Optimization by Max-Propagation Using Kikuchi Approximations

Robin Höns[1], Roberto Santana[2], Pedro Larrañaga[3],and Jose A. Lozano[2]

[1]Fraunhofer Institute for Autonomous Intelligent Systems,
53754 Sankt Augustin, Germany
[2]Intelligent Systems Group
Department of Computer Science and Artificial Intelligence
University of the Basque Country
Paseo Manuel de Lardizábal 1, 20080. San Sebastian - Donostia, Spain
[3]Department of Artificial Intelligence, Technical University of Madrid,
28660 Boadilla del Monte, Madrid, Spain.
roberto.santana@ehu.es, pedro.larranaga@fi.upm.es, ja.lozano@ehu.es

Abstract

In this paper we address the problem of using region-based approximations to find the optimal points of a given function. Our approach combines the use of Kikuchi approximations with the application of generalized belief propagation (GBP) using maximization instead of marginalization. The relationship between the fixed points of maximum GBP and the free energy is elucidated. A straightforward connection between the function to be optimized and the Kikuchi approximation (which holds only for maximum GBP, not for marginal GBP) is proven. Later, we show that maximum GBP can be combined with a dynamic programming algorithm to find the most probable configurations of a f graphical model. We then analyze the dynamics of the procedure proposed and show how its different steps can be manipulated to influence the search for optimal solutions.

1 Introduction

Graphical models are one of the most recurred machine learning paradigms used to specify interactions in complex systems in terms of probabilistic dependencies. They represent dependencies among random variables by a graph in which each random variable is a node. The model can be used to solve inference tasks by means of different methods, among them an iterative message-passing algorithm known as belief propagation (BP) [1, 28].

Generalized belief propagation (GBP) [45] can be seen as an extension of BP that works by local message-passing computations on region graphs. GBP allows the solution of problems where conventional BP achieves poor results. BP and GBP have been successfully employed in fields such as computer vision [35], error-correcting codes [21], bioinformatics [43] and statistical physics

[45]. On the other hand, several authors have analyzed different theoretical properties related with BP and GBP. The connection between GBP and region-based approximations of the free energy used in statistical physics [15, 30] has been revealed [45]. Other important issues such as the correctness of BP on loopy graphs [7, 38, 41] and the convergence properties of GBP have been also investigated [10, 45].

BP and GBP can be applied to two main problems: In the first case, the goal is to obtain marginal probabilities for some of the problem variables. In the second case, the objective is to obtain the most probable global state of the problem given the model. The proposal introduced in this paper relies on the use of GBP with maximization (Max-GBP) for the solution of this second problem in the context of combinatorial optimization. Max-GBP is introduced as an alternative for the solution of optimization problems with integer representation.

Two different approaches can be identified in the application of region-based approximations for optimization. One approach [31] considers their application for factorizing and sampling probability distributions that arise during the search for the optimal solutions. In this approach, no message-passing algorithm is used. Instead, the graphical model is learned from samples of the search space, and its region-based factorization is used to sample new points. Although promising results have been achieved with this approach, the algorithms exhibit a number of drawbacks (e.g. expensive sampling methods, poor convergence results for some type of functions, etc.).

The other possibility is to use message-passing algorithms to obtain the most probable global state of a graphical model that assigns probabilities to the solutions according to its function value. Message-passing algorithms do not necessarily perform belief propagation. For example, survey or warning propagation [5, 6] is an alternative. On the other hand, BP (which is conventionally only defined for acyclic models) can be generalized to loopy belief propagation on cyclic graphical models [43]. In this paper we conceive an optimization algorithm based on the use of GBP.

The application of GBP to optimization requires attention to a number of complex issues:

- First, the study of the theoretical properties of the GBP when maximum messages (Max-GBP) are used instead of marginal messages.
- Second, the understanding of how the different aspects or components of GBP (e.g. choice of the regions, type of message passing schedules, algorithm parameters, stop conditions, etc.) can impact on the ultimate optimization goal. Eventually, as we show in this paper, some theoretical properties of GBP can be extended to the Max-GBP case. Additionally, some of the algorithm components can be modified to advance the goal of optimization.

The article is organized as follows. In Sect. 2, the optimization approaches using probabilistic models are presented. Then, Sect.3 introduces the factor graph, the region graph and the free energy. Sect. 4 presents the Max-GBP algorithm and explains the way it is related with the free energy. Sect. 5 presents a dynamic programming schedule to search the M most probable configurations of the model. Sect. 6 presents the loopy max-flow propagation (LMFP) algorithm which combines these approaches. Numerical experiments of this algorithm on the Ising model are given in Sect. 7. Then, in Sect. 8 some previous and related work is reviewed. Finally, Sect. 9 concludes the article and discusses further work.

2 Optimization Based on Probabilistic Models

For our analysis, we classify the optimization methods that use probabilistic models into two classes:

1. Estimation of distribution algorithms (EDAs) and
2. Optimization methods based on inference.

2.1 EDAs

EDAs [18, 25] are evolutionary algorithms that work with a set (or population) of points. Initially, a random sample of points is generated. These points are evaluated using the objective function, and a subset of points is selected based on this evaluation. This causes points with higher function value to have a higher probability to be selected. Then a probabilistic model of the selected solutions is built, and a new set of points is sampled from the model. The process is iterated until the optimum has been found or another termination criterion is fulfilled.

One essential assumption of these algorithms is that it is possible to build a probabilistic model of the search space that can be used to guide the search for the optimum. A key characteristic and crucial step of EDAs is the construction of this probabilistic model. If there is available information about the function (for instance, information about dependencies between the variables) this can be exploited; otherwise the model is learned using the selected population. EDAs have been successfully applied to a wide class of optimization problems [18, 20, 29] and different theoretical frameworks have been used to study their dynamics and convergence properties [8, 46, 47].

2.2 Optimization Methods Based on Inference

In contrast to EDAs, optimization methods based on inference [43, 44] are not iterative methods, in the sense that the probability model is constructed only once. There are no subsequent generations of points. Instead, the idea is to build a probabilistic model from the given function in such a way that the most probable configuration using the model corresponds to the optimum of the problem. Exact or approximate inference can then be employed to find the most probable configuration of the model. When the original independence that represents the interactions between the variables of a problem graph is chordal, a *junction tree* can be constructed [13, 19]. Junction trees are acyclic graphical models. For these models, the most probable configuration can be calculated using dynamic programming [2, 14, 28]. This technique was then extended for calculating the M most probable configurations [26].

2.3 Probabilistic Models That Use Region-based Decompositions

As discussed above, when the independence graph is chordal, the maximization problem can be addressed using efficient methods. However, for many problems the graph that represents the interactions between the variables is not chordal. In these cases, it is possible to obtain a chordal graph by triangulating the original graph. The size of the maximum clique of this graph (or, equivalently, the treewidth [4] of the junction tree) is a critical factor both for EDAs and for optimization methods based on inference. Learning and sampling in EDAs, as well as dynamic

programming methods in inference, have complexity exponential in the size of the maximum clique.

Finding the optimal triangulation of the graph, in terms of the size of the maximum clique of the graph, is an NP-hard problem. Therefore, it is important to consider alternative methods of (approximative) sampling and inference, which work on non-chordal graphs and can be applied to optimization. One of these methods is based on the application of region-based decompositions.

Given an independence graph, a region-based decomposition [37, 42, 45] can be roughly defined as a decomposition of the graph into overlapping subgraphs, the *regions*. As region-based decomposition originates from statistical physics, this has influenced the used terminology. On each region, a local energy is defined, consisting of the local components of the objective function. Also, each region contains a *local belief* for the contained variables. The local notions must be combined to an approximative global counterpart.

The connection made in [45] between GBP and region-based approximations of the free energy used in statistical physics [15, 30] has been a major breakthrough in the advance of belief propagation algorithms. Region-based decompositions have been applied for the definition of propagation methods in a region graph which could also be seen as a generalization of the junction tree. GBP algorithms are used for inferring the local beliefs in the region graph.

3 Factor Graph and Region Graph

In this section we introduce the notation used in the paper and present a formal definition of factor and region graphs.

3.1 Definitions

We revise very shortly the definitions. They are presented in detail in [11, 22, 37, 45].

Let $\mathbf{X} = (X_1, \ldots, X_n)$ denote a vector of discrete random variables. We will use $\mathbf{x} = (x_1, \ldots, x_n)$ to denote an assignment to the variables. S will denote a set of indices in $N = \{1, \ldots, n\}$, and $\mathbf{X}_S$ (respectively $\mathbf{x}_S$) a subset of the variables of $\mathbf{X}$ (respectively a subset of values of $\mathbf{x}$) determined by the indices in S. We will work with positive probability distributions denoted by $p(\mathbf{x})$. Similarly, $p(\mathbf{x}_S)$ will denote the marginal probability distribution for $\mathbf{X}_S$. We use $p(x_i|x_j)$ to denote the conditional probability distribution of X_i given $X_j = x_j$.

An undirected graph $G = (V, E)$ is defined by a non-empty set of vertices V, and a set E of unordered pairs of V called edges. Given a probability distribution $p(\mathbf{x})$, its independence graph $G = (V, E)$ associates one vertex with every variable of $\mathbf{X}$, where two vertices are connected if the corresponding variables are conditionally dependent given the rest of the variables.

Let there be given an additively decomposable fitness function that we try to maximize

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

where the $\mathbf{x}_{s_i}$ are subvectors of $\mathbf{x} = (x_1, \ldots, x_n)$ and $x_i \in \{0, \ldots, k_i\}$. Notice that the class of functions considered are defined on a discrete, countable space. These are bounded functions with finite partitions which serve to represent the wide class of combinatorial and optimization problems we deal with.

This structure can be regarded graphically in a *factor graph*. This is a bipartite graph with n variable nodes and m factor nodes. There is an edge between variable node ν and factor node μ if $x_\nu \in \mathbf{x}_\mu$, i. e. if the subfunction f_μ depends on x_ν.

From this structure, a region graph is defined. A *region* R consists of a set of variables $\mathbf{X}_R$ and a set of factors $\mathbf{f}_R$, such that each variable on which the factors $\mathbf{f}_R$ depend is contained in $\mathbf{X}_R$. Thus, the local part of the fitness

$$f_R(\mathbf{x}_R) := \sum_{f_i \in \mathbf{f}_R} f_i(\mathbf{x}_i) \tag{2}$$

is well-defined. Note that the fitness is to be maximized, whereas in statistical physics (e. g. the Ising problem) the energy is to be minimized; so to define an energy of the region, a minus sign should be added.

A *region graph* is a directed acyclic graph $G = (\mathcal{R}, E_\mathcal{R})$, where $\mathcal{R}$ is a set of regions and $E_\mathcal{R}$ a set of edges, such that

$$(R_1, R_2) \in E_\mathcal{R} \Longrightarrow \mathbf{X}_{R_2} \subset \mathbf{X}_{R_1} \tag{3}$$

For example, a junction tree is a region graph, when its clusters and separators are considered as regions, and there are edges from each cluster to each neighboring separator.

If there is an edge $(R_1, R_2) \in E_\mathcal{R}$, we call R_2 a child of R_1, or alternatively, R_1 a parent of R_2. If there is a path from R_1 to R_2 in the graph, we call R_1 an ancestor and R_2 a descendent. For a region R, the set of all its ancestors is denoted $A(R)$, its descendants $D(R)$. Furthermore we define

$$A'(R) := A(R) \cup \{R\} \tag{4}$$

$$D'(R) := D(R) \cup \{R\} \tag{5}$$

For each region R, we define a *counting number*

$$c_R := 1 - \sum_{R' \in A(R)} c_{R'} \tag{6}$$

This is well-defined, since maximal regions have no ancestors, thus their counting numbers are equal to 1, and one can work his way down the region graph for calculating all c_R.

A region graph is called *valid* if

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

$$\sum_{R \in \mathcal{R}_{x,i}} c_R = 1 \tag{7}$$

and

2. For all subfunction indices $i \in \{1, \ldots, m\}$ the set $\mathcal{R}_{f,i} := \{R \in \mathcal{R} \mid f_i \in \mathbf{f}_R\}$ of all regions R that contain f_i form a connected subgraph with

$$\sum_{R \in \mathcal{R}_{f,i}} c_R = 1 \tag{8}$$

The connectivity of the subgraph, analogous to the junction property of a junction tree, prevents that in different parts of the graph contradictory beliefs evolve. The condition on the counting numbers makes sure that every variable and every subfunction is counted exactly once.

3.2 Free Energy

The GBP algorithm can be used for inference tasks like classification, low-density parity-check codes, turbo codes, etc. A different application is the approximation of the free energy. Let $p_\beta(\mathbf{x})$ be the Boltzmann distribution

$$p_\beta(\mathbf{x}) = \frac{1}{Z} e^{\beta f(\mathbf{x})} \tag{9}$$

for the function $f(\mathbf{x})$ with the inverse temperature β. Z normalizes the distribution.

Given a function $f(\mathbf{x})$, the *tempered free energy* of a distribution $q(\mathbf{x})$ is

$$F_\beta(q) = U_\beta(q) - H(q) \tag{10}$$

where

$$U_\beta(q) = -\beta U(q) = -\beta \sum_{\mathbf{x}} f(\mathbf{x}) q(\mathbf{x}) \tag{11}$$

is the temperate average energy (note the minus sign: we are considering energy here, not fitness) and

$$H(q) = -\sum_{\mathbf{x}} q(\mathbf{x}) \log q(\mathbf{x}) \tag{12}$$

the entropy of $q(\mathbf{x})$.

The tempered *free energy* is a generalization of the free energy to consider the influence of the inverse temperature parameter β. In [11], the effect of varying β within GBP is analyzed. Obviously, maximizing $f(\mathbf{x})$ is equivalent to maximizing $p_\beta(\mathbf{x})$.

It is not difficult to show [24] that for $\beta = 1$ the relative entropy (Kullback Leibler divergence) between a distribution $q(\mathbf{x})$ and the Boltzmann distribution $p_\beta(\mathbf{x})$ is

$$D(q \| p_\beta) = F(q) + \log Z \ . \tag{13}$$

In the analysis that follows we consider that β is fixed to the value of 1 and disregard this term from the free energy expression.

We now assume that on each region R a *local belief* $q_R(\mathbf{x}_R)$ is given. We call $\mathbf{q}_\mathcal{R}$ the set of all these distributions. This allows us to define the *free energy* of a region graph:

$$\begin{aligned} F_\mathcal{R}(\mathbf{q}_\mathcal{R}) &= U_\mathcal{R}(\mathbf{q}_\mathcal{R}) - H_\mathcal{R}(\mathbf{q}_\mathcal{R}) && (14)\\ &= \sum_{R \in \mathcal{R}} c_R U_R(q_R) - \sum_{R \in \mathcal{R}} c_R H_R(q_R) && (15)\\ &= -\sum_{R \in \mathcal{R}} c_R \sum_{\mathbf{x}_R} f_R(\mathbf{x}_R) q_R(\mathbf{x}_R) && (16)\\ &\quad + \sum_{R \in \mathcal{R}} c_R \sum_{\mathbf{x}_R} q_R(\mathbf{x}_R) \log q_R(\mathbf{x}_R) \end{aligned}$$

We call the combination of the local beliefs

$$k(\mathbf{x}) = \prod_{R \in \mathcal{R}} q_R(\mathbf{x}_R)^{c_R} \tag{17}$$

the Kikuchi approximation of the Boltzmann distribution.

Notice that this approximation, unlike the Boltmann distribution, is not normalized. Therefore, our approach clearly departures from common used schemes that look for consistent probability distribution approximations. There are a number of reasons that make this approach worth to consider. Firstly, in several situations, the computation of normalized distributions is infeasible. For instance, the number of terms to be considered in the partition function can grow exponentially with the number of variables. Second, recent applications of region-based decompositions [12, 31] show that they can be used not only for the common tasks of inference but can also be applied for optimization and classification. In these contexts, the decompositions will be used for tasks such as sampling that can be accomplished using unnormalized factorizations.

We emphasize that the proposal presented in this paper builds up on the analysis of the relatioship between the region based approximation and marginal GBP as done in [45]. In addition, we notice that an important issue on this relationship is the determination of the region graph and in particular the selection of the initial regions. This issue seems to has received scarce attention from the research community but it has an important influence in the class optimization problems we consider.

4 Maximum GBP Algorithm

The conventional (marginal) GBP algorithm can be found in [45] and [11]. In this article we present a variant in which the local beliefs $q_R(\mathbf{x}_R)$ are not marginal distributions, but contain the maximal probability.

Let $q_R(\mathbf{x}_R)$ be the maximal probability:

$$q_R(\mathbf{x}_R) = \max_{\mathbf{x}_{N\setminus R}} p_\beta(\mathbf{x}), \tag{18}$$

where $\mathbf{x}_{N\setminus R}$ comprises only those values of $\mathbf{X}_N$ with $\mathbf{X}_R$ fixed to $\mathbf{x}_R$.

In comparison with GBP, the sum is replaced by a max operator. The other difference is that the local beliefs are not normalized. The only constraints are consistency of neighboring beliefs, so for all edges $(P, R) \in E_{\mathcal{R}}$:

$$\max_{\mathbf{x}_{P\setminus R}} q_P(\mathbf{x}_P) = q_R(\mathbf{x}_R) \tag{19}$$

In [45], the parent-to-child GBP algorithm uses an equivalent consistency constraint. We adopt this in the following lemma, again replacing the sum by maximization.

Lemma 1. *In a region graph with no counting number equal to zero, the local region beliefs are consistent if and only if for all* $(P, R) \in E_{\mathcal{R}}$

$$c_R q_R(\mathbf{x}_R) = - \sum_{T \in \mathcal{A}(R)\setminus\mathcal{A}'(P)} c_T \max_{\mathbf{x}_{T\setminus R}} q_T(\mathbf{x}_T) \tag{20}$$

where $\mathcal{A}'(P) := \mathcal{A}(P) \cup \{P\}$.

Proof. We start with the calculation

$$c_R = 1 - \sum_{T \in \mathcal{A}(R)} c_T \tag{21}$$

$$= 1 - c_P - \sum_{T \in \mathcal{A}(P)} c_T - \sum_{T \in \mathcal{A}(R) \setminus \mathcal{A}'(P)} c_T \tag{22}$$

$$= - \sum_{T \in \mathcal{A}(R) \setminus \mathcal{A}'(P)} c_T \tag{23}$$

"⇒": If all regions are consistent, we know that for all ancestors T of R

$$q_R(\mathbf{x}_R) = \max_{\mathbf{x}_{T \setminus R}} q_T(\mathbf{x}_T) \tag{24}$$

Therefore, multiplying this with (23) gives the desired equation.

"⇐": Suppose (20) holds. We show consistence by induction over all nodes, from top to bottom of the region graph. Since region graphs are directed acyclic, the regions can be ordered in such a way that there are no backward edges.

- Induction start: For the first region, there are no edges so consistence is trivial.
- Induction step: Assume that all ancestor regions of R are consistent with each other. That means that all maxima of these sets are the same:

$$\forall T \in \mathcal{A}(R) : \max_{\mathbf{x}_{T \setminus R}} q_T(\mathbf{x}_T) = q_{\text{same}}(\mathbf{x}_R) \tag{25}$$

This is inserted into (20):

$$c_R q_R(\mathbf{x}_R) = -q_{\text{same}}(\mathbf{x}_R) \sum_{T \in \mathcal{A}(R) \setminus \mathcal{A}'(P)} c_T \tag{26}$$

Inserting (23) and dividing by c_R (remember that $c_R \neq 0$ was assumed) shows that $q_R(\mathbf{x}_R) = q_{\text{same}}(\mathbf{x}_R)$.

□

The condition that no counting number is equal to zero is not problematic. If such a region occurs, it can be removed without changing $k(\mathbf{x})$ when all its parents are connected with all its children [27].

Now we introduced the main components of the Max-GBP algorithm that reassembles the GBP introduced in [45] but substituting marginalization by maximization.

Let M_R denote the set of all relevant edges for the region R.

$$M_R := \{(P, C) \in E_{\mathcal{R}} : C \in \mathcal{D}'(R) \wedge P \notin \mathcal{D}'(R)\} \tag{27}$$

It consists of all edges entering into R or one of its descendants from a non-descendent of R. This definition is also analogous to [45].

The fixed point equations for the messages are:

$$m_{P\to R}(\mathbf{x}_R) = \frac{\max_{\mathbf{x}_{P\setminus R}} \prod_{f_i\in\mathbf{f}_P\setminus\mathbf{f}_R} e^{f_i(\mathbf{x}_i)} \prod_{(I,J)\in N(P,R)} m_{I\to J}(\mathbf{x}_J)}{\prod_{(I,J)\in D(P,R)} m_{I\to J}(\mathbf{x}_J)} \tag{28}$$

The message sets of the nominator and denominator are

$$N(P,R) := M_P \setminus (M_P \cap M_R) \tag{29}$$
$$D(P,R) := M_R \setminus \{(M_P \cap M_R) \setminus \{(P,R)\}\} \tag{30}$$

The intersection $M_P \cap M_R$ cancels out in the equation. More specifically, the edge sets are

$$N(P,R) = \{(I,J) \in E_{\mathcal{R}} : J \in \mathcal{D}'(P) \setminus \mathcal{D}'(R) \wedge I \notin \mathcal{D}'(P)\} \tag{31}$$
$$D(P,R) = \{(I,J) \in E_{\mathcal{R}} : (I,J) \neq (P,R) \wedge J \in \mathcal{D}'(R) \wedge I \in \mathcal{D}'(P) \setminus \mathcal{D}'(R)\} \tag{32}$$

Definition 1. *The local beliefs in the Max-GBP parent to child algorithm are*

$$q_R(\mathbf{x}_R) = e^{f_R(\mathbf{x}_R)} \prod_{(P,C)\in M_R} m_{P\to C}(\mathbf{x}_C) \tag{33}$$

4.1 Max-GBP and free energy

In this section, the relationship between the fixed points of Max-GBP belief propagation and the free energy that uses the same region graph is investigated. We will begin by proving an important property of Max-GBP.

While in most BP algorithms we have, at each step of the algorithm, a factorization that is proportional to the original distribution (as illustrated by the study of message passing algorithms in terms of tree-like reparameterizations [16, 38, 39]), in Max-GPB this proportionality is even stronger, and at each step the Kikuchi approximation is equal to the initial Kikuchi approximation. We consider this a surprising result, that is not fulfilled by other BP algorithms.

This property is directly connected to the function $f(\mathbf{x})$ to optimize. This allows to replace fitness function evaluation by evaluation within the region graph. To prove this theorem we need the following lemma.

Lemma 2. *In a valid region graph, for each edge* $(P,C) \in E_{\mathcal{R}}$

$$\sum_{R:(P,C)\in M_R} c_R = 0 \tag{34}$$

Proof. We use the definition of M_R and the counting numbers of P and C.

$$\sum_{R:(P,C)\in M_R} c_R = \sum_{R:C\in\mathcal{D}'(R)\wedge P\notin\mathcal{D}'(R)} c_R \tag{35}$$

$$= c_C + \sum_{R\in\mathcal{A}(C)\setminus\mathcal{A}'(P)} c_R \tag{36}$$

$$= 1 - \sum_{R\in\mathcal{A}(C)} c_R + \sum_{R\in\mathcal{A}(C)\setminus\mathcal{A}'(P)} c_R \tag{37}$$

$$= 1 - \sum_{R\in\mathcal{A}(C)\cap\mathcal{A}'(P)} c_R \tag{38}$$

$$= 1 - \sum_{R\in\mathcal{A}'(P)} c_R \tag{39}$$

$$= 1 - c_P - \sum_{R\in\mathcal{A}(P)} c_R \tag{40}$$

$$= 0 \tag{41}$$

□

Theorem 3. *In Max-GBP (as well as in conventional GBP without normalization) the Kikuchi approximation for a valid region graph is independent of the messages:*

$$k(\mathbf{x}) = e^{f(\mathbf{x})} \tag{42}$$

Proof.

$$k(\mathbf{x}) = \prod_R q_R(\mathbf{x}_R)^{c_R} \tag{43}$$

$$= \prod_R \left(e^{f_R(\mathbf{x}_R)} \prod_{(P,C)\in M_R} m_{P\to C}(\mathbf{x}_C) \right)^{c_R} \tag{44}$$

$$= \prod_R e^{c_R f_R(\mathbf{x}_R)} \prod_R \left(\prod_{(P,C)\in M_R} m_{P\to C}(\mathbf{x}_C) \right)^{c_R} \tag{45}$$

Since the region graph is valid, the first factor is equal to $e^{f(\mathbf{x})}$. In the second part, the products can be rearranged, yielding

$$k(\mathbf{x}) = e^{f(\mathbf{x})} \prod_{(P,C)\in E_\mathcal{R}} \prod_{R:(P,C)\in M_R} m_{P\to C}(\mathbf{x}_C)^{c_R} \tag{46}$$

$$= e^{f(\mathbf{x})} \prod_{(P,C)\in E_\mathcal{R}} m_{P\to C}(\mathbf{x}_C)^{\sum_{R:(P,C)\in M_R} c_R} \tag{47}$$

$$= e^{f(\mathbf{x})} \tag{48}$$

because of Lemma (2). □

From Theorem 3 the following corollary constraining the values taken by the free energy $F(q)$ is derived:

Corollary 4. *At every iteration of Max-GBP, the free energy of the Kikuchi approximation is zero.*

Proof.

$$F(q) = U(q) - H(q) \tag{49}$$
$$= -\sum_{\mathbf{x}} f(\mathbf{x})q(\mathbf{x}) + \sum_{\mathbf{x}} q(\mathbf{x}) \log q(\mathbf{x}) \tag{50}$$
$$= -\sum_{\mathbf{x}} f(\mathbf{x})e^{f(\mathbf{x})} + \sum_{\mathbf{x}} e^{f(\mathbf{x})} \log e^{f(\mathbf{x})} \tag{51}$$
$$= 0 \tag{52}$$

□

The Kikuchi approximation provides a factorization of the function but, in order to be useful for determining the maximum of the function, the different factors must be consistent. An idea is to derive an expression for the stationary points of the constrained free energy and compare these values with the fixed points of the Max-GBP. Unfortunately, the analysis of the derivative for the constrained free energy is cumbersome. Some of the results presented in this paper (e.g. Lemma 1) can be used for this purpose.

We follow an alternative path. We prove that at a fixed point of Max-GBP, the consistency constraints are satisfied, and this fact, together with Corollary 4 determines the relationship between GBP and the constrained free energy.

Theorem 5. *At a fixed point of Max-GBP (when messages have converged), the consistency constraints are satisfied.*

Proof. Let P be parent of R, we will prove that in a fixed point of the algorithm we have:

$$\hat{q}_R(\mathbf{x}_R) = \max_{\mathbf{x}_{P \setminus R}} \hat{q}_P(\mathbf{x}_P) \tag{53}$$

In order to see that, we depart from $\hat{q}_P(\mathbf{x}_P)$:

$$\max_{\mathbf{x}_{P \setminus R}} \hat{q}(\mathbf{x}_P) = \max_{\mathbf{x}_{P \setminus R}} e^{f_P(\mathbf{x}_P)} \prod_{(S,T) \in M_P} m_{S \to T}(\mathbf{x}_T) \tag{54}$$
$$= \max_{\mathbf{x}_{P \setminus R}} e^{f_P(\mathbf{x}_P)} \prod_{(S,T) \in M_P \setminus (M_P \cap M_R)} m_{S \to T}(\mathbf{x}_T) \prod_{(S,T) \in M_P \cap M_R} m_{S \to T}(\mathbf{x}_T)$$
$$= \max_{\mathbf{x}_{P \setminus R}} e^{f_P(\mathbf{x}_P)} \prod_{(S,T) \in N(P,R)} m_{S \to T}(\mathbf{x}_T) \prod_{(S,T) \in M_P \cap M_R} m_{S \to T}(\mathbf{x}_T) \tag{55}$$

Now we depart from $\hat{q}_R(\mathbf{x}_R)$:

$$\hat{q}_R(\mathbf{x}_R) = e^{f_R(\mathbf{x}_R)} \prod_{(S,T)\in M_R} m_{S\to T}(\mathbf{x}_T) \tag{56}$$

$$= \prod_{f_i\in\mathbf{f}_R} e^{f_i(\mathbf{x}_i)} \prod_{(U,V)\in M_R\setminus(M_P\cap M_R)} m_{U\to V}(\mathbf{x}_V) \prod_{(U,V)\in M_P\cap M_R} m_{U\to V}(\mathbf{x}_V) \tag{57}$$

$$= \prod_{f_i\in\mathbf{f}_R} e^{f_i(\mathbf{x}_i)} m_{P\to R}(\mathbf{x}_R) \prod_{(U,V)\in M_R\setminus\{(M_P\cap M_R)\setminus\{(P,R)\}\}} m_{U\to V}(\mathbf{x}_V) \tag{58}$$

$$\prod_{(U,V)\in M_P\cap M_R} m_{U\to V}(\mathbf{x}_V)$$

$$= \prod_{f_i\in\mathbf{f}_R} e^{f_i(\mathbf{x}_i)} m_{P\to R}(\mathbf{x}_R) \prod_{(U,V)\in D(P,R)} m_{U\to V}(\mathbf{x}_V) \prod_{(U,V)\in M_P\cap M_R} m_{U\to V}(\mathbf{x}_V) \tag{59}$$

Now taking into account (55) and (59), in order to see that $\hat{q}_R(\mathbf{x}_R) = \max_{\mathbf{x}_{P\setminus R}} \hat{q}_P(\mathbf{x}_P)$ we need to prove that:

$$\max_{\mathbf{x}_{P\setminus R}} e^{f_P(\mathbf{x}_P)} \prod_{(S,T)\in N(P,R)} m_{S\to T}(\mathbf{x}_T) = \prod_{f_i\in\mathbf{f}_R} e^{f_i(\mathbf{x}_i)} m_{P\to R}(\mathbf{x}_R) \prod_{(U,V)\in D(P,R)} m_{U\to V}(\mathbf{x}_V)$$

but this result is clear in view of the definition of the message $m_{P\to R}(\mathbf{x}_R)$, see (28).

□

4.2 Max-GBP update scheme and construction of candidate maximal points

Max-GBP works by iterating (28). In iteration τ, the update equation is

$$m^{\tau,\mathrm{upd}}_{P\to R}(\mathbf{x}_R) = \frac{\max_{\mathbf{x}_P\setminus\mathbf{x}_R} \prod_{f_i\in\mathbf{f}_P\setminus\mathbf{f}_R} e^{f_i(\mathbf{x}_i)} \prod_{(I,J)\in N(P,R)} m^{\tau-1}_{I\to J}(\mathbf{x}_J)}{\prod_{(I,J)\in D(P,R)} m^{\tau,\mathrm{upd}}_{I\to J}(\mathbf{x}_J)} \tag{60}$$

This sequential update scheme is proposed in [45]. The messages can be ordered in such a way that it is well-defined.

It is computationally favorable to use a damping technique: The variables are updated to a value between the old and new ones. Since the messages are multiplied with each other, *geometrical damping* [36] is plausible:

$$m^{\tau}_{P\to R}(\mathbf{x}_R) = m^{\tau-1}_{P\to R}(\mathbf{x}_R)^{1-\alpha} m^{\tau,\mathrm{upd}}_{P\to R}(\mathbf{x}_R)^{\alpha} \tag{61}$$

where α is the *damping factor* with $0 < \alpha < 1$.

When the message passing algorithm has converged, this means that all regions are locally consistent. This fact does not necessarily imply that globally consistency can be achieved [40]. The problem arises when there are regions for which the maximal belief is reached at multiple

configurations. In this situation it is possible to construct a frustrated circle. Therefore, if the maxima in the single nodes are not unique, it might be impossible to combine them to a consistent maximum [38].

The existence of ties within nodes is an element of difficulty for other belief propagation algorithms like the tree reweighted belief propagation (TRBP) [39]. TRBP is a variant of BP that differs in the message update equations. The algorithm is closely related to linear programming relaxations [3]. Recently, different alternatives have been proposed to deal with cases where there are ties in the TRBP beliefs [17, 23].

Sometimes the Max-GBP algorithm does not converge. And even if it does, the result might not be globally consistent, as just pointed out. Also, we would like to construct points during the run. In these cases, the regions are not necessarily consistent, and the beliefs could be contradictory.

To construct the maximal points, we define an ordering of the maximal regions (those without parents). Let

$$\mathcal{R}_M = \{R_{M,1}, \ldots, R_{M,r}\} \subset \mathcal{R} \tag{62}$$

be this ordering. The variables of each region are separated into those which have appeared in a previous region and the new ones:

$$\mathbf{X}_{R_{M,i}}^{\text{old}} := \mathbf{X}_{R_{M,i}} \cap \left(\bigcup_{j=1}^{i-1} \mathbf{X}_{R_{M,j}} \right) \tag{63}$$

$$\mathbf{X}_{R_{M,i}}^{\text{new}} := \mathbf{X}_{R_{M,i}} \setminus \mathbf{X}_{R_{M,i}}^{\text{old}} \tag{64}$$

In practice, this is very similar to defining a factorization system with these regions, like in [11]. For example, the regions of a tetravariate or pentavariate factorization [24, 11] of a 2-D grid could be ordered top-left to bottom-right.

Now the current estimate of the maximum $\hat{\mathbf{x}}$ is constructed by setting in each region $R_{M,i}$ the variables $\mathbf{X}_{R_{M,i}}^{\text{new}}$ to the values with maximal belief, given the values of $\mathbf{X}_{R_{M,i}}^{\text{old}}$ chosen in the previous steps:

$$\hat{\mathbf{x}}_{R_{M,i}}^{\text{new}} = \underset{\mathbf{x}_{R_{M,i}}^{\text{old}} = \hat{\mathbf{x}}_{R_{M,i}}^{\text{old}}}{\operatorname{argmax}} \; q_{R_{M,i}}(\mathbf{x}_{R_{M,i}}) \tag{65}$$

Sometimes these estimations during the run of Max-GBP are better than the estimate at the end of the algorithm. It is therefore sensible to save the best configuration found which could provide the optimal or a close suboptimal solution.

The whole algorithm is recapitulated in the pseudocode of Alg. 1.

5 The M Most Probable Configurations

Nilsson [26] has devised an algorithm for finding the most probable configurations in a junction tree. He used max-messages and dynamic programming. His algorithm for finding a maximum can be understood as the special case of the above algorithm for cycle-free graphical models. In this case, the result is exact.

Nilsson [26] proposed two schedules for finding the subsequent maxima. We now present the first proposed schedule.

Algorithm 1: **Max-GBP – Maximum Generalized Belief Propagation**

```
1  Let there be given a region graph and an ordering of the maximal regions (62)
2  Choose a damping factor α
3  τ ⇐ 1
4  do {
5     Update messages using (60) and (61)
6     for i = 1 to r do {
7        Set variables of x̂_{R_{M,i}} using (65)
8     }
9     if k(x̂) > k(x_max)
10       x_max ⇐ x̂
11    τ ⇐ τ + 1
12 } until messages converged or τ = τ_max
```

We denote the M maximal configurations by $\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M$. After finding $\mathbf{x}^1 = (x_1^1, x_2^1, \ldots, x_n^1)$, the domain $\mathcal{H}^1 = \{0,1\}^n$ is split into n subdomains:

$$\begin{aligned}
\mathcal{H}_1^1 &= \{\mathbf{x} \in \mathcal{H}^1 \mid x_1 \neq x_1^1\} \\
\mathcal{H}_2^1 &= \{\mathbf{x} \in \mathcal{H}^1 \mid x_1 = x_1^1 \wedge x_2 \neq x_2^1\} \\
\mathcal{H}_3^1 &= \{\mathbf{x} \in \mathcal{H}^1 \mid x_1 = x_1^1 \wedge x_2 = x_2^1 \wedge x_3 \neq x_3^1\} \\
&\vdots \\
\mathcal{H}_i^1 &= \{\mathbf{x} \in \mathcal{H}^1 \mid \forall j < i \ : x_j = x_j^1 \wedge x_i \neq x_i^1\} \\
\mathcal{H}_n^1 &= \{\mathbf{x} \in \mathcal{H}^1 \mid \forall j < n \ : x_j = x_j^1 \wedge x_n \neq x_n^1\}
\end{aligned}$$

This definition ensures that

$$\mathcal{H}_1^1 \dot{\cup} \mathcal{H}_2^1 \dot{\cup} \ldots \dot{\cup} \mathcal{H}_n^1 = \mathcal{H}^1 \setminus \{\mathbf{x}^1\} \tag{66}$$

so the domain which contains the second best configuration is the disjunct union of these subdomains.

The conditions of the $\mathcal{H}_i^1$ can be included as evidence into the runs. In practice, the Boltzmann distribution potentials which do not match the conditions are simply set to zero.

Then, in each of the subsets $\mathcal{H}_i^1$ the maximal configuration is found using the algorithm. We call the found optima $\mathbf{x}^{1,i}$. If we want to find the M best configurations, we only need to save a list of the best M of these.

Then

$$\mathbf{x}^2 = \max_i \mathbf{x}^{1,i} \tag{67}$$

We also set

$$\mathcal{H}^2 = \mathcal{H}_i^1 \tag{68}$$

Now the algorithm can be iterated: Starting with $m = 2$, we separate $\mathcal{H}^m$ in subdomains with

$$\mathcal{H}_i^m = \{\mathbf{x} \in \mathcal{H}^m \mid \forall j < i \ : x_j = x_j^m \ \wedge \ x_i \neq x_i^m\} \tag{69}$$

and find the best configurations $\mathbf{x}^{m,i}$ in all these sets. The best ones of these are ordered into the list of M best values that we save.

Then the next best value in the list is identified as $\mathbf{x}^{m+1}$, the corresponding domain is split up, and so forth.

The second schedule proposed in [26] is similar, only that the ordering of the bits is changed in order to comply with the junction tree structure. This allows to economize message passing steps within the junction tree clusters in which the given evidence is identical. For the loopy Max-GBP algorithm such an ordering is not possible; therefore we will not use this approach here.

6 Optimization Algorithm Based on the M Most Probable Configurations

We employ Nilsson's scheme with Max-GBP. It requires a list of maximal configurations. We call this list $L = \{\mathbf{x}^1, \ldots, \mathbf{x}^M\}$. The list is always kept sorted by fitness. Each new found maximum is sorted into this list, if its fitness is high enough.

Sometimes, especially when Max-GBP does not converge and finds a suboptimal maximum, it happens that a later found optimum has a better fitness than a previous one (with less given evidence). For the exact algorithm on the junction tree [26] this cannot happen. We handle this phenomenon by sorting each new optimum into L and allowing it to supersede the previous one.

Each configuration in the list can be marked *to do* or *done*. In every cycle the highest configuration in the list that is marked *to do* is taken as the starting point of Nilsson's scheme to generate new maxima and then marked *done*. This is iterated until all points in the list are marked *done*.

Furthermore, in each configuration $\mathbf{x}^i$ the number of bits that are already fixed is saved as ϕ^i. Children of this configuration must only be generated for the bits that are not yet fixed. This allows saving some Max-GBP runs.

The algorithm is given completely in the pseudocode of Alg. 2.

In [44], a different scheme is proposed which is computationally superior to the one of Nilsson [26]. In the future we will compare the behavior of this scheme to the one of Alg. 2.

7 Experiments

In the experiments we study the dynamics of the introduced algorithm and compare its performance with other methods used to optimize functions by sampling the search space. First, we introduce the Ising model which is the benchmark problem used in our experiments.

7.1 The Ising Problem

The generalized Ising model is described by the energy functional (Hamiltonian)

$$H = -\sum_{i<j\in L} J_{ij}\sigma_i\sigma_j - \sum_{i\in L} h_i\sigma_i \tag{70}$$

Algorithm 2: **LMFP – Loopy Max-Flow Propagation**

```
1  Find first maximum x^1 with Max-GBP
2  Put x^1 in L, marked to do and φ^1 = 0
3  while L contains configuration marked to do do {
4     Let x^i be the highest such configuration in L
5     for j = φ^i + 1 to n do {
6        Generate H_j^i using (69)
7        Insert evidence for H_j^i and find maximum x^new by Max-GBP
8        If f(x^new) is high enough, sort it into L with φ^new = j and marked to do
9     }
10    If x^i is still in L, mark x^i done
11 }
```

where L is the set of sites called a lattice. Each spin variable σ_i at site $i \in L$ either takes the value 1 or -1. One specific choice of values for the spin variables is called a configuration. The constants J_{ij} are the interaction coefficients. In our experiments we take $h_i = 0,\ \forall i \in L$. The ground state is the configuration with minimum energy. We address the problem of maximizing $f(\mathbf{x}) = -H(\mathbf{x})$.

We use a set of 30 random instances with J_{ij} chosen uniformly within $[-1, 1]$. There are ten instances each on a 7×7, 10×10, and 15×15 2-D lattice.[1]

The type of region graph used for them is the one depicted in Figure 1. The graph shown in the figure corresponds to a pentavariate factorization of a 4×4 grid. In [11] this type of graphs was shown to be superior to the conventional tetravariate grid.

7.2 Study of the Dynamics of the Maximum GBP

In the first experiment, we illustrate the behavior of the Max-GBP algorithm with one example. An instance defined on a 7×7 grid is used. We evaluate the value of the fitness achieved by the Max-GBP algorithm at each iteration. The results are shown in Figure 2. In this case, a cyclic behavior can be appreciated in the values of the best configuration found. This might be due to the fact that messages are contradictory, or that they are sent around in circles, which produces cycles well recognizable in the figure. However, the algorithm finds and saves the best solution early during the Max-GBP run. The best solution is then given as the output of the algorithm.

7.3 Behavior of the Algorithm for Finding the M Most Probable Configurations

Now, we study the performance of the algorithm for finding the M most probable configurations and the fitness of the solutions found with this method. For a 10×10 instance, Figure 3 shows the

[1]The same benchmark instances have been used in [11]. They are available for download at http://www.hoens.net/robin/ising

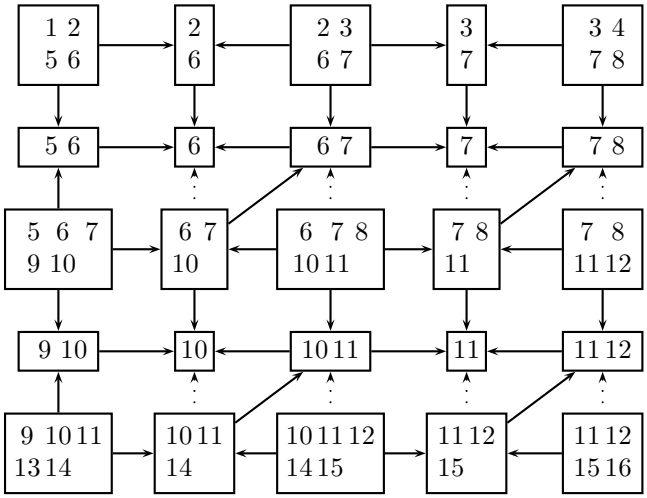

Figure 1: The region graph for Kikuchi, pentavariate factorization.

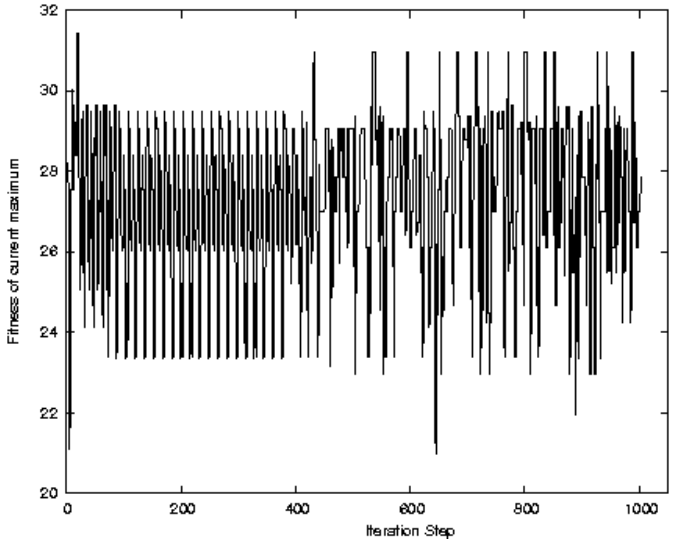

Figure 2: Development of fitness during a run of Max-GBP which does not converge.

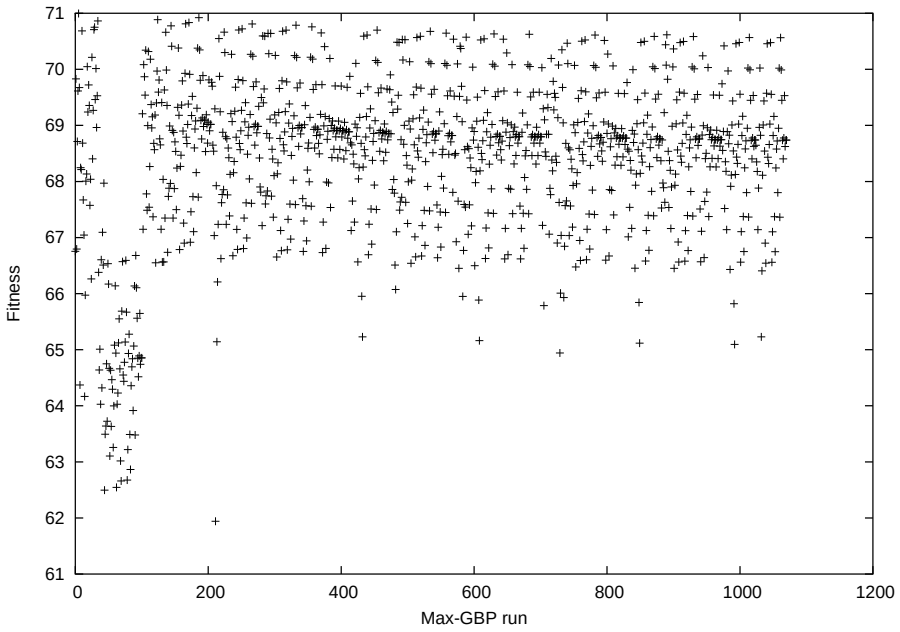

Figure 3: The best configurations found during all the runs of the algorithm for instance 9 of size 10×10.

results of all the Max-GBP runs invoked during the search for the most probable configurations. It can be seen that the best solutions are almost always found in the earliest stage of the algorithm. Figure 4 (top) shows the fitness of the 25 best configurations obtained by the algorithm.

7.4 Convergence of the Algorithm for all the Instances

The goal of the following experiments is an exhaustive investigation of the performance of the Max-GBP algorithm for all the instances. We calculate the three best configurations found by LMFP and compare them to the known optimum. The calculation of the three best configurations is also useful to study the way the optimum is found by tracking which Max-GBP call produces the optimal solution. The results are shown in Table 1. The first column in the table represents the dimension of the instance, in the second column is the number of the instance. Then the global optimum of the instance is given. The following three columns respectively represent the best three configurations found by Alg. 2.

Some information about the runs is given in the following columns. First, it is stated whether the optimum was found by LMFP, then whether both optima were found (remember that the Ising problem is bit-flip symmetric), and finally, whether the optimum was found in the first run or in one of the subsequent runs of Max-GBP.

It can be seen that for the smallest instances the method works very well. Almost all of them are easily solved. It can be also appreciated that for the 10×10 instances the optimum is found only twice in the first call to GBP. Therefore, calculating more than one maximal configuration can help to find alternative ways that lead to the optimum.

For the 15×15 instances, the picture is very different. Here the global optimum is found in four of the ten instances. On these large instances, the Max-GBP algorithm does not converge. Therefore we cannot expect to find a global optimum using the local optima of the regions, because the regions are probably contradictory. In these cases the technique for extracting the optimum from the contradictory region graph described by (65) is vital. It helps to find the optimum for four of the ten cases tried. A better (maybe randomized) technique for this could help the algorithm to scale better.

We have conducted new experiments for the largest set of Ising instances (15×15) given in Table 1. In these experiments we have set the numbers of iterations for Max-GBP to 2000 (The value used in the previous experiments was 1000). We have analyzed the 10 most probable configurations output by the algorithm for each of the instances. The best solutions correspond to those shown in Table 1. This means, that at least for these ten problems, an increase in the number of interactions does not produce an improvement in the algorithm. This might be explained by the fact that the solutions with best fitness are attained at relatively early steps of Max-GBP similar to the case of the 10×10 instances as illustrated by Figure 2.

Another simple trick is to increase M. For instance, in the next section, we run the algorithm with $M = 25$ and find the solution of instance 8 of size 15×15, which was not solved in Table 1.

7.5 Comparison with Other Sampling Methods

The Max-GBP method can be compared with marginal GBP, as done in [11]. The normal method usually can produce a distribution which with very high probability samples the optimum (if β of the Boltzmann distribution is set high enough) for the same problems where Max-GBP converges

Table 1: Result of LMFP on example instances of the Ising problem. Given is first the size of the grid, the number of the instance, and the global maximum of the fitness. Then the values of the best three configurations found by LMFP with $M = 3$ are given. The last three columns state whether the optimum was found, whether both optima were found, and whether an optimum was found in the first run of Max-GBP (without evidence).

inst.	Opt.	1st	2nd	3rd	fnd	both	1st run
$n = 7 \times 7$							
1	34.6649	34.6649	34.6649	34.6396	yes	yes	yes
2	35.0256	35.0256	35.0256	34.9603	yes	yes	no
3	35.4166	35.4166	35.4166	35.4068	yes	yes	yes
4	33.7376	33.7376	33.7376	33.4830	yes	yes	yes
5	37.3632	37.3632	37.3632	37.1808	yes	yes	yes
6	31.4087	31.4087	31.3160	31.2355	yes	no	no
7	34.4746	34.4746	34.4746	34.0303	yes	yes	yes
8	34.5140	34.5140	34.5140	34.4294	yes	yes	yes
9	32.8847	32.8847	32.8847	32.8216	yes	yes	yes
10	35.5478	35.5478	35.5478	35.4491	yes	yes	yes
$n = 10 \times 10$							
1	65.9971	65.9971	65.9971	65.9392	yes	yes	no
2	73.3590	73.3590	73.3590	73.3015	yes	yes	yes
3	72.6994	72.6994	72.6994	72.6655	yes	yes	yes
4	72.5166	72.5166	72.5166	72.4789	yes	yes	no
5	73.5374	73.5374	73.5374	73.5255	yes	yes	no
6	72.7964	72.7964	72.7964	72.7766	yes	yes	no
7	73.9768	73.9768	73.9768	73.8602	yes	yes	no
8	70.7322	70.7322	70.7322	70.6717	yes	yes	no
9	70.9965	70.7513	70.7073	70.6756	no	no	no
10	76.2122	76.2122	76.2122	76.1339	yes	yes	no
$n = 15 \times 15$							
1	165.426	165.426	165.358	165.306	yes	no	no
2	170.999	168.561	168.491	168.466	no	no	no
3	167.659	167.659	167.515	167.459	yes	no	no
4	164.473	159.776	159.690	159.576	no	no	no
5	168.540	164.560	164.539	164.381	no	no	no
6	168.824	168.824	168.768	168.688	yes	no	no
7	169.818	169.818	169.763	169.559	yes	no	no
8	169.024	165.165	165.065	165.055	no	no	no
9	166.844	161.460	161.401	161.274	no	no	no
10	162.118	159.276	159.131	159.107	no	no	no

easily. In difficult instances, the conventional GBP produces a low probability of the optimum, whereas Max-GBP does not converge and finds suboptimal solutions.

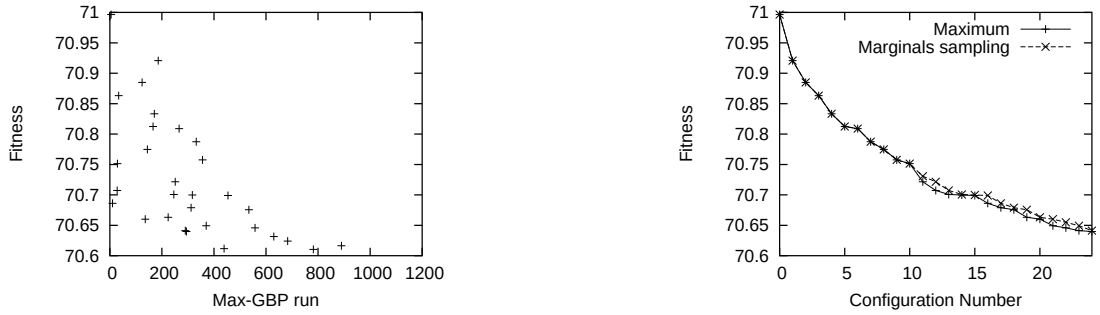

Figure 4: Left: Extract of Figure 3. Right: The 25 best configurations for instance 9 of size 10×10. Found by the maximum algorithm or by sampling 100000 times using marginal GBP with $\beta = 5$ (fitness axis cropped to same scale as left-hand picture).

For marginal GBP, when the maximum is known, its probability p gives us an idea about the sample size which is required for sampling the maximum with a high probability (about $3p^{-1}$ [11]). We compare the performance of the Max-GBP algorithm with marginal GBP.

In Figure 4 (right), we have also included the results of marginal sampling ($\beta = 5$) using a sample size of 100000. It can be observed that the result is almost identical. Sampling found one solution (Number 11 in the ordering of the 25 best configurations) that Max-GBP has missed. Here probably Max-GBP has lost an important branch.

The Ising problem is symmetrical, so actually each configuration should appear twice. It does not. The reason for this is that the very first run of Max-GBP (without any evidence) performs very badly (fitness: 66.7555). Then, 100 runs are performed with 100 pieces of evidence.

The first one of these contains the evidence $x_1 = 0$ (since $x_1 = 1$ in the first maximum) and finds a point with fitness 69.8297. All the other 99 contain the evidence $x_1 = 1$ (among others). Many of these and their subsequent children have fitness higher than 69.8297, so the whole branch $x_1 = 0$ drops out of the top 25 list. This way, this half of the solutions is not found. For the Ising problem, this behavior has no serious implications. For other problems, it might be a good idea to memorize these branches with very little bits fixed so far in an additional to-do list.

Figure 5 shows the result for a 15×15 instance. This particular instance is very difficult for marginal GBP: With $\beta = 5$, the probability of the optimum in the sampling distribution is only $2.72045 \cdot 10^{-7}$. Therefore, it is no surprise that with 10^6 samples the maximum is not found. In this instance, Max-GBP performs really better than the sampling technique.

It should also be noted, however, that the complete run of Max-GBP takes a few hours, whereas sampling a million values from the distribution is done in a few minutes. But here again, we see that Max-GBP finds the best values in the beginning of the algorithm. Maybe it is even possible to recognize bad runs beforehand, skip them and thus speed up the run.

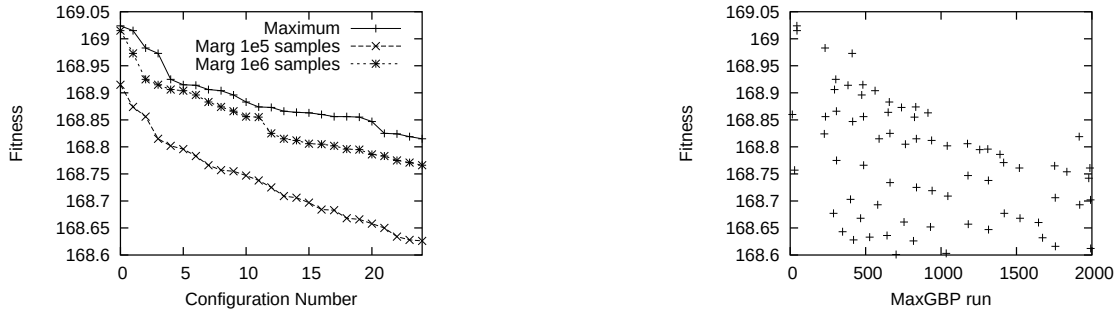

Figure 5: Left:The 25 best configurations of instance 8 of size 15×15, found by the maximum algorithm or by sampling $100,000$ or $1,000,000$ times using marginal GBP with $\beta = 5$. Right: The results of the first $2,000$ (of $3,203$) runs of Max-GBP (fitness axis cropped to same scale as left-hand picture).

8 Related Work

The work presented in this paper is related to a number of previous approaches used for optimization.

In [34], Nilsson's algorithm was used to calculate the most probable configurations in the context of optimization by EDAs. The algorithm was applied to obtain the most probable configurations of the univariate marginal model, and models based on trees and polytrees. For the univariate model no BP is needed. For trees and polytrees BP is guaranteed to converge to the correct solution. The results presented show that EDAs that combine probabilistic logic sampling (PLS) [9] with the computation of the most probable configuration were more efficient in terms of function evaluations than the EDA that use only PLS.

In [31], the Kikuchi approximation constructed from a clique-based decomposition of an independence graph is employed as the probabilistic model of an EDA. Clique-based decompositions are a particular type of region decompositions that satisfy a number of Markovian and decomposability properties [32]. Clique-based decompositions can be obtained from previous information about the problem or learned from data. In [11], region-based approximations were used for estimating and sampling the probability in the context of EDAs. The local beliefs of a region graph are calculated using GBP, and then a factorization is used for sampling promising points in the search space. In contrast to the work presented in [31], the sampling method employed avoids the use of the more expensive Gibbs sampling algorithm. Additionally, the important role of the inverse temperature β in optimization is analyzed.

Propagation algorithms have also been employed to find the optimal solutions of constraint problems. In [5, 6], warning and survey propagation is introduced for the solution of the satisfiability problem (SAT). Warning and survey propagation belongs to a new type of propagation algorithms that intend to find satisfiable assignments to a set of clauses. The algorithm uses factor graphs to represent the structure of the problem and organizes the passing of messages from variables nodes to factor nodes and vice versa. Although these message passing algorithms do not use region based decompositions, their purpose is also to find optimal solutions. The relationship

between warning and survey propagation algorithms and sum-propagation algorithms is discussed in [5].

A common problem of EDAs that use sampling based on factorizations or Gibbs sampling is that the most probable configurations have an exponentially small probability. In this case, Monte Carlo sampling is not a favorable technique because to hit the optimum, a large number of configurations needs to be visited before. The results presented in this paper show that methods for calculating the most probable configurations can be combined with region-based approximations to obtain more efficient algorithms.

Our work is also related with results obtained in the application of approximate inference algorithms in optimization problems [43, 44]. In [43], the best max-marginal fit algorithm (BMMF) for calculating the most probable configurations, based on the use of loopy belief propagation is presented. BMMF is based on a dynamic programming method computationally more efficient than Nilsson's method. For the problems investigated, BMMF was able to find better most probable configurations than those obtained by Gibbs sampling and another method based on a greedy optimization of the posterior probability.

Our approach is similar to the work presented in [43, 44], but there are a number of differences:

1. We have focused on the analysis of the GBP dynamics and its relationship with the search for the optimum. To this regard, we have shown that optimal solutions can be found not only when the GBP algorithm converges, but also obtained during the GBP run.

2. We have also addressed the question of the choice for the region-based decompositions used for the solution of an optimization problem by means of approximate inference. This topic is not addressed in [43]. Our work can also be seen as a different application of the type or region-based decompositions presented in [11] and tested in the context of EDAs.

9 Conclusions and Further Work

We have introduced in this paper a new optimization algorithm based on the use of Kikuchi approximations, a maximum variant of GBP, and a method for finding the most probable configurations. We have shown that the search for optimal solutions can be inserted in the dynamics of the GBP algorithm. Furthermore, we have proved that in order to identify the best solutions found during GBP, it is not necessary to evaluate the configurations. The Kikuchi approximation can be used to evaluate the fitness function instead. This result is very important, particularly for the optimization of functions whose evaluation is costly in terms of time or resources.

Even for instances where GBP does not converge, the method introduced achieves a quite good list of the best instances.

9.1 Further Work

The work presented in this paper can be extended in different directions.

1. In this paper we have used a rather simplistic approach to organize the search for the most probable configurations. The search scheme requires $O(Mn)$ runs of Max-GBP to find M optima. Although our analysis of the use of max-propagation algorithm in optimization has been focussed on the study of GBP, it is expected that the runtime requirements of

Max-GBP could diminish by using a more efficient algorithm for finding the most probable configurations, like the algorithm introduced in [43].

2. Several schemes are conceivable for finding the maximal configuration of an unconverged (therefore contradictory) region graph. For example, using several different orderings of the maximal regions could broaden the search scope. Also, more sophisticated methods for solving the contradictions between the regions can be conceived.

3. The role of β in Max-GBP should be investigated. We have found that the results become worse if β is set to 5 or 10. It appears that a high β is bad for the dynamics of Max-GBP. Therefore, it might even be favorable to use a $\beta < 1$.

4. There exist optimization problems whose structure is partially or totally unknown. We envision that the Max-GBP algorithm could be modified to address these situations. There exist different alternatives to deal with this type of problems:

   - One possible alternative is the construction, in a step previous to the optimization, of a model of the function. These type of models can be learned from data [33], and could be used instead of the original energy function to find the optimum.
   - Prior information about the function, expressed in terms of approximate marginal distributions of the Boltzmann distribution corresponding to the fitness function could be incorporated to the algorithm by means of constraints represented by Lagrange multipliers. This is a promising area of research that would permit the combination of exact information about the function with approximate information obtained from samples of the search space or previous knowledge about the problem structure.

5. Preliminary further experiments have shown that the choice of the region-based decomposition influences the performance of the optimization algorithm. The conception of an automatic method for selecting region-based decompositions would contribute to a generalization of the algorithm.

10 Acknowledgements

Robin Höns's stay in San Sebastián was supported by a doctorand scholarship of the German academic exchange service (DAAD). Authors thank to Heinz Mühlenbein for useful comments on the paper and to Vladimir Kolmogorov for his comments on tree-reweighted max-product message passing algorithms. The contribution to this work of Pedro Larrañaga, José A. Lozano and Roberto Santana was supported by the SAIOTEK-Autoinmune (II) 2006 and Etortek research projects from the Basque Government. It has been also supported by the Spanish Ministerio de Ciencia y Tecnología under grant TIN 2005-03824.

References

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

[2] U. Bertelè and F. Brioschi. *Nonserial Dynamic Programming*. Academic Press, New York, 1972.

[3] D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997.

[4] H. L. Bodlaender. A Tourist Guide Through Treewidth. Technical Report RUU-CS-92-12, Department of Computer Science, Utrecht University, March 1992. Revised: March 1993.

[5] A. Braunstein, M. Mézard, and R. Zecchina. Survey propagation: An algorithm for satisfiability. *Random Structures and Algorithms*, 2005. In press.

[6] A. Braunstein and R. Zecchina. Survey and belief propagation on random K-SAT. *Lecture Notes in Computer Science*, 2919:519–528, 2004.

[7] W. T. Freeman and Y. Weiss. On the optimality of solutions of the max-product belief propagation algorithm in arbitrary graphs. *IEEE Transactions on Information Theory*, 47(2):723–735, 2001.

[8] C. González, J. A. Lozano, and P. Larrañaga. Analyzing the PBIL algorithm by means of discrete dynamical systems. *Complex Systems*, 12(4):465–479, 2001.

[9] M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence*, 2:317–324, 1988.

[10] T. Heskes. On the uniqueness of belief propagation fixed points. *Neural Computation*, 16:2379–2413, 2004.

[11] R. Höns. *Estimation of Distribution Algorithms and Minimum Relative Entropy*. Doctoral dissertation, University of Bonn, 2005. URL http://hss.ulb.uni-bonn.de/diss_online/math_nat_fak/2006/hoens_robin.

[12] A. Jakulin and I. Bratko. Testing the significance of attribute interactions. In *Proceedings of the 21th Conference on Machine Learning (ICML-2004)*, pages 409–416, Banff, Canada, 2004. ACM Press.

[13] 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.

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

[15] R. Kikuchi. A theory of cooperative phenomena. *Physical Review*, 81(6):988–1003, 1951.

[16] V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.

[17] V. Kolmogorov and C. Rother. Comparison of Energy Minimization Algorithms for Highly Connected Graphs. Technical Report MSR-TR-2006-19, Microsoft Research, Redmond, WA, February 2006.

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

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

[20] J. A. Lozano, P. Larrañaga, I. Inza, and E. Bengoetxea, editors. *Towards a New Evolutionary Computation: Advances on Estimation of Distribution Algorithms.* Springer-Verlag, 2006.

[21] R. J. McEliece, D. MacKay, and J. Cheng. Turbo decoding as an instance of Pearl's belief propagation algorithm. *IEEE Journal on Selected Areas in Communication*, 16:140–152, 1998.

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

[23] T. Meltzer, C. Yanover, and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *Tenth IEEE International Conference on Computer Vision*, pages 428–435, 2005.

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

[25] 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, *Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, Berlin, 1996. Springer Verlag. LNCS 1141.

[26] D. Nilsson. An efficient algorithm for finding the M most probable configurations in probabilistic expert systems. *Statistics and Computing*, 8:159–173, 1998.

[27] P. Pakzad and V. Anantharam. Estimation and marginalization using Kikuchi approximation methods. *Neural Computation*, 17(8):1836–1876, 2005.

[28] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann, San Mateo, California, 1988.

[29] M. Pelikan, K. Sastry, and E. Cantú-Paz, editors. *Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications.* Studies in Computational Intelligence. Springer, 2006.

[30] A. Pelizzola. Exactness of the cluster variation method and factorization of the equilibrium probability for the wako-saitô-muñoz-eaton model of protein folding. *Journal of Statistical Mechanics: Theory and Experiment*, 2005(11):P11010, 2005.

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

[32] R. Santana, P. Larrañaga, and J. A. Lozano. Properties of Kikuchi Approximations Constructed From Clique Based Decompositions. Technical Report EHU-KZAA-IK-2/05, Department of Computer Science and Artificial Intelligence, University of the Basque Country, April 2005.

[33] S. Shakya, J. McCall, and D. Brown. Using a Markov network model in a univariate EDA: An empirical cost-benefit analysis. In *Proceedings of Genetic and Evolutionary Computation Conference (GECCO 2005)*, pages 727–734, Washington, D.C., USA, 2005. ACM.

[34] M. R. Soto. *A Singled Connected Factorized Distribution Algorithm and its Cost of Evaluation*. Doctoral dissertation, University of Havana, July 2003. In Spanish.

[35] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, 2003.

[36] Y. W. Teh. *Bethe Free Energy and Contrastive Divergence Approximations for Undirected Graphical Models*. Doctoral dissertation, University of Toronto, 2003.

[37] 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.

[38] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. Tree consistency and bounds on the performance of the max-product algorithm and its generalizations. *Statistics and Computing*, 14:143–166, 2004.

[39] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on trees: message-passing and linear programming. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005.

[40] M. J. Wainwright and M. I. Jordan. Graphical Models, Exponential Families, and Variational Inference. Technical Report 649, Department of Statistics, University of California, Berkeley, September 2003.

[41] Y. Weiss. Correctness of local probability propagation in graphical models with loops. *Neural Computation*, 12:1–41, 2000.

[42] M. Welling. On the choice of regions for generalized belief propagation. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence (UAI-2004)*, pages 585–592, Banff, Canada, 2004. Morgan Kaufmann Publishers.

[43] C. Yanover and Y. Weiss. Approximate inference and protein-folding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1457–1464. MIT Press, Cambridge, MA, 2003.

[44] 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.

[45] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.

[46] Q. Zhang. On stability of fixed points of limit models of univariate marginal distribution algorithm and factorized distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 8(1):80–93, 2004.

[47] Q. Zhang and H. Mühlenbein. On the convergence of a class of estimation of distribution algorithms. *IEEE Transactions on Evolutionary Computation*, 8(2):127–136, 2004.