simple example. In section 7 our learning algorithm *LFDA* is analyzed and its behavior and

performance compared to *FDA* is investigated.

## 2 Factorization of the Search Distribution

EDA has been derived from a search distribution point of view. Here we will simply recapitulate the major steps published in (Mühlenbein et al., 1999; Mühlenbein and Mahnig, 2000; Mühlenbein and Mahnig, 2002a). We will use 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 $x_i$.
Let a function $f : \mathbf{X} \rightarrow \mathbb{R}_{\geq 0}$ 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*

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

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

The Boltzmann distribution concentrates on increasing $\beta$ around the global optima of the function. Obviously, the distribution converges for $\beta \rightarrow \infty$ 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 $\beta$, optimization would be an easy task. But the computation of the partition function 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 is easy to prove.

**Theorem 3.** *If $p_\beta(\mathbf{x})$ is a Boltzmann distribution, then $p^s(\mathbf{x})$ is a Boltzmann distribution with inverse temperature $\beta(t+1) = \beta(t) + \Delta\beta(t)$.*

Algorithm 1 describes *BEDA*, the Boltzmann Estimated Distribution Algorithm.

*BEDA* is a conceptional algorithm, because the calculation of the distribution requires a sum over exponentially many terms. In the next section we transform *BEDA* into a practical numerical algorithm.

### 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 $\beta$ instead of the temperature.

**Algorithm 1** *BEDA* – Boltzmann Estimated Distribution Algorithm

1 $t \Leftarrow 1$. Generate $N$ points according to the uniform distribution $p(\mathbf{x}, 0)$ with $\beta(0) = 0$.
2 **do** {
3 With a given $\Delta\beta(t) > 0$, let
$$p^s(\mathbf{x}, t) = \frac{p(\mathbf{x}, t)e^{\Delta\beta(t)f(\mathbf{x})}}{\sum_{\mathbf{y}} p(\mathbf{y}, t)e^{\Delta\beta(t)f(\mathbf{y})}}.$$
4 Generate $N$ new points according to the distribution $p(\mathbf{x}, t+1) = p^s(\mathbf{x}, t)$.
5 $t \Leftarrow t + 1$.
6 } **until** (stopping criterion reached)

**Definition 4.** *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*

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

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

**Definition 5.** *Let an ADF be given. Then the graph $G_{ADF}$[3] is defined as follows: The vertices represent the variables of the ADF . Two vertices are connected by an arc iff the corresponding variables are contained in a common sub-function.*

Given an ADF we want to estimate the Boltzmann distribution (2) using a product of marginals. We need the following sets:

**Definition 6.** *Given $s_1, \ldots, s_m$, we define for $i = 1, \ldots, m$ the sets $d_i$, $b_i$ and $c_i$:*

$$d_i := \bigcup_{j=1}^{i} s_j, \qquad b_i := s_i \setminus d_{i-1}, \qquad c_i := s_i \cap d_{i-1} \tag{5}$$

*We demand $d_m = \{1, \ldots, n\}$ 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).*

The next definition is stated a bit informally.

**Definition 7.** *A set of marginal distributions $\tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})$ is called* consistent *if the marginal distributions fulfill the laws of probability, e.g.*

$$\sum_{\mathbf{x}_{b_i}, \mathbf{x}_{c_i}} \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) = 1 \tag{6}$$

$$\sum_{\mathbf{x}_{b_i}} \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) = \tilde{q}(\mathbf{x}_{c_i}) \tag{7}$$

**Proposition 8.** *Let a consistent set of marginal distributions $\tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})$ be given. If $b_i \neq \emptyset$ then*

$$q_\beta(\mathbf{x}) = \prod_{i=1}^{m} \tilde{q}_\beta(\mathbf{x}_{b_i} | \mathbf{x}_{c_i}) \tag{8}$$

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

*defines a valid distribution ($\sum q_\beta(\mathbf{x}) = 1$). Note that by definition $\mathbf{x}_{c_1} = \emptyset$ and $\tilde{q}(\emptyset) = 1$. Furthermore*

$$q_\beta(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}) = \tilde{q}_\beta(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}), \quad i = 1, \dots m \tag{9}$$

*whereas in general*

$$q_\beta(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) \neq \tilde{q}_\beta(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}), \quad i = 1, \dots m \tag{10}$$

The proof follows from the definition of marginal probabilities. The proof of equation (9) is somewhat technical, but straightforward. The inequality (10) is very important, but hardly known. We need an additional constraint in order that the marginal distributions become equal. This has been proven by Mühlenbein et al. (1999).

**Theorem 9** (Factorization Theorem)**.** *Let $f(\mathbf{x}) = \sum_{i=1}^{m} f_i(\mathbf{x}_{s_i})$ be an additive decomposition. If*

$$\forall i = 1, \dots, m; \quad b_i \neq \emptyset \tag{11}$$

$$\forall i \geq 2 \; \exists j < i \;\; \textit{such that } c_i \subseteq s_j \tag{12}$$

*then*

$$q_\beta(\mathbf{x}) = \prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}) = \frac{\prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})}{\prod_{i=1}^{m} p_\beta(\mathbf{x}_{c_i})} = p_\beta(\mathbf{x}) \tag{13}$$

Thus the true distribution can be obtained from some of its *marginal distributions*. There always exists a factorization fulfilling the assumptions of the factorization theorem. We just mention

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \cdots p(x_n|x_1, \dots, x_{n-1}) \tag{14}$$

But this factorization uses marginal distributions of size $O(n)$, thus the computation is exponential in $n$. Therefore we are looking for factorizations where the size of the marginals is bounded, independent of $n$.

**Definition 10.** *The constraint defined by equation* (12) *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 connection of the factorization theorem to research done in non-sequential dynamic programming is not well known. If a factorization with *RIP* is possible, we can compute the maximum directly. In fact, the following maximization theorem has been proven earlier than the factorization theorem by Bertelé and Brioschi (1972).

**Theorem 11** (Maximization Theorem)**.** *Let the assumptions of the factorization theorem be fulfilled. If the factorization is polynomially bounded, then $\max p_\beta(\mathbf{x})$ and $\operatorname{argmax} p_\beta(\mathbf{x})$ can be computed recursively in polynomial time.*

A proof of the theorem using graphical models can be found in Jordan (1999). The basic idea is that maximization can be done in the same manner as marginalization. We have

$$\max_{a,b,c} f_1(a, b) f_2(b, c) = \max_{a,b} f_1(a, b) \max_{c} f_2(b, c)$$

For later use we just remark that any *FDA* factorization can be transformed into an acyclic Bayesian network (acBN) (see equation (14)).

$$q_\beta(\mathbf{x}) = \prod_{i=1}^{n} q_\beta(x_i|\pi_i) \tag{15}$$

where $\pi_i$ are called the parents of $x_i$.

We next describe a well-known algorithm to obtain a factorization with marginals of small size and fulfilling the *RIP* given an arbitrary graph.

### 2.2 Computing a Factorization by Junction Trees

The algorithm is defined for any *graphical model* $G$, an example is $G_{\mathrm{ADF}}$. In order to find the separators $c_i$ the method computes cliques and generates a junction tree $J$. A junction tree is an undirected tree the nodes of which are clusters of variables. The clusters satisfy the *junction property*: For any two clusters $a$ and $b$ and any cluster $h$ on the unique path between $a$ and $b$ in the junction tree

$$a \cap b \subseteq h \tag{16}$$

The edges between the clusters are labeled with the intersection of the adjacent clusters; we call these labels *separating sets* or *separators*.

**Remark:** *The junction property is equivalent to the running intersection property* (12).

A junction tree is constructed from the graphical model by the following steps:

**Triangulating the graph $G$:** A graph is triangulated if it contains no chord-less circle with more than three vertices. An algorithm for adding the necessary edges is described by Huang and Darwiche (1996).

**Finding the cliques:** A clique $C$ in a graph is a maximal totally connected subgraph. That means that in $C$ every node is connected to every other node in $C$, and there is no clique $C'$ which contains $C$.

**Generating the clusters:** For each clique generate a cluster containing its variables. This cluster will become a node of the junction tree $J$.

**Building the junction tree:** Find pairs of clusters with maximal intersection and connect them. Label the edge with the separating set. Repeat this until the tree is complete.

This results in a tree which fulfills the junction property. There is plenty of literature available about this method, e.g., Lauritzen (1996); Huang and Darwiche (1996); Jensen and Jensen (1994).

A simple example to demonstrate this method is a circular graph $G$. It can be triangulated by connecting one node with all other nodes. The resulting junction tree is shown for 8 nodes in figure 1. The distribution can be factored into the cliques given by the clusters of the junction tree.

$$p(\mathbf{x}) = p(x_1, x_2, x_8) \prod_{i=3}^{7} p(x_i | x_{i-1}, x_8) \tag{17}$$

However, the junction tree contains non-local marginal distributions of order three.[4] One can show that there exists no exact factorization of a 1-D circle using bi-variate marginals only. For EDA this poses no problem, because all required marginals can be computed from the selected sample. But for larger marginals more samples are needed for a reliable estimate. If the graph contains many loops, the junction tree might be difficult to compute or might contain marginals with an exponential number of variables. We will investigate this problem for a 2-D grid.

[4] Local marginals are defined in a 1-D neighborhood like $p(x_{i-1}, x_i, x_{i+1})$

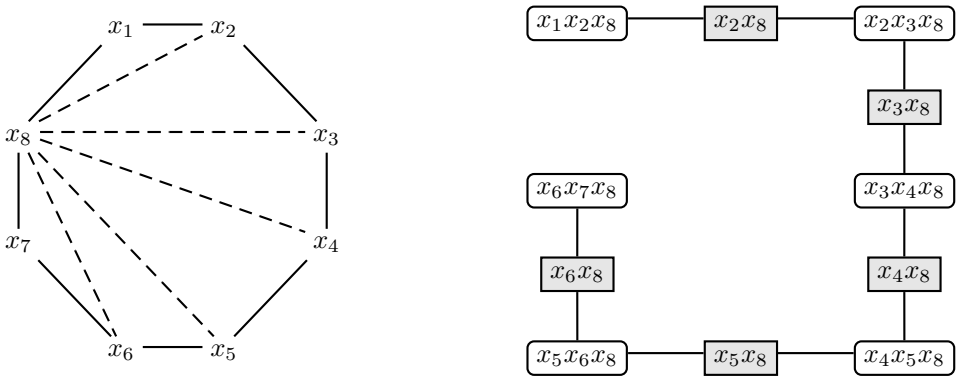


Figure 1: Graph model with triangulation and junction tree for a 1-D bi-variate circle. The left figure shows the graph $G$; the dashed lines are inserted for the triangulation. The cliques of the triangulated graph are the clusters of the junction tree $J$ (right figure, white boxes). The separators are the shaded boxes.

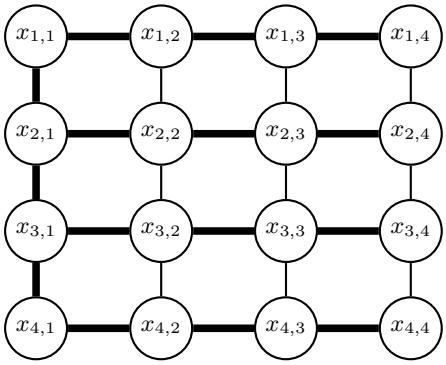


Figure 2: Graph model for a 2-D grid. The thick lines give a possible spanning tree.

### 2.3 The 2-D Grid

Let there be a 2-D grid of variables $x_{i,j}$, $i, j = 1, \ldots, n$. Let the fitness function be composed of the sub-functions of pairs of neighboring variables, $x_{i,j}, x_{i+1,j}$ and $x_{i,j}, x_{i,j+1}$. The goal is to compute a factorized distribution which is a good approximation to the true distribution.

An exact factorization can be found with a junction tree. The difficulty of the computation lies in the triangulation of the graphical model. One valid triangulation uses the rows of the grid. Each variable is connected with all variables in the same row and the neighboring rows. This adds $O(n)$ edges to the graph. The cliques in the junction tree consist of pairs of neighboring rows and have size $2n$. Thus the exact factorization is not polynomially bounded.

Therefore it is advisable to look for approximations. A very straightforward approximation is to leave out some of the marginals and build a spanning tree of the grid. This could be the vertical edges in the first column and all the horizontal edges, forming a big "E" (see thick lines in figure 2).

Given this subset of the edges and disregarding the rest, we can define the follow-


Figure 3: A $3 \times 3$ grid and its factorization using (19).

ing distribution:

$$q(\mathbf{x}) = p(x_{1,1}, x_{2,1}) \prod_{i=2}^{n-1} p(x_{i+1,1}|x_{i,1}) \prod_{i=1}^{n} \prod_{j=1}^{n-1} p(x_{i,j+1}|x_{i,j}) \tag{18}$$

This is a valid probability distribution insofar as it sums up to 1 and the marginals are consistent. But obviously the choice of some marginals, while abandoning the rest, retains the stain of arbitrariness. Another possibility, which regards all the given marginals, consists of combining blocks of four variables $(x_{i,j}, x_{i+1,j}, x_{i,j+1}, x_{i+1,j+1})$. The complete distribution can then be built up by:

$$\begin{aligned} q(\mathbf{x}) = p(x_{1,1}, x_{2,1}, x_{1,2}, x_{2,2}) & \prod_{i=2}^{n-1} p(x_{i+1,1}, x_{i+1,2}|x_{i,1}, x_{i,2}) \\ & \prod_{j=2}^{n-1} p(x_{1,j+1}, x_{2,j+1}|x_{1,j}, x_{2,j}) \prod_{i=2}^{n-1} \prod_{j=2}^{n-1} p(x_{i+1,j+1}|x_{i,j}, x_{i+1,j}, x_{i,j+1}) \end{aligned} \tag{19}$$

However, the factorization (19) violates the running intersection property (12). It reproduces the given marginals only in the first tetra-variate row and column, but not in the areas where the running intersection property is violated. For simplicity we assume a $3 \times 3$ grid. For abbreviation, we enumerate the variables with 1 through 9 (figure 3). We can calculate the marginal distribution $q(x_6, x_9)$ as follows:

$$\begin{aligned} q(x_6, x_9) &= \sum_{x_5, x_8} p(x_9|x_5, x_6, x_8) \sum_{x_4} p(x_8|x_4, x_5) \sum_{x_2} p(x_6|x_2, x_5) p(x_2, x_4, x_5) \\ &\approx \sum_{x_5, x_8} p(x_9|x_5, x_6, x_8) \sum_{x_4} p(x_8|x_4, x_5) p(x_4, x_5, x_6) \qquad (20) \\ &\approx \sum_{x_5, x_8} p(x_9|x_5, x_6, x_8) p(x_5, x_6, x_8) \qquad (21) \\ &= p(x_6, x_9) \end{aligned}$$

The approximations (20) and (21) would be correct if

$$\begin{aligned} p(x_6|x_2, x_5) &= p(x_6|x_2, x_4, x_5) \\ p(x_8|x_4, x_5) &= p(x_8|x_4, x_5, x_6) \end{aligned}$$

However, these equations cannot be deduced from the graphical model. The first approximation, for instance, is only true if $x_6$ and $x_4$ are conditionally independent given $(x_2, x_5)$. But this is not the case. There exists a path between $x_4$ and $x_6$ which does not have $x_2$ or $x_5$ as a node.
The optimal decomposition of a grid has already been investigated for non-serial dynamic programming by Martelli and Montari (1972). We next describe our factorized distribution algorithm *FDA* which tries to compute efficient approximate factorizations.

### 2.4 The Factorized Distribution Algorithm *FDA*

If the factorization violates the assumption of the factorization theorem, then non-serial dynamic programming does not work. But an algorithm which estimates the marginals from samples might still find the optimum. One only has to compute a good approximate factorization given the graph $G_{\text{ADF}}$. We first describe our *FDA*.

**Algorithm 2** *FDA* – Factorized Distribution Algorithm

1. Calculate $b_i$ and $c_i$ by the Sub-function Merger Algorithm.
2. $t \Leftarrow 1$. Generate an initial population with $N$ individuals from the uniform distribution.
3. **do** {
4. Select $M \leq N$ individuals using Boltzmann selection[a] (see Def. 2).
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)

[a] The algorithm works with any selection method

We next describe the sub-function merger algorithm which computes the *FDA* factorization. Let us first discuss the assumption $b_i \neq \emptyset$ of the factorization theorem. This assumption is violated already for the loop

$$s_1 = \{1, 2\}, s_2 = \{2, 3\}, s_3 = \{1, 3\}$$

All possible sequences end in $b_3 = \emptyset$ because the variables of the sub-function left are already contained in the two previous sets. One possibility to solve this problem is to choose only a subset of the $s_i$ and disregard the others; in our example, we can use the factorization $q(\mathbf{x}) = p(x_1, x_2)p(x_3|x_2)$ using $s_1$ and $s_2$. An exact factorization is $p(\mathbf{x}) = p(x_1, x_2)p(x_3|x_2, x_1)$. This factorization will be generated if the two sub-functions $s_2$ and $s_3$ are merged. This observation leads to the idea of computing approximate factorizations by merging sub-functions[5].

A good merging heuristic tries to minimize the number of mergers but simultaneously tries to use all dependencies in $G_{\text{ADF}}$. Thus the heuristic generates graphs with $b_i \neq \emptyset$ which violate the RIP only in a few regions.

[5] Bertelé and Brioschi (1972) have called it fusion.

**Algorithm 3** Sub-function Merger

```
1  S ⇐ {s_1, ..., s_m}
2  j ⇐ 1
3  while d̃_j ≠ {1, ..., n} do {
4      Chose an s_i ∈ S to be added
5      S ⇐ S \ {s_i}
6      Let the indices of the new variables in s_i be b_i = {k_1, ..., k_l}
7      for λ = 1 to l do {
8          δ_λ ⇐ {k ∈ d̃_{j-1} | (x_k, x_{k_λ}) ∈ G_ADF}
9      }
10     for λ = 1 to l do {
11         if exists λ' ≠ λ with δ_λ ⊆ δ_λ' and k_λ' not marked superfluous
12             δ_λ' ⇐ δ_λ' ∪ {k_λ}
13             Mark k_λ superfluous
14     }
15     for λ = 1 to l do {
16         if not k_λ superfluous
17             s̃_j ⇐ δ_λ ∪ {k_1, ..., k_λ}
18             j ⇐ j + 1
19     }
20 }
```

Algorithm 3 describes our heuristic. The idea of the sub-function merger algorithm is that 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. The algorithm calculates $\tilde{c}_j$, $\tilde{b}_j$ and $\tilde{d}_j$ analogous to (5).

This sub-function merger algorithm might still compute cliques that are too large. Therefore a cut parameter $k$ is needed which bounds the clique size. If the size of a clique becomes larger than $k$ our implementation will randomly leave out arcs from $G_{ADF}$.

Our presentation of the sub-function merger algorithm has been very short. The interested reader is referred to Bertelé and Brioschi (1972) for an in depth discussion of different fusion and folding heuristics. In the area of Bayesian networks, the problem has been investigated by Almond (1995).

If the conditions of the factorization theorem are fulfilled, the convergence proof of *BEDA* is valid for *FDA*, too. Since *FDA* uses finite samples of points to estimate the conditional probabilities, convergence to the optimum will depend on the size of the sample. For small sample sizes the convergence rate is higher if a number of steps with low selection is used instead of just one step using strong selection. Thus this method is numerically more efficient than using a very large sample size and strong selection.

Table 1 gives some numerical results. For $n = 30$ the probability to generate the maximum increases from about $10^{-7}$ to $10^{-1}$, and from $10^{-16}$ to $10^{-2}$ for $n = 60$. Note that in the first generations the maximum of the estimated probability is not achieved by the maximum of the function.

Table 1: Runs of *FDA* with truncation selection ($\tau = 0.3$) on a separable function (deceptive-3). For each generation, the probabilities of the most probable configuration and of the optimum are shown.

| | $n = 30$, $N = 50$ | |
|---|---|---|
| Gen | Max prob | Opt prob |
| 1 | $5.6 \cdot 10^{-6}$ | $1.4 \cdot 10^{-7}$ |
| 2 | $4.0 \cdot 10^{-4}$ | $3.5 \cdot 10^{-6}$ |
| 3 | $2.8 \cdot 10^{-3}$ | $7.7 \cdot 10^{-5}$ |
| 4 | $5.9 \cdot 10^{-2}$ | $4.7 \cdot 10^{-4}$ |
| 5 | $9.7 \cdot 10^{-2}$ | $1.9 \cdot 10^{-2}$ |
| 6 | $1.8 \cdot 10^{-1}$ | $1.2 \cdot 10^{-1}$ |
| 7 | $4.0 \cdot 10^{-1}$ | $4.0 \cdot 10^{-1}$ |

| | $n = 60$, $N = 70$ | |
|---|---|---|
| Gen | Max prob | Opt prob |
| 1 | $1.2 \cdot 10^{-12}$ | $1.6 \cdot 10^{-16}$ |
| 2 | $3.5 \cdot 10^{-10}$ | $9.5 \cdot 10^{-13}$ |
| 3 | $3.0 \cdot 10^{-8}$ | $2.7 \cdot 10^{-10}$ |
| 4 | $2.0 \cdot 10^{-6}$ | $1.2 \cdot 10^{-7}$ |
| 5 | $9.3 \cdot 10^{-5}$ | $4.1 \cdot 10^{-5}$ |
| 6 | $1.2 \cdot 10^{-3}$ | $9.5 \cdot 10^{-4}$ |
| 7 | $1.1 \cdot 10^{-2}$ | $1.1 \cdot 10^{-2}$ |

*FDA* has experimentally proven to be very successful on a number of functions where standard genetic algorithms fail to find the global optimum. In (Mühlenbein and Mahnig, 1999) the scaling behavior for various test functions has been studied. For recent surveys the reader is referred to (Mühlenbein and Mahnig, 2002a, 2003).

## 3 The Maximum Entropy Principle

*FDA* uses marginals for a factorization $q(\mathbf{x})$ which estimates the unknown distribution. This problem can be formulated more generally.

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

If only a small number of marginals is given the problem is under-specified. Consequently, for incomplete specifications the missing information must be added by some automatic completion procedure. This is achieved by the *maximum entropy principle*. Let us recall

**Definition 12.** *The entropy (Cover and Thomas, 1989) of a distribution is defined by*

$$H(p) = -\sum_x p(\mathbf{x}) \ln(p(\mathbf{x})) \tag{22}$$

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

The maximum entropy principle formulates the *principle of indifference.* If no constraints are specified, the uniform random distribution is assumed. MaxEnt has a long history in physics and probabilistic logic. The interested reader is referred to (Jaynes, 1957, 1978). MaxEnt is especially attractive because it offers a constructive way to obtain the solution. The following important theorem holds:

**Theorem 13.** *If the given marginals are consistent then there exists a unique distribution $q(\mathbf{x})$ of maximum entropy which satisfies the marginals. The distribution can be obtained by* Iterative Proportional Fitting *(IPF).*

IPF iteratively computes a distribution $q_\tau(\mathbf{x})$ from the given marginals $p_k(\mathbf{x}_k)$, $k = 1, \ldots, K$, where $\mathbf{x}_k$ is a sub-vector of $\mathbf{x}$ and $\tau = 0, 1, 2, \ldots$ is the iteration index. Let $n$

be the dimension of $\mathbf{x}$ and $d_k$ be the dimension of $\mathbf{x}_k$. $q_{\tau=0}$ is the uniform distribution. The update formula is

$$\forall \mathbf{x} \quad q_{\tau+1}(\mathbf{x}) = q_\tau(\mathbf{x}) \frac{p_k(\mathbf{x}_k)}{\sum_{y \in \{0,1\}^{n-d_k}} q_\tau(\mathbf{x}_k, \mathbf{y})} \tag{23}$$

with $k = ((\tau - 1) \mod K) + 1$.

The proof that IPF converges to the maximum entropy solution was first tried by Kullback (1968), but was faulty. The correct proof in a most general measure-theoretic framework was given in (Csiszár, 1975). Since the distribution $q$, which has to be stored and updated in every time step, has exponential size, this implementation takes exponential time and space.

We next connect the solution given by the factorization theorem with the MaxEnt solution.

**Theorem 14.** *Let consistent marginal distributions $p(\mathbf{x}_{s_i})$ be given. Let the assumptions of the factorization theorem be fulfilled. Then the factorization*

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

*is the MaxEnt solution.*

*Proof.* Use IPF to compute the MaxEnt solution. Apply the junction tree algorithm. Using these cliques one can show that IPF converges in just one sweep. □

**Remark:** This theorem has important implications. It shows that all factorizations computed by the junction tree algorithm generate the same distribution, namely the unique MaxEnt solution. We have discussed a specific version of the MaxEnt principle, namely a consistent set of marginals is given as constraints. The original MaxEnt has been introduced with other classes of constraints, mainly with moments of given functions. If the averages of the sub-functions are taken as the constraints then the MaxEnt solution is an exponential distribution, in our case the Boltzmann distribution of the ADF (Jaynes, 1978; Cover and Thomas, 1989)!

For many problems IPF cannot be performed in polynomial time. But IPF can be easily used locally to estimate higher order marginals from lower marginals computed from data. Thus it might be advantageous to compute only marginals of low order from the data, but use a factorization containing higher order marginals. The higher order marginals can be computed by *IPF*. A confirmation of this result can be found in (Ochoa et al., 2003). In this paper the structure is learned from the data. The structure is restricted to singly connected poly-trees. The poly-tree is constructed by bi-variates only. For sampling the junction tree is used. The higher order marginals are computed from the bi-variates using IPF. This algorithm performs far better than computing the higher order marginals directly from the data.

Optimization problems which have a polynomially bounded factorization fulfilling *RIP* can be solved in polynomial time. But 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. Our results obtained so far can be formulated in a conjecture.

**Conjecture:** *In the class of ADF's with non-polynomially bounded factorization 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.*

**Example:** Functions with non-polynomially bounded factorization

$$\begin{aligned} f(\mathbf{x}) &= \prod_{i=1}^{n} x_i \\ f(\mathbf{x}) &= \prod_{i=1}^{n} x_i + \sum_{i=1}^{n} x_i \end{aligned}$$

Both problems do not admit a polynomially bounded exact factorization. But whereas the first problem can only be solved in exponential time, the second problem can be solved by a simple univariate approximation:

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

**Summary:** *For all sets of consistent marginal distributions, there exists a unique MaxEnt distribution. If the graph corresponding to the marginal distributions admits a junction tree with polynomially bounded clique sizes, then the MaxEnt solution can be computed in polynomial time. If the graph does not admit a bounded junction tree, then the known algorithm (IPF) to compute the MaxEnt solution is exponential in the number of vertices. Nevertheless, FDA with an approximate factorizations might still converge to the optimum. An exact factorization is sufficient for convergence, not necessary.*

In the previous sections we have described how *FDA* computes an approximate factorization. Before we describe *LFDA*, an algorithm which computes an approximate factorization from an empirical distribution derived from a sample of function values of high fitness, we discuss the methods pursued in statistical physics.

## 4 Computing Approximate Factorizations in Statistical Physics

We discuss the method using an important example, the *2-D Ising model*. Each cell of the grid, called spin $s$, is in one of two states, $+1$ or $-1$. The cell is influenced by the four neighbors only. It is important to note (at least for a computer scientist) that Ising did *not specify any dynamics.* Instead Ising assumed that the system behaves according to a stationary distribution which is given by the *Boltzmann* distribution

$$p_\beta(\mathbf{s}) = \frac{1}{\tilde{Z}_\beta} e^{\beta \sum_{(ij)} \tilde{J}_{ij} s_i s_j + \sum_i \tilde{J}_i s_i}$$

$\tilde{J}_{ij}$ are the coupling constants. $(i, j)$ denotes a neighboring pair. For all non-neighbors, we can set $\tilde{J}_{ij} = 0$. In particular, without loss of generality, we set $\tilde{J}_{ii} = 0$, because this adds only a constant, which cancels out with $\tilde{Z}_\beta$.

Thus we again encounter the problem of *approximating a Boltzmann distribution*.

Using $s_i = 2x_i - 1$ we can change the variables to $x_i \in \{0, 1\}$. We obtain

$$p_\beta(\mathbf{x}) = \frac{1}{Z_\beta} e^{\beta \sum_{(ij)} J_{ij} x_i x_j + \sum_i J_i x_i} \tag{24}$$

$$J_{ij} = 4\tilde{J}_{ij} \tag{25}$$

$$J_i = 2\tilde{J}_i - 2 \sum_j \left( \tilde{J}_{ij} + \tilde{J}_{ji} \right) \tag{26}$$

and a different partition function $Z_\beta$.

In statistical physics the maximum entropy principle is replaced by an extension of it. Instead of minimizing the distance to the uniform random distribution (this is another formulation of MaxEnt), the distance to the Boltzmann distribution is minimized. As distance measure the *Kullback-Leibler divergence* is used.

**Definition 15.** *The Kullback-Leibler divergence ($KLD$) between two distributions is defined by*

$$KLD(q||p) = \sum_{\mathbf{x}} q(\mathbf{x}) \ln \frac{q(\mathbf{x})}{p(\mathbf{x})} \tag{27}$$

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

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

**Remark:** If $p(\mathbf{x})$ is the uniform random distribution, then MinRel is identical to MaxEnt. This justifies the above definition. But from a mathematical point of view, it is also possible to minimize the complementary divergence $KLD(p||q)$ instead. Cover and Thomas (1989) (p. 18) call $KLD(p||q)$ 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 a distribution q with these marginals 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 later used for learning Bayesian networks.

In physics MinRel is used. Let us now introduce some physical terms. This is not necessary, but it will help the reader to understand the statistical physics papers better.

**Definition 16.** *The average energy U and Gibbs free energy G are defined by*

$$U(q) = -\sum_x q(\mathbf{x}) f(\mathbf{x}) \tag{28}$$

$$G(q) = U(q) - H(q) \tag{29}$$

We can set $\beta = 1$. Then we obtain

$$KLD(q||p) = U(q) - H(q) + \ln Z. \tag{30}$$

$KLD = 0$ will be achieved for $q(\mathbf{x}) = p(\mathbf{x})$. This gives the minimal value of $G(q) = -\ln Z$. This leads to the idea of minimizing $G(q)$ for a class of distributions $q$. This means we have to compute $U(q)$ and $H(q)$. The computation of $U(q)$ is easy because the following theorem holds.

**Theorem 17.** *(Mühlenbein and Mahnig, 2002a) : Let $I = \{1, \ldots n\}$. Let $\mathbf{s} \subseteq I$ be a multi-index. Then every binary function can be written as*

$$f(\mathbf{x}) = \sum_{s \subseteq I} a_s \mathbf{x}_s \tag{31}$$

*Furthermore, the average of $f$ with respect to a distribution $q(\mathbf{x})$ is given by*

$$E_q(f) = \sum_{\mathbf{x}} q(\mathbf{x}) f(\mathbf{x}) = \sum_{s \subseteq I} a_s q_s \tag{32}$$

*where $q_s$ is the marginal distribution $q_s := q(X_i = 1 | i \in s)$ . If an ADF is given we have*

$$E_q(f) = \sum_{i=1}^{m} \sum_{\mathbf{x}_{s_i}} q(\mathbf{x}_{s_i}) f(\mathbf{x}_{s_i}) \tag{33}$$

The above theorem can easily be applied to the Ising problem, and more generally to ADF's defined on grids. Thus $U(q)$ can efficiently be computed, but the computation of $H(q)$ is more difficult. We will discuss two approximations, the first one uses univariate marginals and the second one bi-variate marginals.

### 4.1 The Mean-Field Approximation

Let us assume that a product distribution is given.

$$q(\mathbf{x}) = \prod_{i=1}^{n} q(x_i) \tag{34}$$

Then we can compute its entropy

$$\begin{aligned} H(q) &= -\sum_{x} \prod_{i=1}^{n} q(x_i) \sum_{j=1}^{n} \ln q(x_j) \\ &= -\sum_{x_1} q(x_1) \ln q(x_1) - \sum_{x_2, \ldots x_n} \prod_{i=2} q(x_i) \sum_{j=2} \ln q(x_j) \\ &= -\sum_{i=1}^{n} \sum_{x_i} q(x_i) \ln q(x_i) \end{aligned}$$

We can now try to find a local minimum by setting the derivative of $KLD$ equal to zero, using the uni-variates as variables. We abbreviate $q_i = q(x_i = 1)$

**Theorem 18.** *The mean-field approximation minimizes the Kullback-Leibler divergence to the Boltzmann distribution. The local minima of the divergence are given by the nonlinear equation*

$$q_i^* = \frac{1}{1 + e^{\frac{\partial U}{\partial q_i}}} \tag{35}$$

*Proof.* From (30) we obtain

$$\frac{\partial KLD}{\partial q_i} = \ln \frac{q_i}{1-q_i} + \frac{\partial U}{\partial q_i} = 0 \tag{36}$$

The solution gives (35). □

In (36) the derivative of the average energy $U(q)$ enters. From theorem 17 we obtain for $\beta = 1$

$$U(q) = -E_q(f) = -\sum_{(ij)} J_{ij} q_i q_j - \sum_i J_i q_i \tag{37}$$

Taking the derivative gives

$$q_i^* = \frac{1}{1 + e^{-\sum_{j \neq i}(J_{ij}+J_{ji})q_j - J_i}} \tag{38}$$

This equation can be solved by iteration.

**Remark:** In the mean-field approximation the univariate marginals are considered as variables. The marginals are computed from (38). $G(q)$ is an upper bound of $-\ln Z$. In contrast, the uni-variate approximation *UMDA* (Mühlenbein and Mahnig, 2001) computes the marginals from the samples. It has to be investigated if the additional computational effort needed for the mean-field approach pays off.

### 4.2 Bethe-Kikuchi Approximation and the *FDA* Factorization

An obvious extension of the mean-field approach is the use of higher order marginals. This has been done by Bethe (1935) using bi-variate marginals and Kikuchi (1951) for higher order marginals. The interested reader is referred to Yedidia et al. (2001) for the original statistical physics approach. A state-of the art report has recently been written by the same authors (Yedidia et al., 2004). For a 1-D loop the Bethe factorization is given by

$$Q(\mathbf{x}) = \prod_{i=1}^{n} \frac{b(x_i, x_{i+1})}{b(x_i)} \tag{39}$$

$b(x_i, x_{i+1})$ are consistent bi-variate marginals to be computed by minimizing the free energy. But note that $Q(\mathbf{x})$ contains a loop and is not normalized, i.e. it does not sum to one. This makes the minimization of the free energy more than problematic[6]. The same is true for the higher order Kikuchi factorization. For EDA we face a second problem, because sampling from a factorization with loops is a difficult problem by itself. Therefore we decided to use our *FDA* factorization instead. This factorization does not contain cycles (note that $\mathbf{x}_{c_1} = \emptyset$ and therefore $q(\mathbf{x}_{c_1}) = 1$). For

$$q(\mathbf{x}) = \prod_{i=1}^{m} \frac{\tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})}{\tilde{q}(\mathbf{x}_{c_i})} \tag{40}$$

we obtain

$$H(q) = -\sum_{i=1}^{m} \sum_{\mathbf{x}_{b_i}, \mathbf{x}_{c_i}} q(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) \ln \frac{\tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})}{\tilde{q}(\mathbf{x}_{c_i})} \tag{41}$$

[6] Entropy and $KLD$ are not defined.

The proof is based on marginalization and left to the reader. But from equation (9) we know that $q(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) \neq \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})$ if the RIP is violated. Therefore we approximate

$$H(q) \approx -\sum_{i=1}^{m} \sum_{\mathbf{x}_{b_i}, \mathbf{x}_{c_i}} \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) \ln \frac{\tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})}{\tilde{q}(\mathbf{x}_{c_i})} \tag{42}$$

$U(q)$ can be computed using (33). We obtain the approximation

$$U(q) \approx -\sum_{i=1}^{m} \sum_{\mathbf{x}_{b_i}, \mathbf{x}_{c_i}} \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) f_i(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) \tag{43}$$

Having approximations of $U(q)$ and $H(q)$ we can minimize $KLD$ as before. But now we have a *constrained minimization* problem, because the marginals have to be consistent. To our knowledge there exists no computer implementation of the full Bethe-Kikuchi method. Yedidia et al. (2004) have proposed iterative algorithms using the Lagrange multipliers approach. This is technically demanding, because all intersections of $c_i$ with $b_j$ have to be considered for a consistency constraint. During the iteration the marginals have to be made consistent. Yedidia et al. (2004) have shown that their algorithm, called generalized belief propagation, converges to stationary points of the approximated $KLD$, if it converges at all. For an efficient implementation of the algorithm special intersection sets are constructed. We will test promising algorithms for our *FDA* factorization in the near future.

**Summary:** *In the Bethe-Kiguchi approach the marginals used for the factorization are not computed from a sample, but are determined by computing a stationary point of the Kullback-Leibler divergence to the Boltzmann distribution. The Bethe-Kiguchi factorization is not normalized, therefore $KLD$ is not defined in a strict sense. The FDA factorization circumvents this problem, but also for this approximation $KLD$ can be computed only approximately. Therefore the goodness of this heuristic cannot be assessed theoretically.*

We next turn to an approach which uses only samples of good fitness values, the structure of the function is unknown. This approach has first been investigated in artificial intelligence. It is called learning of the model (Jordan, 1999).

## 5 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 (Larrañaga and Lozano, 2002). We will just motivate some of the major design decisions. Let $p(\mathbf{x})$ be the true distribution. The learning algorithm uses acyclic Bayesian networks (acBN) as models.

$$q(\mathbf{x}) = \prod_{i=1}^{n} q(x_i|\pi_i) \tag{44}$$

$\Pi_i$ are called the parents of $X_i$, $q(x_i|\pi_i)$ is a numerical approximation of the true conditional marginal $p(x_i|\pi_i)$. (We recall that any *FDA* factorization can be written in this form.)

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 sample, we can choose which marginal distributions should be used in order to obtain a Bayesian network which reproduces the data accurately. This is called *model selection*.

Therefore we have to deal with the problem *how to choose the appropriate model*. This problem can be solved in the following way. Let $Q$ be the set of all possible distributions $q$ defined by the Bayesian networks considered. (Because of the efficiency the number of parents is bound usually by a small number.) We next try to find a good approximation in $Q$ by minimization of $KLD(p||q)$ We obtain with the average of $\ln q$ over the true distribution $p$

$$E_p(\ln q) = \sum_{\mathbf{x}} p(\mathbf{x}) \ln q(\mathbf{x}) = E_p(\ln q) = -H(p) - KLD(p||q) \tag{45}$$

Therefore the minimization of $KLD(p||q)$ in $Q$ is equivalent to maximization of $E_p(\ln q)$. The next proposition shows that $E_p(\ln q)$ can be computed efficiently.

**Proposition 19.** *For the distribution $q(x) = \prod_{i=1}^{n} q(x_i|\pi_i)$ we have*

$$E_p(\ln q) = \sum_{i=1}^{n} \sum_{x_i,\pi_i} p(x_i, \pi_i) \ln q(x_i|\pi_i) \tag{46}$$

*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 (46) can be approximated using a finite sample. For simplicity, we introduce the following notation. Let $N$ denote the size of the sample. Let $N_{ijk}$ denote the number of instances with $x_i = k$ and $\pi_i = j$, where the states of $\Pi_i$ are numbered $1 \leq j \leq 2^{|\Pi_i|}$. Let $N_{ij} = \sum_k N_{ijk}$. We can now approximate

$$q(x_i = k|\pi_i = j) \approx \frac{N_{ijk}}{N_{ij}} \tag{47}$$

$$p(x_i = k, \pi_i = j) = \lim_{N\to\infty} \frac{N_{ijk}}{N} \tag{48}$$

$$E_p(\ln q) = \lim_{N\to\infty} \left( E_N(\ln q) = \sum_{i=1}^{n} \sum_{j=1}^{2^{|\Pi_i|}} \sum_{k=1}^{2} \frac{N_{ijk}}{N} \ln \frac{N_{ijk}}{N_{ij}} \right) \tag{49}$$

Thus we have arrived at the following principle

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

$$max_{q\in Q} E_N(\ln q) = \sum_{i=1}^{n} \sum_{j=1}^{2^{|\Pi_i|}} \sum_{k=1}^{2} \frac{N_{ijk}}{N} \ln \frac{N_{ijk}}{N_{ij}} \tag{50}$$

FinMaxLike can also be derived from the maximum log-likelihood principle.

**Proposition 20.** *Let $l(q|D)$ be the log-likelihood of $q$ given the data $D$. Then*

$$l(q|D) \approx N \cdot E_N(\ln q) \tag{51}$$

*Proof.* Let $D = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$. Then the likelihood of $D$ is given by

$$L(q|D) = \prod_{l=1}^{N} \prod_{i=1}^{n} q(x_{li}|\pi_{li}) \tag{52}$$

Therefore we obtain

$$\begin{aligned} l(q|D) &= \ln L(q|D) = \sum_{l=1}^{N} \sum_{i=1}^{n} \ln q(x_{li}|\pi_{li}) \\ &= \sum_{i=1}^{n} \sum_{l=1}^{N} \ln q(x_{li}|\pi_{li}) \\ &\approx \sum_{i=1}^{n} \sum_{j=1}^{2^{|\Pi_i|}} \sum_{k=1}^{2} N_{ijk} \ln \frac{N_{ijk}}{N_{ij}} \end{aligned}$$

□

For given $N$ maximizing $E_N(\ln q)$ gives the same result as maximizing $l(q|D)$. But this principle is not yet sufficient for practical computations. $FinMaxLike$ does not prefer exact models of small complexity (small number of variables) to exact models of large complexity. This can be seen as follows. If we approximate $q(x_i, \pi_i) \approx p(x_i, \pi_i)$ for $N \to \infty$, then we obtain

$$E_p(\ln q) = \sum_{i=1}^{n} \left( \sum_{x_i, \pi_i} p(x_i, \pi_i) \ln \frac{p(x_i, \pi_i)}{p(x_i) p(\pi_i)} + \sum_{x_i} p(x_i) \ln p(x_i) \right) \tag{53}$$

The first term is called the *mutual information* $I(X;Y)$ (Cover and Thomas, 1989).

$$I(X;Y) = \sum_{x,y} p(x,y) \ln \frac{p(x,y)}{p(x)p(y)} \tag{54}$$

If $X$ and $Y$ are independent, we have $I(X;Y) = 0$. Using the mutual information we obtain

$$E_p(\ln q) = \sum_{i=1}^{n} (I(X_i; \Pi_i) + \sum_{x_i} p(x_i) \ln p(x_i)) \tag{55}$$

$E_p(\ln q)$ remains unchanged if an arc between two independent variables is inserted. In order to solve this over-fitting problem, we need a criterion which *maximizes the log-likelihood* and *minimizes the complexity of the model*. Such a criterion can be derived using Bayesian principles or a concept called the *minimum description length* MDL. The interested reader is referred to (Jordan, 1999). One of the most popular criterion has been derived by Schwarz (1978). It has been called the *Bayesian Information Criterion* BIC.

**Definition 21.** *Let $V$ be the degrees of freedom of the marginal distributions of $q$. Then the weighted BIC measure is defined by*

$$BIC_\alpha = N \cdot E_N(\ln q) - \alpha \ln N \cdot V$$

It has been shown that $-BIC = -N \cdot E_N(\ln q) + 0.5 \ln(N) \cdot V$ is asymptotically equivalent to the minimum description length. Schwarz (1978) computed $\alpha = 0.5$ as the best weighting factor for $N \to \infty$.

The $BIC$ criterion can be used to construct a Bayesian network. Basically there are two approaches. The first one starts with an *empty* network and adds arcs between *dependent* variables, the second one starts with the *fully connected* network and *deletes* arcs between *independent* variables. In most implementations the first approach is used together with a simple greedy heuristic for adding a single arc at each step. The reader is referred to (Larrañaga and Lozano, 2002; Mühlenbein and Mahnig, 1999). The quality of both approaches depends on a *good estimate of the mutual information*, the heuristic to *construct the network*, and the *weighting factor*.

These topics will be briefly investigated next. A detailed report will be available soon.

## 6 A Comparison of the Presented Approaches

In this section we compare the three different principles presented in this paper - MaxEnt, MinRel, and MinLike. We restrict the discussion to problems where an additive decomposed function (ADF) is given.

*FDA* uses a factorization defined by equation (13). If the factorization fulfills the running intersection property it follows from the factorization theorem that the factorization gives the true distribution. But in general, the factorization will not fulfill the *RIP*. Our *FDA* factorization algorithm tries to create a graphical model $G_{\text{FDA}}$ which does not leave out arcs of $G_{\text{ADF}}$. It can easily be transformed into an acyclic Bayesian network. This class is used by *LFDA*. Therefore *FDA* and *LFDA* use the same class of graphical models, but the computation of the graph is very different.

The approximations originally proposed by Bethe-Kikuchi contain $G_{\text{ADF}}$. But they compute only the marginals, not the whole distribution. Let us discuss the methods using a simple example.

**Example:** A 1-D circle

$$\begin{aligned} f(\mathbf{x}) &= J_{12}x_1x_2 + J_{23}x_2x_3 + J_{34}x_3x_4 + J_{14}x_1x_4 + \sum J_i x_i \\ p(\mathbf{x}) &= \frac{e^{f(\mathbf{x})}}{Z} \end{aligned}$$

$G_{\text{ADF}}$ is a loop. The following approximations can be used

1. $q(x) = \tilde{q}(x_1)\tilde{q}(x_2|x_1)\tilde{q}(x_3|x_2)\tilde{q}(x_4|x_3)$
   - It defines an acBN with *RIP*, but it does not contain $G_{\text{ADF}}$ (the arc between $x_4$ and $x_1$ is missing).

2. $q(x) = \tilde{q}(x_1)\tilde{q}(x_2|x_1)\tilde{q}(x_3|x_2)\tilde{q}(x_4|x_1, x_3)$
   - It defines an acBN *without RIP*. It contains $G_{\text{ADF}}$. This factorization is computed by *FDA* using merging of sub-functions. The last factor is a tri-variate marginal. For large samples *LFDA* computes also a graph, which contains $G_{\text{ADF}}$

3. $q(x) = \tilde{q}(x_1)\tilde{q}(x_2|x_1)\tilde{q}(x_3|x_1, x_2)\tilde{q}(x_4|x_3, x_1)$
   - It defines an acBN with *RIP* which contains $G_{\text{ADF}}$. It uses two tri-variate marginals. This graph will be obtained by the junction tree algorithm. This factorization is exact.

4. Original Bethe approach: Compute marginals $(\tilde{q}(x_i, x_{i+1}), i = 1, \ldots 4)$ by minimizing $G(q) = U(q) - H(q)$
   - The marginals define a graph with a loop. It needs an iterative method to sample from the marginals.

5. Given all bi-variate marginals of the unknown distribution $p(x_1, x_2), \ldots p(x_1, x_4)$, compute the unique $MaxEnt$ distribution $q_{\text{ME}}(\mathbf{x})$
   - We have shown that $q_{\text{ME}}(\mathbf{x}) = p(\mathbf{x})$, but the computation of $q_{\text{ME}}(\mathbf{x})$ is exponential.

From a theoretical standpoint the junction tree factorization is the best. But for 2-D grids this method leads to large clique sizes ($O$(size of the grid)). We have proposed using the *FDA* factorization. For 2-D grids the sub-function merge algorithm computes tri-variate marginals in the interior.

## 7 How to Test EDA Algorithms

In our opinion most researchers so far have concentrated on the benchmark method to show the power of EDA algorithms. A popular benchmark or a difficult function is chosen and the success of the optimization algorithm is shown. The success rate is usually the percentage of runs computing the optimum. The internal behavior of the algorithm (e.g. which Bayesian network it has constructed etc.) is not investigated. Therefore a generalization of the results to other problems is difficult.

We propose to test EDA algorithms in carefully selected steps instead – starting from theoretically understood problems to more complex ones. Learning the structure of the Bayesian network from data is a difficult task. Many factors contribute to the success or failure of the learning method. The Bayesian network community has investigated this problem intensively. We just mention (Xiang et al., 1997) as a starting point. The test procedure is obvious. A Bayesian network is used to generate the data, then a Bayesian network is learned from this data. We then measure how close the learned Bayesian network is to the network generating the data. The results show that in general large data sets are needed to learn a network which is close to the given one. We first test *LFDA* on problems where all variables are independent, i.e $G_{\text{FDA}}$ contains no arcs.

### 7.1 The Penalty Weight $\alpha$

Schwarz (1978) has computed an optimal penalty factor $\alpha = 0.5$ under very restrictive assumptions. (One of the assumptions is $N \to \infty$.) Since we want to use fairly small population sizes, we investigate the influence of $\alpha$ on the computed network in the neighborhood of $\alpha = 0.5$. In a first test we generate uniform random data. In this case the exact network has no edges at all. Table 2 shows empirical results. How can we define an optimal $\alpha$? It is obvious that no edges will be generated for a large $\alpha$. For very small $\alpha$ many edges will be generated. Thus we are looking for a value of $\alpha$ at the transition between these two regimes. Informally speaking, we look for $\alpha_{\text{tr}}$ with $\#edges \leq 5$ for $\alpha > \alpha_{\text{tr}}$, $4 \leq \#edges \leq 10$ for $\alpha = \alpha_{\text{tr}}$ and $\#edges > 10$ for $\alpha < \alpha_{\text{tr}}$.

The results of table 2 suggest that a value of $\alpha_{\text{tr}} = 0.75$ fulfills the requirements for small population sizes. The last row shows the results for a very large population. In this case $\alpha_{\text{tr}} = 0.5$ might be indeed the best value.

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

| $\alpha$ | $n$ | $N$ | $\#edges$ | $n$ | $N$ | $\#edges$ | $n$ | $N$ | $\#edges$ |
|---|---|---|---|---|---|---|---|---|---|
| 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 |

We have also analyzed the weighting factor in optimization tasks. In many applications the success of *LFDA* can dramatically be increased by a suitable weighting factor. But in general $\alpha = 0.5$ with truncation selection with $\tau = 0.2$ and is a good starting point.

### 7.2 The Connection between *FDA* and *LFDA*

At first it seems that an internal testing of the learned network of *LFDA* is impossible, because the structure of the true network seems to be unknown. But the theory helps. *FDA* can be seen as the infinite sample size limit of any plausible learning method, e.g., *LFDA*. The relation between *FDA* and *LFDA* is difficult to formulate precisely. The following conjecture is a first try.

**Conjecture:** *Let the empirical distribution $q(\mathbf{x})$ be generated by selected points of high fitness from an ADF . Then for $N \to \infty$ the mutual information is the largest between those variables which are contained in a common sub-function. This means that the graph obtained by connecting the variables with highest mutual information contains $G_{ADF}$.*

Thus for $N \to \infty$ the learning algorithm *LFDA* has to solve the same problem as *FDA*: Given the graph $G_{\mathrm{ADF}}$ compute an acyclic Bayesian network. *LFDA* has an advantage, because it can use the mutual information to leave out less important arcs. For some practical problems with a sparsely connected $G_{\mathrm{ADF}}$ the graph $G_{\mathrm{FDA}}$ contains $G_{\mathrm{ADF}}$. Thus it does not leave out arcs. The same is true for 2-D grids and *LFDA*.

**Observation:** *If the ADF is defined on a 2-D grid, then $G_{LFDA}$ for $N \to \infty$ contains $G_{ADF}$.*

The above observation depends on the specific learning algorithm. *LFDA*, for instance, uses a greedy heuristic which adds a single arc at each step. This method has limitations in constructing the correct network for some artificial distributions, as is shown in (Xiang et al., 1997).

We will investigate our conjectures using three typical benchmark functions. The first function is separable of order 5 and deceptive.

$$F_{\mathrm{Dec5}}(x) = \sum_{i=1}^{m} f_{\mathrm{dec5}}(x_{5i-4}, \ldots, x_{5i}) \qquad (l = 1, 2, 3)$$

Table 3: The minimal population size to find the optimum with $95\%$.

| Alg | $n$ | $F_{\text{Dec5}}$ | $F_{\text{IsoPeak}}$ | Alg | $F_{\text{Dec5}}$ | $F_{\text{IsoPeak}}$ |
|---|---|---|---|---|---|---|
| *FDA* | 25 | 400 | 250 | *LFDA* | *3500 | 900 |
| *FDA* | 50 | 600 | 700 | *LFDA* | *18000 | *5000 |
| *FDA* | 100 | 800 | 500 | | | |
| *FDA* | 200 | 1200 | †3500 | | | |

with

$$f_{\text{dec5}}(x_1, x_2, x_3, x_4, x_5) = \begin{cases} 0.9 & \iff \sum x_i = 0 \\ 0.8 & \iff \sum x_i = 1 \\ 0.7 & \iff \sum x_i = 2 \\ 0.6 & \iff \sum x_i = 3 \\ 0.0 & \iff \sum x_i = 4 \\ 1.0 & \iff \sum x_i = 5 \end{cases} \tag{56}$$

The second function is non-separable. It consists of $m$ overlapping blocks of size 3 ($n = 2m + 1$). We have a different function for the last block.

$$F_{\text{IsoPeak}}(\mathbf{x}) := \sum_{i=1}^{m-1} f(x_{2i-1}, x_{2i}, x_{2i+1}) + g(x_{2m-1}, x_{2m}, x_{2m+1}) \tag{57}$$

$$f(x, y, z) = \begin{cases} m & \iff (x, y, z) = (0, 0, 0) \\ m - 1 & \iff (x, y, z) = (1, 1, 1) \\ 0 & \text{otherwise} \end{cases} \tag{58}$$

$$g(x, y, z) = \begin{cases} m & \iff (x, y, z) = (1, 1, 1) \\ 0 & \text{otherwise} \end{cases} \tag{59}$$

Both functions admit an exact factorization. The third class of functions ($F_{\text{3-SAT}}$) are 3-SAT problems with 20 and 50 variables.

Table 3 gives the population size for which the optimum is found with 95% in 20 runs. The selection threshold for truncation selection was set to $\tau = 0.3$, except * ($\tau = 0.1$), and † ($\tau = 0.7$). For missing values, the required population size is too large.

We first compare the performance of *FDA* and *LFDA*. The population size needed for finding the optimum is much larger for *LFDA* than for *FDA*. The results of *LFDA* for the 'easy' separable function $F_{\text{Dec5}}$ is especially disappointing. The 3-SAT problem with 20 variables problem was very easy to solve. Both *FDA* and *LFDA* need a population size of 250. But for $n = 50$ the 3-SAT problems turn out to be very difficult to solve, both for *FDA* and *LFDA*. The reason is that the graphs $G_{\text{ADF}}$ of 3-SAT problems are irregularly connected. For these graphs the computing of good factorizations with bounded clique size is very difficult or even impossible.
The factorization of the separable function $F_{\text{Dec5}}$ is obvious. For $F_{\text{IsoPeak}}$ *FDA* computes the exact factorization

$$q(\mathbf{x}) = \hat{p}(x_1, x_2, x_3)\hat{p}(x_4, x_5|x_3) \cdots \hat{p}(x_{n-1}, x_n|x_{n-2}) \tag{60}$$

Table 4: Characterization of the Bayesian network computed by *LFDA* in the first generations. In column **arcs comp.** the number in brackets denotes the total number of arcs, the first number gives the number of arcs contained in $G_{\text{ADF}}$. **Mutual information** gives the number of arcs in $G_{\text{ADF}}$ which are contained in the set of arcs having largest mutual information (number in brackets). $*$ denotes that the *LFDA* run did not find the optimum. Maximum number of parents $Q = 8$. $F_{\text{3-SAT}}$ with $n = 50$ was run with very strong selection ($\tau = 0.03$).

| $F$ | n | arcs | N | arcs comp. | mutual information |
|---|---|---|---|---|---|
| $F_{\text{Dec5}}$ | 50 | 100 | *1000 | 3(28),5(38),4(34) | 3(50),7(50),6(50) |
| | | | *5000 | 2(12),2(20),3(23) | 3(50),6(50),6(50) |
| | | | *10000 | 0(2),4(14),6(20) | 4(50),5(50),12(50) |
| | | | *30000 | 28(68),63(95),79(105) | 20(50),37(50),68(70) |
| $F_{\text{IsoPeak}}$ | 24 | 36 | 500 | 16(23),19(23),22(28) | 21(28), 23(28),24(28) |
| | | | 5000 | 34(39),36(39) | 36(40), 36(40) |
| | 48 | 72 | *2000 | 44(47),54(59),. . . 13(18) | 60(70),62(70),. . . 35(70) |
| | | | 5000 | 58(63),66(73),69(74) | 69(80),73(80),75(80) |
| | 74 | 112 | 5000 | 77(85),90(97),98(105) | 103(115),109(115),112(115) |
| $F_{\text{3-SAT}}$ | 20 | 141 | 100 | 55(65),30(47) | 90(120),90(120) |
| | | | 500 | 63(81),65(82),57(70) | 90(120),94(120),95(120) |
| | | | 5000 | 101(108) | 115(120) |
| $\alpha = 0.25$ | 50 | 491 | *30000 | 131(138),131(141),114(135) | 175(200),158(200),135(200) |
| $\alpha = 0.10$ | | | 30000 | 183(288),195(254),180(225) | 180(200),184(200),158(200) |

In table 4 we investigate the structure learned by *LFDA*. Here we see the reason why *LFDA* needs a huge population size to find the optimum of $F_{\text{Dec5}}$. If the population is small, the network computed by *LFDA* contains only a few arcs. A closer look shows that this result is not a problem of the learning procedure, but of the computation of the mutual information. Only for $N = 30000$ a reasonable number of the correct dependencies are contained in the set of the variables with largest mutual information. The reason can easily be given. If selection is done, then it seems that the sub-function is almost linear.

The results for $F_{\text{IsoPeak}}$ are very good. The learned network is almost identical to the network $G_{\text{FDA}}$ computed by the *FDA* sub-function merger algorithm. The performance for $F_{\text{3-SAT}}$ with 20 variables is surprisingly good, despite that $G_{\text{LFDA}}$ contains only $1/3$ of the arcs of $G_{\text{ADF}}$. But the performance changes dramatically for $F_{\text{3-SAT}}$ with 50 variables. The optimum is found only with very strong selection ($\tau = 0.03$), a large population size ($N = 30000$), and a small weight factor ($\alpha = 0.1$). A look into table 4 shows the reason for the bad performance. *LFDA* computes a network which leaves out too many arcs of $G_{\text{ADF}}$, despite that the mutual information gives correct information about the dependencies. Even for $\alpha = 0.1$ the learned network $G_{\text{LFDA}}$ contains only about 180 edges from the 491 edges in $G_{\text{ADF}}$. The same problem occurs with *FDA*.

In figure 4 we analyze the dynamic behavior of our greedy algorithm computing the network. The search starts with complexity of almost 0 and a small log-likelihood. The log-likelihood is increased until the increase is equal to the increase in complexity.

**Summary:** *The learning algorithms face two problems, first to identify the dependent variables, and second to compute an acBN which has a bound on the number of parents, but does not leave out important dependencies. In order to get good optimization results, a large population size has to be used.*

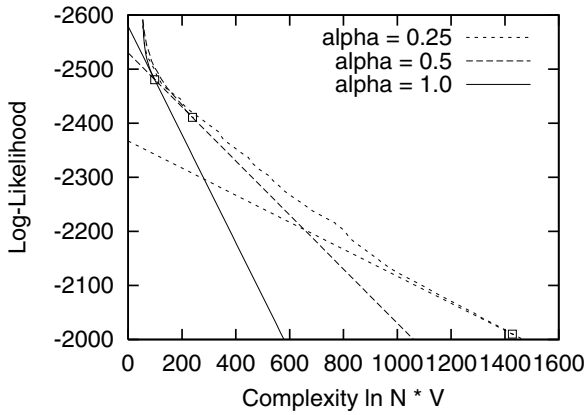


Figure 4: Trajectories of three runs of the *LFDA* learning algorithm in the $BIC$ space. Maximum number of parents $Q = 8$, $N = 600$, $\alpha$ varies. The squares mark the endpoints of the runs. The tangents with gradient $\alpha$ are also shown.

In this paper we have only discussed a learning algorithm which starts with an empty network. In the near future we will investigate learning algorithms which start with the fully connected network and *delete* the arcs between *independent* variables. It seems that the detection of independent variables is more reliable for small population sizes than the test for dependence. Unfortunately the construction of the Bayesian network is much more complicated. Just deleting arcs might lead to a very difficult graphical model, which would not be a Bayesian network at all.

## 8 Conclusion and Outlook

The efficient estimation and sampling of distributions is a common problem in several scientific disciplines. Unfortunately each discipline uses a different language to formulate its algorithms. We have identified two principles suited for the approximation – *minimum relative entropy* and *minimum expected log-likelihood ratio*. Both principles are closely related. If $p$ is the distribution to be estimated, then MinRel minimizes the Kullback-Leibler divergence $KLD(q||p)$ whereas MinLike minimizes $KLD(p||q)$.

We have shown that the basic theory is the same for all algorithms. This theory deals with the decomposition of graphical models and the computation of approximate factorizations. If the unknown distribution allows an exact factorization, then both methods lead to $KLD = 0$, thus they compute the exact distribution.

We have discussed two EDA algorithms in detail. *FDA* computes a factorization from the graph representing the structure. If the corresponding graphical model does not fulfill the assumptions of the factorization theorem the exact distribution is only approximated. Promising factorizations can be obtained by merging some sub-functions.

*LFDA* learns the structure from the data, however it faces serious problems. A crucial point for its success seems to be the correct detection of the important dependencies. Statistical physics uses the most complex approach; selected marginal distributions are computed by minimizing the distance to the Boltzmann distribution. The marginals generate a dependence graph with loops. Therefore they are not sufficient to define a distribution, any factorization $q(\mathbf{x})$ using these marginals has to be normalized

($\sum_{\mathbf{x}} q(\mathbf{x}) = 1$). But this summation is exponential. We have proposed an extension of the original approach which circumvents the normalization by using the marginals of a *FDA* factorization.

One important improvement of *FDA* and *LFDA* could not be discussed in this paper: the *use of a local hill-climber*. This topic is discussed for large bi-partitioning problems in Mühlenbein and Mahnig (2002a). We have now implemented a local hill-climber proposed by Lin and Kernighan (1973), which can be used for many combinatorial optimization problems. It increases the performance of *FDA* and *LFDA* on 3-SAT problems substantially. In fact, problems up to size 250 pose no difficulties. The reason is that local optima have a structure which can be learned more easily by *LFDA*.

The goal of this paper is to inspire researchers to implement some of the methods presented and make a detailed comparison between the different methods. Especially interesting would be an implementation of the full Kikuchi method, which has been implemented only for special *ADF*'s so far. Whereas the theory of EDA algorithms is very convincing, we all have to work hard to design numerically efficient EDA algorithms. Efficiency can only be achieved by having a close look at the different developments in the neighboring disciplines mentioned.

## References

Almond, R. G. (1995). *Graphical Belief Modelling*. Chapman & Hall, London.

Bertelé, U. and Brioschi, F. (1972). *Nonserial Dynamic Programming*. Academic Press, New York.

Bethe, H. (1935). Statistical theory of superlattices. *Proc. Roy. Soc. London A*, 150:552–558.

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

Csiszár, I. (1975). I-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, 3:146–158.

Huang, C. and Darwiche, A. (1996). Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263.

Jaynes, E. T. (1957). Information theory and statistical mechanics. *Phys. Rev*, 6:620–643.

Jaynes, E. T. (1978). Where do we stand on maximum entropy? In Levine, R. D. and Tribus, M., editors, *The Maximum Entropy Formalism*. MIT Press, Cambridge.

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. (1999). *Learning in Graphical Models*. MIT Press, Cambrigde.

Kikuchi, R. (1951). A theory of cooperative phenomena. *Phys.Review*, 115:988–1003.

Kullback, S. (1968). Probability densities with given marginals. *Annals of Mathematical Statistics*, 39(4):1236–1243.

Larrañaga, P. and Lozano, J. (2002). *Estimation of Distribution Algorithms: A New Tool for Evolutionary Optimization*. Kluwer Academic Press, Boston.

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

Lin, S. and Kernighan, B. (1973). An effective heuristic algorithm for the traveling-salesman. *Operations Research*, 21:498–516.

Martelli, A. and Montari, U. (1972). Nonserial dynamic programming: On the optimal strategy of variable elimination for the rectangular lattice. *Jour. Math. Analyis and Applications*, 40:226–242.

Mühlenbein, H. and Mahnig, T. (1999). FDA – a scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376.

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

Mühlenbein, H. and Mahnig, T. (2001). Evolutionary computation and beyond. In Uesaka, Y., Kanerva, P., and Asoh, H., editors, *Foundations of Real-World Intelligence*, pages 123–188. CSLI Publications, Stanford, California.

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 Jong, K. D., 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, Springer Verlag, Berlin.

Ochoa, A., Höns, R., Soto, M., and Mühlenbein, H. (2003). A maximum entropy approach to sampling in EDA - the singly connected case. In *Proceedings of the 8th Iberoamerican Congress on Pattern Recognition (CIARP 2003)*, volume 2905 of *Lecture Notes in Computer Science*, pages 683–690. Springer Verlag, Berlin.

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

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

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2001). Understanding belief propagation and its generalizations. Technical Report 2001-22, Mitsubishi Electric Research Laboratories.

Yedidia, J. S., Freeman, W. T., and Weiss, Y. (2004). Constructing free energy approximations and generalized belief propagation algorithms. Technical Report 2004-40, Mitsubishi Electric Research Laboratories.