The Factorized Distribution Algorithm and the Minimum Relative Entropy Principle

Heinz Mühlenbein and Robin Höns

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

Summary. Estimation of Distribution Algorithms (EDA) have been proposed as an extension of genetic algorithms. In this paper the major design issues of EDA's are discussed using an interdisciplinary framework, the *minimum relative entropy* ($MinRel$) approximation. We assume that the function to be optimized is additively decomposed (*ADF*). The interaction graph G_{ADF} of the *ADF* is used to create exact or approximate factorizations of the Boltzmann distribution. The relation between the Factorized Distribution Algorithm *FDA* and the $MinRel$ approximation is shown. We present a new algorithm, derived from the *Bethe-Kikuchi* approach developed in statistical physics. It minimizes the relative entropy $KLD(q|p_\beta)$ to the Boltzmann distribution p_β by solving a difficult constrained optimization problem. We present in detail the concave-convex minimization algorithm *CCCP* to solve the optimization problem. The two algorithms are compared using popular benchmark problems (2-d grid problems, 2-d Ising spin glasses, Kaufman's $n-k$ function.) We use instances up to 900 variables.

Key words: Estimation of distributions, Boltzmann distribution, factorization of distributions, maximum entropy principle, minimum relative entropy, maximum log-likelihood, Bethe-Kikuchi approximation.

1 Introduction

The *Estimation of Distribution* (EDA) family of population based search algorithms was introduced in (21) as an 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 do not exploit the correlation of the variables in samples of high fitness values.

[1] In (21) they have been named *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.

The family of EDA algorithms can be understood and further developed without the background of genetic algorithms. The problem to estimate empirical distributions has been investigated independently in several scientific disciplines. 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 and deals with slightly different problems. In EDA we want to generate points with a high probability $p(\mathbf{x})$, 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 sample $p(\mathbf{x})$ in areas of high fitness values, i.e. we are interested in a sampling method which generates points with a high value of $p(\mathbf{x})$. But all disciplines need 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.

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 generated. The second branch has been derived from the theory of belief networks (11). It computes a factorization from samples instead from the analytical expression of the fitness function. The underlying theory is the same for both branches. For large real life applications often a hybrid between these two approaches is most successful (17). In this paper we investigate the first branch only.

We discuss the problem of computing approximations of distributions using factorizations is investigated with the framework of *minimum relative entropy*. We distinguish exact factorizations and approximate factorizations. We shortly summarize the results for our well-known algorithm *FDA*. We present in detail a new algorithm *BKDA*. It is derived from an approach used in statistical physics to approximate the Boltzmann distribution. It is called the *Bethe-Kikuchi* approximation. In this approach the marginals from the unknown Boltzmann distribution are not computed from data, but from a difficult constrained optimization problem. This paper extends the theory first described in (15).

In the last section we summarize the functionality of our software system *FDA*. It can be downloaded from *http://www.ais.fraunhofer.de/∼muehlen*.

The different EDA algorithms are shortly numerically compared, using large benchmark optimization problems like 2-D Ising spin glasses, and Kaufman's $n-k$ function. We investigate problems with up to 900 variables, continuing the work in (17), where graph bi-partitioning problems of 1000 nodes have been solved.

2 Factorization of the Search Distribution

EDA has been derived from a search distribution point of view. We just recapitulate the major steps published in (20; 17). 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.
Let a function $f : \mathbf{X} \to \mathbb{R}_{\geq 0}$ be given. We consider the optimization problem

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

A promising search distribution for optimization 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 β and/or f might be omitted.

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 (see (18)). 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 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*

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

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

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.

Algorithm 1: *BEDA* – Boltzmann Estimated Distribution

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)

BEDA is a conceptional algorithm, because the calculation of the distribution $p^s(\mathbf{x}, t)$ 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

From now on we assume that the fitness function is additively decomposed.

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 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). The ADF is k-bounded if $\max_i |s_i| \leq k$.*

Definition 5 *Let an ADF be given. Then the interaction graph G_{ADF} is defined as follows: The vertices of G_{ADF} 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 (equation (2)) using a product of marginal distributions. The approximation has to fulfill two conditions

- The approximation should only use marginals of low order.
- Sampling from the approximation should be easy.

A class of distributions fulfilling these conditions are the acyclic Bayesian network (acBN).

$$q(\mathbf{x}) = \prod_{i=1}^{n} q(x_i|\pi_i) \tag{5}$$

where π_i are called the parents of x_i. For acyclic Bayesian networks sampling can easily be done starting with the root x_1. Cyclic Bayesian networks are difficult to sample from.

Note that any distribution can be written in the form of an acyclic Bayesian network because of

$$p(\mathbf{x}) = p(x_1)p(x_2|x_1)p(x_3|x_1,x_2)\cdots p(x_n|x_1,\ldots,x_{n-1}) \tag{6}$$

But this factorization uses marginal 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 independent of n.

For *ADF*'s the following procedure can be used to create factorizations. 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:*

$$d_i := \bigcup_{j=1}^{i} s_j, \qquad b_i := s_i \setminus d_{i-1}, \qquad c_i := s_i \cap d_{i-1} \tag{7}$$

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 *(13).*

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.*

$$\sum_{\mathbf{x}_{b_i},\mathbf{x}_{c_i}} \tilde{q}(\mathbf{x}_{b_i},\mathbf{x}_{c_i}) = 1 \tag{8}$$

$$\sum_{\mathbf{x}_{b_i}} \tilde{q}(\mathbf{x}_{b_i},\mathbf{x}_{c_i}) = \tilde{q}(\mathbf{x}_{c_i}) \tag{9}$$

Definition 8 *If $b_i \neq \emptyset$ we define a FDA factorization for a given ADF by*

$$q(\mathbf{x}) = \prod_{i=1}^{m} \tilde{q}(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}) \tag{10}$$

A FDA factorization is k-bounded *if the size of the sets $\{b_i, c_i\}$ is bounded by a constant k independent of n.*

Remark: *Any FDA factorization can easily be transformed into an acyclic Bayesian network which has the same largest clique size. The FDA factorization is only a more compact representation. Therefore the class of FDA factorizations is identical to the class of acyclic Bayesian networks. Sampling is done iteratively, starting with* $\tilde{q}(x_{b_1})$

Proposition 9 *Let a consistent set of marginal distributions* $\tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})$ *be given. Then the FDA factorization defines a valid distribution (*$\sum q(\mathbf{x}) = 1$*). Furthermore*

$$q(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}) = \tilde{q}(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}), \;\; i = 1, \ldots m \tag{11}$$

whereas in general

$$q(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}) \neq \tilde{q}(\mathbf{x}_{b_i}, \mathbf{x}_{c_i}), \;\; i = 1, \ldots m \tag{12}$$

The proof follows from the definition of marginal probabilities. The proof of equation (11) is somewhat technical, but straightforward. The inequality (12) is often overlooked. It means that sampling from the factorization does not reproduce the given marginals.

The next theorem was proven in (20). It formulates a condition under which the *FDA* factorization reproduces the marginals.

Theorem 10 (Factorization Theorem). *Let* $f(\mathbf{x}) = \sum_{i=1}^{m} f_{s_i}(\mathbf{x})$ *be an additive decomposition. If*

$$\forall i = 1, \ldots, m; \quad b_i \neq \emptyset \tag{13}$$

$$\forall i \geq 2 \; \exists j < i \;\; \textit{such that} \; c_i \subseteq s_j \tag{14}$$

then

$$p_\beta(\mathbf{x}) = \prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i}|\mathbf{x}_{c_i}) = \frac{\prod_{i=1}^{m} p_\beta(\mathbf{x}_{b_i}, \mathbf{x}_{c_i})}{\prod_{i=2}^{m} p_\beta(\mathbf{x}_{c_i})} \tag{15}$$

The Factorization Theorem is not restricted to distributions and their factorization. It is connected to the decomposition of G_{ADF}. A general formulation and a historical survey of this important theorem can be found in (1).

Definition 11 *The constraint defined by equation* (14) *is called the* running intersection property *(RIP).*

The above theorem does not address the problem how to compute a good or even an exact factorization. The construction defined by equation (7) depends on the sequence $s_1, \ldots, s_m$. For many sequences the *RIP* might not be fulfilled. But if the sequence is permutated, it might be possible that the RIP will be fulfilled. Furthermore, we can join two or more sub-functions, resulting in larger sets $\tilde{s_i}$. It might be that using these larger sets, the *RIP* is fulfilled. For an efficient *FDA* we are looking for $k - bounded$ factorizations which fulfill the *RIP*.

Testing all the sequences is prohibitive. Actually, it turns out that the computation of exact factorizations is done better by investigating the corresponding interaction graph G_{ADF}. A well-known algorithm computes a *junction tree* of $G_{ADF}(see(10))$. From the junction tree a factorization can easily be obtained. This factorization fulfills the *RIP*. A short description of the algorithm can be found in (15). The largest clique of the junction tree gives the largest marginal of the factorization. The decision problem if there exists an k-bounded junction tree is NP in general.

The space complexity of exact factorizations has been investigated in (6). For many problems the size of the largest clique is $O(n)$, making a numerical application using this factorization prohibitive. Thus for real applications we are looking for good k-bounded factorizations which violate the *RIP* in a few cases only.

2.2 The Factorized Distribution Algorithm

We first describe our algorithm *FDA* which runs with any *FDA* factorization.

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.
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)

The computational complexity of *FDA* is $O(N * \sum_{i=1}^{m} 2^{|s_i|})$, where $|s_i|$ denotes the size of the marginals. For a k-bounded *FDA* factorization we obtain $O(N*m*2^k)$, thus the complexity is linear in m. If the *FDA* factorization is exact, then the convergence proof of *BEDA* is valid for *FDA* too. But since *FDA* uses finite samples to estimate the conditional probabilities, convergence to the optimum will depend on the size of the sample. Sampling theory can be used to estimate the probability of convergence for a given sample size.

A factorization fulfilling the *RIP* is sufficient for convergence to the optimum, but not necessary. But such a factorization can be difficult to compute or may be not $k - bounded$. Therefore we use *FDA* mainly with approximate factorizations. A good approximate factorization should include all edges of the interaction graph.

We have implemented a general heuristic which automatically computes a *FDA* factorization. The heuristic uses mainly *merging* of sub-functions. Let us discuss the problem with a simple loop.

$$s_1 = \{1, 2\}, s_2 = \{2, 3\}, s_3 = \{3, 4\}, \mathbf{s}_4 = \{1, 4\}$$

All possible sequences end in $b_4 = \emptyset$. We can use the factorization $q(\mathbf{x}) = \tilde{q}(x_1, x_2)\tilde{q}(x_3|x_2)\tilde{q}(x_4|x_3)$ using s_1, s_2, s_3 only. But if the sub-functions f_3 and f_4 are merged then we obtain from our procedure

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

This factorization uses all edges from G_{ADF} but violates the *RIP*. Merging of sub-functions lead to larger marginals. Therefore a good heuristic has three conflicting goals: to minimize the number of *RIP* violations, to use all dependencies of G_{ADF}, and to find a k-bounded factorization with a small k.

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. It is given a cut size, which bounds the size of the sets. 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. If the size is larger than the cut size, variables are randomly left out. After completing the merge phase, the algorithm calculates $\tilde{c}_j$, $\tilde{b}_j$ and $\tilde{d}_j$ analogous to the construction given by (7).

For 2-D grid problems exact factorizations are not k-bounded. If the *ADF* consists of sub-functions of two neigboring grid points only, our sub-function merger algorithm computes a factorization using marginals up to order 3. The factorization covers the whole interaction graph G_{ADF}. For a 3*3 grid the sub-function merger constructs the following factorization

$$q(\mathbf{x}) = \tilde{q}(x_4, x_5)\tilde{q}(x_3|x_4)\tilde{q}(x_2|x_5)\tilde{q}(x_1|x_2, 4) \\ \tilde{q}(x_7|x_4)\tilde{q}(x_0|x_1, x_3)\tilde{q}(x_8|x_5, x_7)\tilde{q}(x_6|x_3, x_7) \quad (16)$$

Because grids are very common, we have in addition implemented a number of specialized factorizations for functions defined on grids. Our presentation of the sub-function merger algorithm has been very short. In the area of Bayesian networks, the problem has been investigated in (3).

FDA has experimentally proven to be very successful on a number of functions where standard genetic algorithms fail to find the global optimum. In (16) the scaling behavior for various test functions has been studied. For recent surveys the reader is referred to (17; 19; 15).

3 The maximum entropy principle

In this section we investigate the problem of approximating an unknown distribution given some information in a theoretical framework.

Let $\mathbf{x} = (x_1, \ldots, x_n)$, $B = \{0, 1\}^n$. Let $\phi_j : B \to \{0, 1\}, j = 1, m$ be binary functions, often called features. Let a sample S be given, $\tilde{p}(\mathbf{x})$ the observed distribution. Let

$$E_{\tilde{p}}(\phi_j) = \sum_{x \in B} \tilde{p}(\mathbf{x})\phi_j(\mathbf{x}) \quad (17)$$

Note that ϕ_j can specify any marginal distribution, but also more general expectations.

Problem *We are looking for a distribution $p(\mathbf{x})$ which fulfills the constraints*

$$E_p(\phi_j) = E_{\tilde{p}}(\phi_j) \quad (18)$$

and is in some sense plausible.

If only a small number of features 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 of a distribution is defined by*

$$H(p) = -\sum_x p(\mathbf{x}) \ln(p(\mathbf{x})) \tag{19}$$

Maximum entropy principle (MaxEnt): *Let*

$$P = \{p | E_p(\phi_j) = E_{\tilde{p}}(\phi_j), \;\; j = 1, \ldots, m\} \tag{20}$$

Then the MaxEnt solution is given by

$$p^* = \text{argmax}_{p \in P} H(p) \tag{21}$$

The maximum entropy principle formulates the *principle of indifference*. If no constraints are specified, the uniform random distribution will be the solution. MaxEnt has a long history in physics and probabilistic logic. The interested reader is referred to (8; 9).
The MaxEnt solution can be computed from the constrained optimization problem

$$p^* = \text{argmax}_{p \in P} H(p) \tag{22}$$

$$\sum_x p(x) = 1 \tag{23}$$

$$\sum_x p(\mathbf{x}) \phi_j(\mathbf{x}) = E_{\tilde{p}}(\phi_j) \tag{24}$$

This is a convex optimization problem with linear constraints. Therefore it has a unique solution. It can be found by introducing Lagrange multipliers.

$$L(p, \Lambda, \gamma) = -\sum_x p(\mathbf{x}) \ln p(\mathbf{x}) + \gamma(\sum_x p(x) - 1) \tag{25}$$

$$+ \sum_{i=j}^{m} \lambda_j (E_p(\phi_j) - E_{\tilde{p}}(\phi_j)) \tag{26}$$

where $\Lambda = (\lambda_1, \ldots, \lambda_m)$.
The maxima of L can be obtained by computing the derivatives of L. We compute

$$\frac{\partial L}{\partial p(\mathbf{x})} = -\ln p(\mathbf{x}) - 1 - \sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{x}) + \gamma \tag{27}$$

Setting the derivative to zero gives the parametric form of the solution

$$p^*(\mathbf{x}) = \exp(\gamma - 1) \exp \sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{x}) \tag{28}$$

Definition 13 *Let Q be the space of distributions of the parametric form*

$$Q = \{q | q(\mathbf{x}) = \frac{1}{Z} \exp \sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{x})\} \tag{29}$$

In order to characterize the MaxEnt solution, the relative entropy between distributions has to be introduced.

Definition 14 *The relative entropy or Kullback-Leibler divergence between two distributions p and q is defined as (see (4))*

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

Note that $KLD(p, q) \neq KLD(q, p)$, i.e. KLD is not symmetric! If $q(\mathbf{x}) = 0$ and $p(\mathbf{x}) > 0$ we have $KLD(p, q) = \infty$. This means that KLD gives large weights to values near zero. In all other aspects KLD is a distance measure. The following lemma holds (4).

Lemma 15 *For any two probability distributions p and q, $KLD(p, q) \geq 0$ and $KLD(p, q) = 0$ iff $p = q$.*

In our application KLD fulfills the Pythagorean property.

Lemma 16 (Pythagorean Property) *Let $p \in P$, $q \in Q$, and $p^* \in P \cap Q$, then*

$$KLD(p, q) = KLD(p, p^*) + KLD(p^*, q) \tag{31}$$

The proof is straightforward. The following theorem follows easily from the lemma:

Theorem 17 (Maximum Entropy Solution). *If $p^* \in P \cap Q$, then*

$$p^*(\mathbf{x}) = \text{argmax}_{p \in P} H(p) \tag{32}$$

Furthermore, p^* is unique.

The constrained optimization problem can be solved by standard mathematical algorithms. But also specialized algorithms have been invented, a popular one is the *Generalized Iterative Scaling Algorithm (GIS)* (5). Unfortunately the computational amount of the algorithm is exponential in general.

Obviously the MaxEnt approximation minimizes the relative entropy $KLD(p, u)$ to the uniform distribution u. Thus MaxEnt is a special case of the *minimum relative entropy* $MinRel$ principle. But there exists another justification of the MaxEnt solution, it is given by the *Maximum Log-Likelihood* principle.

Definition 18 *Let $S = \{X_1, \ldots, X_N\}$ be an empirical sample, $\tilde{p}(\mathbf{x})$ the empirical distribution. Let $q(\mathbf{x})$ be a distribution. Then the likelihood that q generates the data is given by*

$$LH(q) = \prod_{i=1}^{N} q(X_i) = \prod_{x \in B} q(\mathbf{x})^{N\tilde{p}(\mathbf{x})} \tag{33}$$

The log-likelihood is defined as

$$LogLH(q) = \sum_{x \in B} N\tilde{p}(\mathbf{x}) \ln q(\mathbf{x}) \tag{34}$$

Theorem 19 (Maximum Log-Likelihood solution). *If $p^* \in P \cap Q$, then*

$$p^*(\mathbf{x}) = \mathrm{argmax}_{q \in Q}\, LogLH(q) \tag{35}$$

Furthermore, p^* is unique.

Proof. Let $\tilde{p}(\mathbf{x})$ be the observed distribution. Clearly $\tilde{p} \in P$. Suppose $q \in Q$ and $p^* \in P \cap Q$. We show that $LogLH(q) \leq LogLH(p^*)$. The Pythagorean property gives

$$KLD(\tilde{p}, q) = KLD(\tilde{p}, p^*) + KLD(p^*, q)$$

Therefore

$$\begin{gathered} KLD(\tilde{p}, q) \geq KLD(\tilde{p}, p^*) \\ -H(\tilde{p}) - \frac{1}{N} LogLH(q) \geq -H(\tilde{p}) - \frac{1}{N} LogLH(p^*) \\ LogLH(q) \leq LogLH(p^*) \end{gathered}$$

Thus the MaxEnt solution can be viewed under both the maximum entropy framework as well as the maximum log-likelihood framework. This means that p^* will fit the data as closely as possible while as the maximum entropy solution it will not assume facts beyond those given by the constraints.

We next investigate the relation of *FDA* factorizations and the MaxEnt solution.

Definition 20 *Given an ADF the MaxEnt problem is called* complete marginal *if all marginal distributions $\tilde{p}(x_{s_k})$ are given. The FDA factorization is called* complete*, if the graphical model of the factorization contains the interaction graph G_{ADF}.*

Theorem 21. *The MaxEnt solution of a complete marginal MaxEnt problem is the exact distribution. The MaxEnt solution of any complete FDA factorization is the exact distribution.*

Proof. Let a complete marginal MaxEnt problem be given. Then the features $\phi(\mathbf{x}_{s_i})$ are defined by $E_{\tilde{p}}\phi(\mathbf{x}_{s_i}) = \tilde{p}(\mathbf{x}_{s_i})$. We abbreviate the parameters in equation (29) by $\lambda(\mathbf{x}_{s_i})$. Now set $\lambda(\mathbf{x}_{s_i})\tilde{p}(\mathbf{x}_{s_i}) = \beta f(\mathbf{x}_{s_i})$. Thus the exact distribution is in the set Q. Obviously fulfills the exact distribution the marginalization constraints. Therefore the exact distribution is the MaxEnt solution. The proof for complete *FDA* factorizations works accordingly.

This theorem is the justification of the MaxEnt principle for *FDA* factorizations. If all relevant information of an *ADF* is given, the unique MaxEnt solution is the exact distribution. But the computation of the MaxEnt solution for a complete *FDA* factorization is *exponential* if it does not fulfill the *RIP*. Therefore *FDA* just samples from the factorization using the computed marginals. But if the RIP is violated the generated distribution might be different from the exact distribution, even if the factorization is complete. Thus in contrast to the MaxEnt solution, the *FDA* factorization with its simple sampling procedure might not converge to the optima of the function.

We next describe another approach to approximate the Boltzmann distribution. In this method the Kullback-Leibler divergence to the Boltzmann distribution is minimized without computing the marginal distributions from samples. Instead the values of the marginals are computed from a difficult constrained minimization problem.

4 Approximating the Boltzmann distribution

The Boltzmann distribution plays an important role in statistical physics. Therefore a number of approximation techniques have been developed. We present an approach where an approximation q

$$q(\mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{k}\tilde{q}(\mathbf{x}_k) \tag{36}$$

is computed which minimizes the relative entropy to the Boltzmann distribution. The method is described in (15) using the terminology of physics. We give here a short mathematical derivation. The relative entropy is given by

$$\begin{aligned} KLD(q|p_\beta) &= \sum_x q(\mathbf{x})\ln q(\mathbf{x}) - \sum_x q(\mathbf{x})\ln p_\beta(\mathbf{x}) \\ &= -H(q) + \ln Z - \beta E_q(f) \end{aligned}$$

We again assume that the function is defined by an *ADF*. Then we easily obtain

$$E_q(f) = \sum_{i=1}^{m} q(\mathbf{x}_{s_i}) f_i(x_{s_i}) \tag{37}$$

The expected average of the function can be computed using the marginals. More difficult problem is the computation of $H(q)$. We will restrict our discussion to *FDA* factorizations.

$$q(\mathbf{x}) = \frac{\prod_{i=1}^{m} \tilde{q}(\mathbf{x}_{s_i})}{\prod_{i=1}^{m} \tilde{q}(\mathbf{x}_{c_i})} \tag{38}$$

For this factorization one computes

$$\begin{aligned} H(q) &= -\sum_{i=1}^{m} q(\mathbf{x}_{s_i}) \ln \tilde{q}(\mathbf{x}_{s_i}) + \sum_{i=1}^{m} q(\mathbf{x}_{c_i}) \ln \tilde{q}(\mathbf{x}_{c_i}) \\ &\approx -\sum_{i=1}^{m} q(\mathbf{x}_{s_i}) \ln q(\mathbf{x}_{s_i}) + \sum_{i=1}^{m} q(\mathbf{x}_{c_i}) \ln q(\mathbf{x}_{c_i}) \end{aligned} \tag{39}$$

Note that for *FDA* factorizations which do not fulfill the RIP we have $q(\mathbf{x}_{s_i}) \neq \tilde{q}(\mathbf{x}_{s_i})$ if we use *FDA* sampling. But in order to make the problem tractable we assume $q(\mathbf{x}_{s_i}) = \tilde{q}(\mathbf{x}_{s_i})$ Then minimizing KLD leads to the following constraint optimization problem.

Definition 22 (Bethe-Kikuchi approximation) *Compute the minimum of all FDA factorizations $q(\mathbf{x})$ (equation* (38)*)*

$$\begin{aligned} \textbf{\textit{argmin}}_q KLD(q|p_\beta) &= \sum_{i=1}^{m} q(\mathbf{x}_{s_i}) \ln q(\mathbf{x}_{s_i}) \\ &\quad - \sum_{i=1}^{m} q(\mathbf{x}_{c_i}) \ln q(\mathbf{x}_{c_i}) - \beta \sum_{i=1}^{m} q(\mathbf{x}_{s_i}) f_i(x_{s_i}) \end{aligned} \tag{40}$$

subject to the constraints for all s_j with $c_i \subset s_j$

$$\sum_{x_{s_i}} q(\mathbf{x}_{s_i}) = 1 \tag{41}$$

$$\sum_{x_{s_j} \setminus x_{c_i}} q(\mathbf{x}_{s_j}) = q(\mathbf{x}_{c_i}) \tag{42}$$

Remark: The *minimization problem is not convex! There might exist many local minima. Furthermore, the exact distribution might not be a local minimum if the factorization violates the RIP.*
The constraints make the the solution of the problem difficult. We again use the Lagrange function.

$$\begin{aligned} L(p, \Lambda, \Gamma) &= KLD(q|p_\beta) + \sum_{i=1}^{m} \gamma_i \Big(\sum_{x_{s_i}} q(\mathbf{x}_{s_i}) - 1\Big) \\ &\quad + \sum_{i=1}^{m} \sum_{x_{c_i}} \lambda(s_j, c_i) \Big(\sum_{x_{s_j} \setminus x_{c_i}} q(\mathbf{x}_{s_j}) - q(\mathbf{x}_{c_i}) \Big) \end{aligned} \tag{43}$$

The minima of L are determined be setting the derivatives of L zero. The independent variables are $q(\mathbf{x}_{s_i})$,$q(\mathbf{x}_{c_i})$,γ_i,and $\lambda(s_j, c_i)$. We obtain

$$\frac{\partial L}{\partial q(\mathbf{x}_{s_i})} = \ln q(\mathbf{x}_{s_i}) + 1 - \beta q(\mathbf{x}_{s_i}) f_(\mathbf{x}_{s_i}) + \gamma_i + r(\Lambda) \tag{44}$$

Setting the derivative to zero, we obtain the parametric form

$$q(\mathbf{x}_{s_i}) = e^{-1-\gamma_i} e^{-r(\Lambda)} e^{\beta f(x_{s_i})} \tag{45}$$

Note that the parametric form is again exponential. The Lagrange factors Γ are easily computed from $\sum_{x_{s_i}} q(x_{s_i}) = 1$. The factors Λ have to be determined from a non-linear system of equations. Before we describe an algorithm for solving it, we describe a simple special case, the mean-field approximation.

4.1 The mean-field approximation

In the mean-field approximation uni-variate marginals only are used.

$$q(\mathbf{x}) = \prod_{i=1}^{n} q(x_i) \tag{46}$$

We obtain for its entropy and $E_q(f)$.

$$H(q) = -\sum_x \prod_{i=1}^{n} q(x_i) \sum_{j=1}^{n} \ln q(x_j) = -\sum_{i=1}^{n} \sum_{x_i} q(x_i) \ln q(x_i)$$

$$E_q(f) = \sum_x \prod_{i=1}^{n} q(x_i) f(\mathbf{x}) = \sum_{i=1}^{m} \prod_{j \in s_i}^{n} q(x_j) f(\mathbf{x}_{s_i})$$

For the mean-field approximation the Kullback-Leibler divergence is convex, thus the minimum exists and is unique. The minimum is obtained by setting the derivative of KLD equal to zero, using the uni-variates as variables. We abbreviate $q_i = q(x_i = 1)$.

Theorem 23. *The uni-variate marginals of the mean-field approximation are given by the nonlinear equation*

$$q_i^* = \frac{1}{1 + e^{\frac{\partial E_q}{\partial q_i}}} \tag{47}$$

Proof. We compute the derivative

$$\frac{\partial KLD}{\partial q_i} = \ln \frac{q_i}{1 - q_i} + \frac{\partial E_q}{\partial q_i} = 0 \tag{48}$$

The solution gives (47).

Equation (47) can be solved by an iteration scheme.

5 Loopy Belief Models and Region Graphs

The computation of the Bethe-Kikuchi approximation is difficult. We decided to use a specialized algorithm, recently proposed in (26). It is based on the concept of a *region graph*. A region graph is a loopy graphical model. It is strongly related to partially ordered sets (posets) or Hasse diagrams. Similar or identical structures have been presented in (2; 14; 24). This section follows largely the notation of (26).

5.1 Regions

The region graph was introduced in (26) using a different graphical model, the *factor graph*. The factor graph is a more detailed way to describe an additive decomposition. We decided not to use the factor graph to show the connection of region graphs to junction trees.

Definition 24 *Let $S = \{s_1, \ldots, s_m\}$ be the index set of an additive decomposition for a fitness function f, such that*

$$f(\mathbf{x}) = \sum_{s_i \in S} f_i(\mathbf{x}_{s_i}) \tag{49}$$

*A **region** $R = (s_R, I_R)$ is a set of variable indices $s_R \subseteq \{1, \ldots, n\}$ and a set of sub-function indices $I_R \subseteq \{1, \ldots, m\}$, such that*

$$\forall i \in I_R : s_i \subseteq s_R \tag{50}$$

The variables contained in the region are indexed by s_R, whereas I_R contains the indices of the sub-functions which are contained in the region. It is asserted by (50) *that all variables needed for the contained sub-functions are in s_R.*

Our goal is to approximate the Boltzmann distribution with the energy $E(\mathbf{x}) = -f(\mathbf{x})$ by minimizing the relative entropy. For a region we define a local energy.

Definition 25 *For a region R, define the **region energy***

$$E_R(\mathbf{x}_{s_R}) := -\sum_{i \in I_R} f_i(\mathbf{x}_{s_i}) \tag{51}$$

Region energies are defined only for those regions which contain the variables of at least one sub-functions. We will try to compute the marginals q_R on R from the Bethe-Kikuchi minimization problem. In (26) the marginals are called the *beliefs* on R. This is the terminology of Bayesian networks.

5.2 Region Graph

Definition 26 *A **region graph** is a graph $G = (\mathcal{R}, E_{\mathcal{R}})$, where $\mathcal{R}$ is a set of regions and $E_{\mathcal{R}}$ is a set of directed edges. An edge $(R_p, R_c) \in E_{\mathcal{R}}$ is only allowed if $s_{R_c} \subset s_{R_p}$. If $(R_p, R_c) \in E_{\mathcal{R}}$, we call R_p a parent of R_c and R_c child of R_p.*

Since $E_{\mathcal{R}}$ imposes a partial ordering on the set of regions, in (14) the same structure was called a partially ordered set or *poset*.

Lemma 27 *A region graph is directed acyclic.*

Proof. This follows immediately from the requirement that edges are only allowed from supersets to subsets.

A junction tree can be turned into a region graph by creating a region for every cluster and every separator and adding edges from each node to each neighboring separator.

The global distribution of a junction tree is the product of all distributions on the clusters divided by the distributions of all the separators (see (15)). We generalize this factorization by introducing counting numbers of the regions.

Definition 28 *The **counting number** c_R of a region R is defined recursively as*

$$c_R = 1 - \sum_{R' \in A(R)} c_{R'} \tag{52}$$

where $A(R)$ is the set of all ancestors of R.

This is well-defined, because the region graph is cycle-free. The maximal regions (without ancestors) have counting number 1. From there, the counting numbers can be calculated from the top to the bottom of the graph.

5.3 Region Graph and Junction Tree

The junction tree property has an equivalent on the region graph, called the region graph condition.

Definition 29 *We call a region graph **valid** if it fulfills the **region graph condition**, which states that*

1. *For all variable indices $i \in \{1, \ldots, n\}$ the set $\mathcal{R}_{X,i} := \{R \in \mathcal{R} | i \in s_R\}$ of all regions R that contain X_i form a connected subgraph with*

$$\sum_{R \in \mathcal{R}_{X,i}} c_R = 1 \ , \tag{53}$$

and

2. *For all sub-function indices $i \in \{1,\ldots,m\}$ the set $\mathcal{R}_{f,i} := \{R \in \mathcal{R} | i \in I_R\}$ of all regions R that contain f_i form a connected subgraph with*

$$\sum_{R \in \mathcal{R}_{f,i}} c_R = 1 \ . \tag{54}$$

The connectivity of the subgraph, like the junction property, prevents that in different parts of the graph contradictory beliefs can evolve. The condition on the counting numbers makes sure that every variable and every sub-function is counted exactly once.

The Kikuchi approximation of the Boltzmann distribution is defined as follows (see also (23)).

Definition 30 *The **Kikuchi approximation** of the Boltzmann distribution for a region graph is*

$$k(\mathbf{x}) = \prod_{R \in \mathcal{R}} q_R(\mathbf{x}_{s_R})^{c_R} \tag{55}$$

*In general, it is not normalized and therefore no probability distribution. The **normalized Kikuchi approximation***

$$p_k(\mathbf{x}) = \frac{k(\mathbf{x})}{\sum_{\mathbf{y}} k(\mathbf{y})} \tag{56}$$

is a probability distribution.

If the region graph is derived from a junction tree, with q_R being the marginal distributions on the clusters and separators, $k(\mathbf{x})$ is a valid distribution, since its definition coincides with the junction tree distribution. It has been proven that cycle-free region graphs reproduce the exact distribution (25). Thus they give the same result a a junction tree.

The problem of sampling from a Kikuchi approximation is discussed later. We next describe a local iteration algorithm, based on the region graph and message passing between regions. The iteration algorithm minimizes the Kullback-Leibler divergence.

6 The Concave Convex Procedure

The Concave Convex Procedure (CCCP) (27) is a variant of Generalized Belief Propagation GBP proposed in (25). It is based on the observation that the Lagrangian consists of a convex and a negative convex (concave) term. The *CCCP* algorithm alternates between updates of the convex and the concave term.

6.1 The convex and concave Lagrangian

We now derive the *CCCP* update procedure, following (27). The algorithm is fairly complex. A detailed description can be found in the dissertation (7). The Lagrangian to be minimized is given by equation (43)

$$\begin{aligned} L = \sum_{R \in \mathcal{R}} c_R \left(\sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \beta E(\mathbf{x}_{s_R}) + \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \log q_R(\mathbf{x}_{s_R}) \right) \\ + \sum_{R \in \mathcal{R}} \gamma_R \left(1 - \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \right) \\ + \sum_{(P,R) \in E_{\mathcal{R}}} \sum_{\mathbf{x}_{s_R}} \lambda_{PR}(\mathbf{x}_{s_R}) \left(\sum_{\mathbf{x}_{s_P \setminus s_R}} q_P(\mathbf{x}_{s_P}) - q_R(\mathbf{x}_{s_R}) \right) \end{aligned} \quad (57)$$

The basic idea of *CCCP* is to split L in a convex and a concave part. The problematical part is the entropy term: For regions with $c_R > 0$, the entropy term is convex, for regions with $c_R < 0$ it is concave. The average energy and the constraints are linear in the q_R, so it does not matter where we put them.

To avoid an arbitrary separation into convex and concave regions, we set

$$c_{\text{max}} = \max_R c_R \quad (58)$$

and use this definition to split up L into a convex part

$$\begin{aligned} L_{\text{vex}} = \sum_{R \in \mathcal{R}} c_{\text{max}} \left(\sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \beta E_R(\mathbf{x}_{s_R}) + \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \ln q_R(\mathbf{x}_{s_R}) \right) \\ + \sum_{R \in \mathcal{R}} \gamma_R \left(1 - \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \right) \\ + \sum_{(P,R) \in E_{\mathcal{R}}} \sum_{\mathbf{x}_{s_R}} \lambda_{PR}(\mathbf{x}_{s_R}) \left(\sum_{\mathbf{x}_{s_P \setminus s_R}} q_P(\mathbf{x}_{s_P}) - q_R(\mathbf{x}_{s_R}) \right) \end{aligned} \quad (59)$$

and a concave part

$$\begin{aligned} L_{\text{ave}} = \sum_{R \in \mathcal{R}} (c_R - c_{\text{max}}) (\sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) E_R(\mathbf{x}_{s_R}) \\ + \sum_{\mathbf{x}_{s_R}} q_R(\mathbf{x}_{s_R}) \ln q_R(\mathbf{x}_{s_R})) \end{aligned} \quad (60)$$

Obviously $L = L_{\text{vex}} + L_{\text{ave}}$.

6.2 The outer and inner loops

CCCP consists of an *inner loop* in which the messages are updated until convergence, and an *outer loop* in which the current estimates of the marginals are updated. For the inner loop we use the iteration index τ and for the outer the index ξ.

The Outer Loop

For the outer loop we make the ansatz

$$\nabla L_{\text{vex}}^{\xi+1} + \nabla L_{\text{ave}}^{\xi} = 0 \tag{61}$$

where ∇L denotes the vector of the partial derivatives of L with respect to the marginals $q_R(\mathbf{x}_{s_R})$. The derivatives are given by

$$\frac{\partial L_{\text{vex}}}{\partial q_R(\mathbf{x}_{s_R})} = c_{\max}\left(\beta E_R(\mathbf{x}_{s_R}) + \ln q_R(\mathbf{x}_{s_R}) + 1\right) - \gamma_R \\ - \sum_{P|(P,R)\in E_{\mathcal{R}}} \lambda_{PR}(\mathbf{x}_{s_R}) + \sum_{C|(R,C)\in E_{\mathcal{R}}} \lambda_{RC}(\mathbf{x}_{s_C}) \tag{62}$$

and

$$\frac{\partial L_{\text{ave}}}{\partial q_R(\mathbf{x}_{s_R})} = (c_R - c_{\max})\left(\beta E_R(\mathbf{x}_{s_R}) + \ln q_R(\mathbf{x}_{s_R}) + 1\right) \quad . \tag{63}$$

Inserting (62) and (63) into (61) gives

$$c_{\max}\left(\beta E_R(\mathbf{x}_{s_R}) + \ln q_R^{\xi+1}(\mathbf{x}_{s_R}) + 1\right) - \gamma_R \\ - \sum_{P|(P,R)\in E_{\mathcal{R}}} \lambda_{PR}(\mathbf{x}_{s_R}) + \sum_{C|(R,C)\in E_{\mathcal{R}}} \lambda_{RC}(\mathbf{x}_{s_C}) \\ + (c_R - c_{\max})\left(\beta E_R(\mathbf{x}_{s_R}) + \ln q_R^{\xi}(\mathbf{x}_{s_R}) + 1\right) = 0 \quad . \tag{64}$$

Solving this for $q_R^{\xi+1}(\mathbf{x}_{s_R})$ gives the update equations for the marginals in the outer loop:

$$q_R^{\xi+1}(\mathbf{x}_{s_R}) = q_R^{\xi}(\mathbf{x}_{s_R})^{\frac{c_{\max}-c_R}{c_{\max}}} \exp\left[-\frac{c_R}{c_{\max}}\beta E_R(\mathbf{x}_{s_R})\right] \\ \exp\left[\frac{\gamma_R - c_R}{c_{\max}} + \frac{1}{c_{\max}}\Big(\sum_{P|(P,R)\in E_{\mathcal{R}}} \lambda_{PR}(\mathbf{x}_{s_R})\Big)\right] \\ \exp -\frac{1}{c_{\max}}\left[\sum_{C|(R,C)\in E_{\mathcal{R}}} \lambda_{RC}(\mathbf{x}_{s_C})\right] \tag{65}$$

For the regions with $c_R = c_{\max}$ the previous marginal $q_R^{\xi}(\mathbf{x}_{s_R})$ cancels out. We next introduce messages (see (25))

$$m_{PC}(\mathbf{x}_{s_C}) := e^{\frac{1}{c_{\max}}\lambda_{PC}(\mathbf{x}_{s_C})} \tag{66}$$

and choose γ_R appropriately for normalization, which changes the update equation to

$$q_R^{\xi+1}(\mathbf{x}_{s_R}) \;\propto\; q_R^{\xi}(\mathbf{x}_{s_R})^{\frac{c_{\max}-c_R}{c_{\max}}} e^{-\frac{c_R}{c_{\max}}\beta E_R(\mathbf{x}_{s_R})} \frac{\prod_{P|(P,R)\in E_{\mathcal{R}}} m_{PR}(\mathbf{x}_{s_R})}{\prod_{C|(R,C)\in E_{\mathcal{R}}} m_{RC}(\mathbf{x}_{s_C})} \tag{67}$$

The Inner Loop

The inner loop update equation for the messages can be derived by inserting (67) into the consistency equation

$$\sum_{\mathbf{x}_{s_P\setminus s_R}} q_P(\mathbf{x}_{s_P}) = q_R(\mathbf{x}_{s_R}) \tag{68}$$

This gives

$$\begin{aligned}\sum_{\mathbf{x}_{s_P\setminus s_R}} & q_P^t(\mathbf{x}_{s_P})^{\frac{c_{\max}-c_P}{c_{\max}}} e^{-\frac{c_P}{c_{\max}}\beta E_P(\mathbf{x}_{s_P})} \frac{\prod_{Q|(Q,P)\in E_{\mathcal{R}}} m_{QP}(\mathbf{x}_{s_P})}{\prod_{C|(P,C)\in E_{\mathcal{R}}} m_{PC}(\mathbf{x}_{s_C})} \\ &= q_R^t(\mathbf{x}_{s_R})^{\frac{c_{\max}-c_R}{c_{\max}}} e^{-\frac{c_R}{c_{\max}}\beta E_R(\mathbf{x}_{s_R})} \frac{\prod_{Q|(Q,R)\in E_{\mathcal{R}}} m_{QR}(\mathbf{x}_{s_R})}{\prod_{C|(R,C)\in E_{\mathcal{R}}} m_{RC}(\mathbf{x}_{s_C})}\end{aligned} \tag{69}$$

The message $m_{PR}(\mathbf{x}_{s_R})$ is independent of the summation variables $\mathbf{x}_{s_P\setminus s_R}$, so it can be extracted from the sum. It appears in the denominator on the left side of (69) and in the numerator on the right side. This allows to solve the equation for this message.

With the abbreviations

$$g_R(\mathbf{x}_{s_R}) := q_R^{\xi}(\mathbf{x}_{s_R})^{\frac{c_{\max}-c_R}{c_{\max}}} e^{-\frac{c_R}{c_{\max}}\beta E_R(\mathbf{x}_{s_R})} \tag{70}$$

$$h_R(\mathbf{x}_{s_R}) := \frac{\prod_{Q|(Q,R)\in E_{\mathcal{R}}} m_{QR}^{\tau}(\mathbf{x}_{s_R})}{\prod_{C|(R,C)\in E_{\mathcal{R}}} m_{RC}^{\tau}(\mathbf{x}_{s_C})} \tag{71}$$

we arrive at the inner loop update equation

$$m_{PR}^{\tau,\mathrm{upd}}(\mathbf{x}_{s_R}) = m_{PR}^{\tau}(\mathbf{x}_{s_R}) \sqrt{\frac{\sum_{\mathbf{x}_{s_P\setminus s_R}} g_P(\mathbf{x}_{s_P}) h_P(\mathbf{x}_{s_P})}{g_R(\mathbf{x}_{s_R}) h_R(\mathbf{x}_{s_R})}} \tag{72}$$

In order to make the iteration more robust, damping is applied Linear damping (26; 27) calculates the messages as a linear combination between the old and update messages:

$$m_{P\to R}^{\tau}(\mathbf{x}_R) = (1-\alpha) m_{P\to R}^{\tau-1}(\mathbf{x}_R) + \alpha m_{P\to R}^{\tau,\mathrm{upd}}(\mathbf{x}_R) \tag{73}$$

6.3 FDA factorizations and region graphs

The Kikuchi factorization and the concept of region graph has also been used for an EDA algorithm by (23). But the marginals are not determined from minimization of the Kullback-Leibler divergence, they are estimated from samples. Thus instead of the *FDA* factorization the Kikuchi factorization is used. But here a difficult problem appears, namely to sample from the Kikuchi approximation. The factorization is a loopy graphical model, it contains cycles. Therefore Gibbs sampling has been used in (23), which gives a valid distribution but is computational very expensive.

In contrast, we have implemented the full Bethe-Kikuchi method. In order to circumvent the sampling problem, we decided to use *FDA* factorizations only for the Kikuchi method. Given an arbitrary *FDA* factorization, we use the marginals used for the factorization to create a region graph. This is always possible. Then the Bethe-Kikuchi approximation is computed using this region graph. After the computation of the marginals the *FDA* factorization is used again for sampling.

7 The FDA Software

The FDA software allows to optimize arbitrary fitness functions of binary variables using various evolutionary algorithms like the simple genetic algorithm *GA*, the Univariate Marginal Algorithm *UMDA*, the Factorized Distribution Algorithm *FDA*, the Learning Factorized Distribution Algorithm *LFDA*, the Bethe-Kikuchi Approximation *BKDA*, and the Iterated Kernighan-Lin algorithm *IKL*. Details about these algorithms and a *free download of the FDA software* can be found at the web site
http://www.ais.fraunhofer.de/∼muehlen.
The following list summarizes the implemented algorithms.

- $-ag$ chooses the simple genetic algorithm GA. It allows one-point or two-point crossover and mutation.
- $-au$ chooses the *UMDA* algorithm. This algorithm estimates the univariate marginal distributions from the population and then samples the next generation with them.
- $-af$ chooses the *FDA* algorithm. The *FDA* algorithm expects the fitness function to be additively decomposable. To this end, the fitness functions are given as a sum of ”local functions” on a subset of the variables. If there is no such structure given, *UMDA* is used instead. $-af$ should be combined with one of the implemented automatic factorization algorithms described under $-j$.
- $-al$ selects the *LFDA* algorithm. This algorithm estimates the structure from the data using the BIC measure. The edge that increases the BIC most is added to the graphical model, until there is no more improvement possible. There are two subparameters to this algorithm: ’m’ gives

a maximal number of parents that a variable can have, and 'a' gives the weight of the structure penalty. The default values are '-alm8a0.5'.

- $-ab$ is our implementation of the Bethe-Kikuchi method *BKDA*. It is recommended for experienced users only. The user is prompted for a value of beta used for the Boltzmann distribution. It should be run for a single generation only.
- $-am$ is an iterated random algorithm. It creates the next population using mutation from the maximum of the current population. It requires an argument, which is the number of random bit flips which is performed on the copies of the maximum in order to generate the starting points of the next generation. It should be run together with a local optimizing algorithm like the Kernighan-Lin algorithm. In *FDA* this option should be used together with the $-k$ switch with a population size of 2 till 4.
- $-j$ selects the heuristic to compute the factorization for *FDA*. $-jm$ selects the sub-function join algorithm. The user should bound the size of the marginal distribution by specifying a number, e.g. $-jm9$. In addition a number of specialized factorizations for certain functions can be selected.
- $-k$ turns on the local optimizer Kernighan-Lin algorithm (12). It can be used together with all the other algorithms, excluding *BKDA* of course.

7.1 Local Hill Climbing

The use of a powerful local hill climbing algorithm changes the character of the search dramatically. We have implemented a general Kernighan-Lin algorithm (12). It is not a simple local hill climber, but it has a sophisticated backtracking procedure. Our implementation scales with $O(n^2)$. The algorithm can run together with *GA*, *UMDA*, *FDA* and *LFDA* using a small population size. It can also run as a simple iteration, the Iterated Kernighan-Lin algorithm *IKL* with a population size of two. New and promising start configurations are provided by randomly changing a certain percentage of the best solution obtained so far. The performance of *IKL* strongly depends on using a good value for this percentage.

For the graph bi-partitioning problem a very fast version of Kernighan-Lin has been implemented. It uses hash tables and a cut for backtracking. This implementation scales approximately linearly with n. It allows the optimization of graphs with more than 1000 variables with pop sizes up to 50.

8 Numerical Results

The EDA family of algorithms seems to be mature, at least for binary problems. It is time to demonstrate the state-of-the-art with large instances of popular benchmark problems. In (17) large graph bi-partitioning problems have been solved. Large problems have been also solved in (22). We will continue this work here. We will use problems on 2-D grids and Kauffman's $n - k$

function. The number of variables will be up to 900. Kauffman's function is an example of an *ADF* with random connections, the 2-D grid problems are important problems with regular connections.

2-D Spin glass:

$$f(\mathbf{x}) = \sum_{i,j} f_{i,j}(s_i, s_j) \tag{74}$$

s_j is one of the 4 neighbors of s_i, $s_i \in \{-1, 1\}$. The function values are randomly drawn from a Gaussian distribution in the interval $[-1, +1]$.

2-D grid on plaquettes:

$$f(\mathbf{x}) = \sum_{i,j} f_{i,j}(x_{i,j}, x_{i+1,j}, x_{i,j+1}, x_{i+1,k+1}) \tag{75}$$

The indices define a plaquette on the 2-D grid. The function values for each sub-function are randomly drawn from a Gaussian distribution in the interval $[-1, +1]$.

Kauffman random $n-3$:

$$f(\mathbf{x}) = \sum_{i=1}^{n} f_i(x_i, x_j, x_k) \tag{76}$$

The indices i, j, k are randomly chosen. The function values are drawn from a Gaussian distribution in $[0, 1]$.

The following table can be easily generated by any user of the *FDA* software. We do not have the space to compare all the possible *FDA* algorithms. The reader is encouraged to do tests himself.

The results confirm our theory. The standard *FDA* algorithm with large population sizes ($N = 30000$) performs very good on all instances. It should be no surprise that the population size has to be large. For the 2-D grid problems we used the special factorization $-jg5$. It uses marginals of size 5, even for the Ising problems. For the Kauffman function the sub-function merger algorithm $-jm$ creates marginals up to size 12 for $n = 400$ and size 15 for $n = 625$. It needs a large sample size to compute good estimates of these marginals. We remind the reader that for the *BKDA* algorithm the samples are computed only once, after computing the marginals. Surprisingly the *BKDA* algorithm runs slightly faster than *FDA*. Note that β has to be very large for the Kauffman function. This indicates that the Kauffman function has many local maxima. Still larger values of β do not improve the results for *BKDA*, because the convergence becomes a problem. Given the many assumptios used for *BKDA* we find the performance surprisingly good. But it seems to be not a breakthrough.

problem	size	alg.	sample.	β	best value
Ising	400	*FDA*	30000	-	297.259
Ising	400	*BKDA*	10000	30	297.259
Ising	625	*FDA*	30000	-	466.460
Ising	625	*BKDA*	10000	30	463.622
Plaqu.	400	*FDA*	10000	-	207.565
Plaque.	400	*BKDA*	10000	30	207.565
Plaque.	625	*FDA*	30000	-	320.069
Plaque.	625	*BKDA*	10000	30	320.132
Plauqe.	900	*FDA*	30000	-	459.274
Plaque.	900	*BKDA*	10000	30	454.237
$n-3$	400	*FDA*	10000	-	0.7535
$n-3$	400	*BKDA*	10000	12000	0.7520
$n-3$	625	*FDA*	30000	-	0.7501
$n-3$	625	*BKDA*	10000	15000	0.7436

Table 1. Comparison of *FDA* and $BKDA$ on large problems.

9 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 used for the approximation – minimizing the Kullback-Leibler divergence $KLD(q,u)$ to the uniform distribution u or minimizing $KLD(q,p_\beta)$ to the Boltzmann distribution p_β.

We have shown that the basic theory is the same for the two algorithms. This theory deals with the decomposition of graphical models and the computation of approximate factorizations. If the interaction graph G_{ADF} allows an exact factorization fulfilling the *RIP*, then both methods compute the exact distribution.

We have discussed two EDA algorithms in detail. The standard *FDA* algorithm 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. Factorizations which cover as much as possible from the interaction graph G_{ADF} are obtained by merging of sub-functions. The marginals of the standard *FDA* algorithm are computed from sampling the *FDA* factorization.

The Bethe-Kikuchi algorithm *BKDA* computes the marginals from a difficult constrained minimization problem. Because sampling from the original Bethe-Kikuchi factorization is difficult, we have extended the original approach. We use the *FDA* factorization which contains no loops. From this factorization the marginals are computed using the Bethe-Kikuchi minimization.

Our results show that for binary problems the EDA algorithms perform as good or even better than other heuristics used for optimization. At this

stage our algorithm is not yet optimized from a numerical point of view, nevertheless is is already competitive to more specialized algorithms.

In our opinion too many EDA researchers still investigate 1-D problems. Our theory (and also practice) shows that these problems can be solved exactly in polynomial time if the junction tree factorization is used. They pose no problem for optimization at all.

The interested reader can download our software from the WWW site *http://www.ais.fraunhofer.de/∼muehlen*.

References

[1] S. M. Aji and R. J. McEliece. The generalized distributive law. *IEEE Trans. Inform. Theory*, 46(2):325–343, March 2000.

[2] S. M. Aji and R. J. McEliece. The generalized distributive law and free energy minimization. In *Proceedings of the 39th Annual Allerton Conference on Communication, Control, and Computing*, pages 672–681, 2001.

[3] R. G. Almond. *Graphical Belief Modelling*. Chapman & Hall, London, 1995.

[4] T. M. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 1989.

[5] J. Darroch and D. Ratcliff. Generalized iterative scaling for log-linear models. *Ann.Math. Statistics*, 43:1470–1480, 1972.

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

[7] R. Höns. *Estimation of Distribution Algorithms and Minimum Relative Entropy*. PhD thesis, University of Bonn, 2005.

[8] E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev*, 6:620–643, 1957.

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

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

[11] M. I. Jordan, editor. *Learning in Graphical Models*. MIT Press, Cambrigde, 1999.

[12] B. W. Kernighan and S. Lin. An efficient heuristic for partitioning graphs. *Bell. Syst. Techn. Journ.*, 2:291–307, 1970.

[13] S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, 1996.

[14] R. J. McEliece and M. Yildirim. Belief propagation on partially ordered sets. In *Proceedings of the 15th Internatonal Symposium on Mathematical Theory of Networks and Systems (MTNS 2002)*, 2002.

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

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

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

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

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

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

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

[22] M. Pelikan and D. Goldberg. Hierarchical BOA solves Ising spin glasses and MAXSAT. In *Genetic and Evolutionary Computation Conference 2003*, volume 2724 of *Lecture Notes in Computer Science*, pages 1271–1282. Springer, 2003. Also IlliGAL Report No. 2003001.

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

[24] Y. W. Teh and M. Welling. On improving the efficiency of the iterative proportional fitting procedure. In *Proceedings of the International Workshop on Artificial Intelligence and Statistics*, volume 9, 2003.

[25] C. Yanover and Y. Weiss. Finding the M most probable configurations using loopy belief propagation. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

[26] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. Technical Report 2004-040, Mitsubishi Electric Research Laboratories, May 2004.

[27] A. L. Yuille. CCCP algorithms to minimize the Bethe and Kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14(7):1691–1722, 2002.