# Tree-Based Reparameterization Framework for Analysis of Sum-Product and Related Algorithms

Martin J. Wainwright, Tommi S. Jaakkola, and Alan S. Willsky, *Fellow, IEEE*

***Abstract*—We present a tree-based reparameterization (TRP) framework that provides a new conceptual view of a large class of algorithms for computing approximate marginals in graphs with cycles. This class includes the *belief propagation* (BP) or *sum-product* algorithm as well as variations and extensions of BP. Algorithms in this class can be formulated as a sequence of reparameterization updates, each of which entails refactorizing a portion of the distribution corresponding to an acyclic subgraph (i.e., a tree, or more generally, a hypertree). The ultimate goal is to obtain an alternative but equivalent factorization using functions that represent (exact or approximate) marginal distributions on cliques of the graph. Our framework highlights an important property of the sum-product algorithm and the larger class of reparameterization algorithms: the original distribution on the graph with cycles is not changed. The perspective of tree-based updates gives rise to a simple and intuitive characterization of the fixed points in terms of tree consistency. We develop interpretations of these results in terms of information geometry. The invariance of the distribution, in conjunction with the fixed-point characterization, enables us to derive an exact expression for the difference between the true marginals on an arbitrary graph with cycles, and the approximations provided by belief propagation. More broadly, our analysis applies to any algorithm that minimizes the Bethe free energy. We also develop bounds on the approximation error, which illuminate the conditions that govern their accuracy. Finally, we show how the reparameterization perspective extends naturally to generalizations of BP (e.g., Kikuchi approximations and variants) via the notion of hypertree reparameterization.**

***Index Terms*—Approximate inference, belief propagation (BP), Bethe/Kikuchi free energies, convex duality, factor graphs, graphical models, hypergraphs, information geometry, iterative decoding, junction tree, Markov random fields, sum-product algorithm.**

Manuscript received August 23, 2001; revised June 2, 2002. This work was supported in part by ODDR&E MURI under Grant DAAD19-00-1-0466 through the Army Research Office, by the Air Force Office of Scientific Research under Grant F49620-00-1-0362, and the Office of Naval Research under Grant N00014-00-1-0089. The work of M. J. Wainwright was also supported in part by a 1967 Fellowship from the Natural Sciences and Engineering Research Council of Canada. This work appeared originally as MIT-LIDS technical report P-2510 in May 2001. The material in this paper was presented in part at the Trieste Workshop on Statistical Physics, Coding, and Inference, Trieste, Italy, May 2001; and at the IEEE International Symposium on Information Theory, Lausanne, Switzerland, June/July 2002.

M. Wainwright is with the Department of Electrical Engineering and Computer Science, University of California, Berkeley, Berkeley, CA 94720 USA (e-mail: martinw@eecs.berkeley.edu).

T. Jaakkola and A. Willsky are with the Department of Electrical Engineering and Computer Science, the Massachusetts Institute of Technology (MIT), Cambridge, MA 02139 USA (e-mail: tommi@ai.mit.edu; willsky@mit.edu).

Communicated by A. Kavčić, Associate Editor for Detection and Estimation.
Digital Object Identifier 10.1109/TIT.2003.810642

## I. Introduction

PROBABILITY distributions defined by graphs arise in a variety of fields, including coding theory, e.g., [5], [6], artificial intelligence, e.g., [1], [7], statistical physics [8], as well as image processing and computer vision, e.g., [9]. Given a graphical model, one important problem is computing marginal distributions of variables at each node of the graph. For acyclic graphs (i.e., trees), standard and highly efficient algorithms exist for this task. In contrast, exact solutions are prohibitively complex for more general graphs of any substantial size [10]. As a result, there has been considerable interest and effort aimed at developing approximate inference algorithms for large graphs with cycles.

The *belief propagation* (BP) algorithm [11], [3], [1], also known as the *sum-product algorithm*, e.g., [12], [13], [2], [6], is one important method for computing approximate marginals. The interest in this algorithm has been fueled in part by its use in fields such as artificial intelligence and computer vision, e.g., [14], [9], and also by the success of turbo codes and other graphical codes, for which the decoding algorithm is a particular instantiation of belief propagation, e.g., [5], [2], [6]. While there are various equivalent forms for belief propagation [1], the best known formulation, which we refer to here as the BP algorithm, entails the exchange of statistical information among neighboring nodes via message passing. If the graph is a tree, the resulting algorithm can be shown to produce exact solutions in a finite number of iterations. The message-passing formulation is thus equivalent to other techniques for optimal inference on trees, some of which involve more global and efficient computational procedures. On the other hand, if the graph contains cycles, then it is the local message-passing algorithm that is most generally applicable. It is well known that the resulting algorithm may not converge; moreover, when it does converge, the quality of the resulting approximations varies substantially.

Recent work has yielded some insight into the dynamics and convergence properties of BP. For example, several researchers [15]–[17], [11] have analyzed the single-cycle case, where belief propagation can be reformulated as a matrix powering method. For Gaussian processes on arbitrary graphs, two groups [18], [19], using independent methods, have shown that when BP converges, then the conditional means are exact but the error covariances are generally incorrect. For the special case of graphs corresponding to turbo codes, Richardson [13] developed a geometric approach, through which he was able to establish the existence of fixed points, and give conditions for their stability. More recently, Yedidia *et al.* [3], [33] showed that BP can be viewed as performing a constrained minimization of the so-called Bethe free energy associated with the graphical

distribution,[1] which inspired other researchers (e.g., [22], [23]) to develop more sophisticated algorithms for minimizing the Bethe free energy. Yedidia *et al.* also proposed extensions to BP based on cluster variational methods [24]; in subsequent work, various researchers, e.g., [25], [4] have studied and explored such extensions. Tatikonda and Jordan [26] derived conditions for convergence of BP based on the unwrapped computation tree and links to Gibbs measures in statistical physics. These advances notwithstanding, much remains to be understood about the behavior of this algorithm, and more generally about other (perhaps superior) approximation algorithms.

This important area constitutes the focus of this paper. In particular, the framework presented in this paper provides a new conceptual view of a large class of iterative algorithms, including BP, as well as variations and extensions. A key idea in graphical models is the representation of a probability distribution as a product of factors, each of which involves variables only at a subset of nodes corresponding to a clique of the graph. Such factorized representations are far from unique, which suggests the goal of seeking a *reparameterization* of the distribution consisting of factors that correspond, either exactly or approximately, to the desired marginal distributions. If the graph is cycle free (e.g., a tree), then there exists a unique reparameterization specified by exact marginal distributions over cliques. Indeed, such a parameterization is the cornerstone of the junction tree representation (e.g., [27], [28]).

For a graph with cycles, on the other hand, exact factorizations exposing these marginals do not generally exist. Nevertheless, it is always possible to reparameterize certain *portions* of any factorized representation—namely, any subset of factors corresponding to a cycle-free subgraph of the original graph. We are thus led to consider iterative reparameterization of different subsets, each corresponding to an acyclic subgraph. As we will show, the synchronous form of BP can be interpreted in exactly this manner, in which each reparameterization takes place over the extremely simple tree consisting of a pair of neighboring nodes. This interpretation also applies to a broader class of updates, in which reparameterization is performed over arbitrary cycle-free subgraphs. As a vehicle for studying the concept of reparameterization, the bulk of this paper will focus on updates over *spanning trees*, which we refer to as tree-based reparameterization (or TRP). However, the class of reparameterization algorithms is broad, including not only synchronous BP, TRP, and other variants thereof, but also various generalizations of BP (e.g., [33]). As a demonstration of this generality, we discuss in Section VI how reparameterization can also be performed on *hypertrees* of a graph, thereby making connections with generalized belief propagation [33].

At one level, just as BP message passing can be reformulated as a particular sequence of reparameterization updates, the more global updates of TRP are equivalent to a schedule for message passing based on spanning trees. We find that tree-based updates often lead to faster convergence, and can converge on problems for which synchronous BP fails. At another level, the reparameterization perspective provides conceptual insight into the nature of belief propagation and related algorithms. In particular, a fact highlighted by reparameterization, yet not obvious from the traditional message-passing viewpoint, is that the overall distribution on the graph with cycles is never altered by such algorithms. Thus, from the perspective of tree-based updates arises a simple and intuitive characterization of BP fixed points, and more broadly, any constrained minimum of the Bethe free energy, as a *tree-consistent reparameterization* of the original distribution. This characterization of the fixed points allows us to analyze the approximation error for an arbitrary graph with cycles.

In the next section, we introduce the background and notation that underlies our development. In the process, we illustrate how distributions over trees (i.e., cycle-free subgraphs) can be reparameterized in terms of local marginal distributions. In Section III, we introduce the notion of TRP for graphs with cycles. In this context, it is convenient to represent distributions in an exponential form using an overcomplete basis. Our choice of an overcomplete basis, though unorthodox, makes the idea of reparameterization more transparent, and easily stated. In this section, we also show an equivalent formulation of BP as a sequence of local reparameterizations. Moreover, we present some experimental results illustrating the benefits of more global tree-based updates, which include a greater range of problems for which convergence is obtained, as well as typically faster convergence.

Section IV contains analysis of geometry of reparameterization updates, as well as the nature of the fixed points. We begin by formalizing the defining characteristic of all reparameterization algorithms—namely, they do not change the distribution on the graph with cycles, but simply yield an alternative factorization. Geometrically, this invariance means that successive iterates are confined to an affine subspace of exponential parameters (i.e., an $e$-flat manifold in terms of information geometry (e.g., [30], [31]). We then show how each TRP update can be viewed as a projection onto an $m$-flat manifold formed by the constraints associated with each tree. We prove that a Pythagorean-type result holds for successive TRP iterates under a cost function $\mathcal{G}$ that is an approximation to the Kullback–Leibler (KL) divergence. This result establishes interesting links between TRP and successive projection algorithms for constrained minimization of Bregman distances (e.g., [32]). The Pythagorean result enables us to show that fixed points of the TRP algorithm satisfy the necessary conditions to be a constrained local minimum of $\mathcal{G}$, thereby enabling us to make contact with the work of Yedidia *et al.* [3], [33]. In particular, the cost function $\mathcal{G}$, while not the same as the Bethe free energy [33] in general, does agree with it on the relevant constraint set. This fact allows us to establish that TRP fixed points coincide with points satisfying the Lagrangian stationary conditions associated with the Bethe problem (i.e., with the fixed points of BP). An important benefit of our formulation is a new and intuitive characterization of these fixed points: in particular, any fixed point of BP/TRP must be consistent, in a suitable sense to be defined, with respect to any singly connected subgraph; and at least one such fixed point of this type is guaranteed to exist. By adapting the invariance and fixed-point characterization to the Gaussian (as opposed to discrete) case, we obtain a short

[1]Several researchers have investigated the utility of Bethe tree approximations for graphical models; we refer the reader to, e.g., [20], [21].

and elementary proof of the exactness of the means when BP or TRP converges.

Next, we turn to analysis of the approximation error arising from application of BP and other algorithms that perform reparameterization. Previous results on this error have been obtained in certain special cases. For a single cycle, Weiss [11] derived a relation between the exact marginals and the BP approximations, and for a binary processes showed how local corrections could be applied to compute the exact marginals. In the context of turbo decoding, Richardson [13] provided a heuristic analysis of the associated error. Despite these encouraging results, a deep and broadly applicable understanding of the approximation error remains a challenging and important problem. Our characterization of the BP fixed points, in conjunction with the invariance property, allows us to contribute to this goal by analyzing the approximation error for arbitrary graphs. In particular, our development in Section V begins with the derivation of an exact relation between the correct marginals and the approximate marginals computed by TRP or BP. More generally, our analysis applies to the error in any approximation given by minimizing the Bethe free energy. We then exploit this exact relation to derive both upper and lower bounds on the approximation error. The interpretation of these bounds provides an understanding of the conditions that govern the performance of approximation techniques based on the Bethe approach.

In Section VI, we demonstrate how the notion of reparameterization also applies to generalizations of BP that operate over higher order clusters of nodes [33], [25], [4]. In particular, we consider analogs of TRP updates that perform reparameterization over hypertrees of the graph, and show how our tree-based analysis for BP extends in a natural way to these more general hypertree-based updates. The paper concludes in Section VII with a summary.

## II. Background

This section provides background necessary for subsequent developments. We begin with the basics of graphical models, including the necessary preliminaries on graph theory. (See [34], [35] for more background on graph theory.) As for graphical models, there are a variety of different formalisms, including directed Bayesian networks [1], factor graphs [6], and Markov random fields [36]. With some caveats,[2] these different representations are essentially equivalent. In this paper, we will make use of the formalism of *Markov random fields*, which are defined by undirected graphs. More details on graphical models can be found in the books [37], [28], [7]. Next we discuss the problem of inference in graphical models, which (for this paper) refers to computing marginal distributions. We discuss methods for exact inference on trees, including the message-passing updates of belief propagation. We also show how exact inference on trees can be interpreted as reparameterization, which motivates our subsequent analysis for graphs with cycles.

[2]For example, although any directed graph can be converted to an undirected graph [1], some structure may be lost in the process.

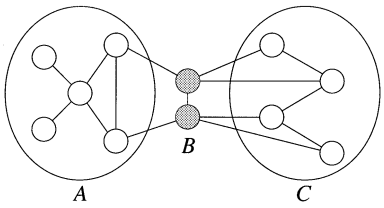



Fig. 1. Illustration of the relation between conditional independence and graph separation. Here the set of nodes $B$ separates $A$ and $C$, so that $\boldsymbol{x}_{A|B}$ and $\boldsymbol{x}_{C|B}$ are conditionally independent.

### A. Basics of Graphical Models

An undirected graph $G = (V, E)$ consists of a set of nodes or vertices $V = \{1, \ldots, N\}$ that are joined by a set of edges $E$. A *cycle* is a sequence of distinct edges forming a path from a node back to itself. A *tree* is a connected graph without any cycles. In order to define an (undirected) graphical model, we place at each node $s \in V$ a random variable $x_s$ taking values in some space. For the bulk of this paper, we focus on random variables that take values in the discrete alphabet $\mathcal{X} = \{0, \ldots, m-1\}$. In this case, the vector $\boldsymbol{x} = \{x_s \mid s \in V\}$ takes values in the set $\mathcal{X}^N$ of all $N$-vectors over $m$ symbols.

Of interest are distributions $p(\boldsymbol{x})$ that are constrained by the graph structure, where the constraints correspond to a set of Markov properties associated with the undirected graph. Let $A$, $B$, and $C$ be subsets of the vertex set $V$. We say that the set $B$ separates $A$ and $C$ if in the modified graph with $B$ removed, there are no paths between nodes in the sets $A$ and $C$ (see Fig. 1). In other words, $B$ is a cutset in the graph $G$.

Let $\boldsymbol{x}_{A|B}$ represent the collection of random variables in $A$ conditioned on those in $B$. The notion of graph separation is at the core of the following definition.

*Definition 1—Markov Random Field:* A random vector $\boldsymbol{x}$ is *Markov* with respect to the graph $G$ if $\boldsymbol{x}_{A|B}$ and $\boldsymbol{x}_{C|B}$ are conditionally independent whenever $B$ separates $A$ and $C$.

A graph strongly constrains the distribution[3] $p(\boldsymbol{x})$ of a vector $\boldsymbol{x}$ that respects its Markov properties. To express these constraints, we introduce the notion of a graph *clique*, which is any fully connected subset of $V$; a clique is maximal if it is not properly contained within any other clique. A *compatibility function* on a clique is a map $\psi_{\mathcal{C}}\colon \mathcal{X}^N \to \mathbb{R}^+$ that depends only on the subvector $\boldsymbol{x}_{\mathcal{C}} = \{x_s \mid s \in \mathcal{C}\}$. The Hammersley–Clifford theorem [38], [36] then guarantees that the distribution of a Markov process on a graph can be expressed as a product of such compatibility functions over the cliques:

*Theorem 1—Hammersley–Clifford:* Let $G$ be a graph with a set of cliques $\boldsymbol{C}$. Suppose that a distribution is formed as a normalized product of compatibility functions over the cliques

$$p(\boldsymbol{x}) = \frac{1}{Z} \prod_{\mathcal{C} \in \boldsymbol{C}} \psi_{\mathcal{C}}(\boldsymbol{x}_{\mathcal{C}}) \tag{1}$$

where $Z \triangleq \sum_{\boldsymbol{x}} \prod_{\mathcal{C} \in \boldsymbol{C}} \psi_{\mathcal{C}}(\boldsymbol{x})$ is the partition function. Then, the underlying process $\boldsymbol{x}$ is Markov with respect to the graph.

[3]Strictly speaking, $p$ is a probability mass function for discrete random variables; however, we will use distribution to mean the same thing.

Conversely, the distribution $p$ of any Markov random field over $G$ that is strictly positive (i.e., $p(\boldsymbol{x}) > 0$ for all $\boldsymbol{x} \in \mathcal{X}^N$) can be represented in this factorized form.

For the bulk[4] of this paper, we shall assume that the maximal cliques of our graph have cardinality two, an assumption which is implicit in the standard message-passing form of BP, as presented in Section II-B2. Given only singleton and pairwise cliques, the clique index set ranges over all edges $(s, t) \in E$, as well as the singleton cliques $\{s\}$. In this case, the compatibility functions $\psi_s$ and $\psi_{st}$ denote real-valued functions of $x_s$ and $(x_s, x_t)$, respectively. With a minor abuse of notation, we will often use the same notation to refer to vectors and matrices, respectively. In particular, for an $m$-state discrete process, the quantity $\psi_{st}$ can be also thought of as an $m \times m$ matrix, where the $(j, k)$ element $\psi_{st;\,jk}$ is equal to the function value of $\psi_{st}$ for $\{x_s = j, x_t = k\}$. Similarly, the single-node functions $\psi_s$ can be thought of as an $m$-vector, where the $j$th component $\psi_{s;\,j}$ equals the value of $\psi_s$ for $\{x_s = j\}$.

### B. Estimation in Graphical Models

In many applications, the random vector $\boldsymbol{x}$ is not observed; given instead are noisy observations $y_s$ of $x_s$ at some (or all) of the nodes, on which basis one would like to draw inferences about $\boldsymbol{x}$. For example, in the context of error-correcting codes (e.g., [2]), the collection $\boldsymbol{y} = \{y_s \,|\, s \in V\}$ represents the bits received from the noisy channel, whereas the vector $\boldsymbol{x}$ represents the transmitted codeword. Similarly, in image processing or computer vision [8], the vector $\boldsymbol{y}$ represents noisy observations of image pixels or features.

One standard inference problem, and that of central interest in this paper, is the computation of the marginal distributions $p(x_s \,|\, \boldsymbol{y})$ for each node $s$. This task, which in this paper will be called *optimal estimation or inference*, is intractable [10], since it requires summations involving exponentially many terms. As indicated previously, for tree-structured graphs, there exist direct algorithms for optimal estimation. For graphs with cycles, suboptimal algorithms (such as BP) are used in an attempt to compute approximations to the desired marginals. In this section, we elaborate on both of these topics.

A remark before proceeding: Note that each individual node forms a singleton clique, so that some of the factors in (1) may involve functions of each individual variable. As a consequence, Bayes' rule implies that the effect of including these measurements—i.e., the transformation from the prior distribution $p(\boldsymbol{x})$ to the conditional distribution $p(\boldsymbol{x} \,|\, \boldsymbol{y})$—is simply to modify the singleton factors of (1). As a result, throughout this paper, we suppress explicit mention of measurements, since the problem of computing marginals for either $p(\boldsymbol{x} \,|\, \boldsymbol{y})$ or $p(\boldsymbol{x})$ are of identical structure and complexity.

*1) Exact Inference on Trees as Reparameterization:* Algorithms for optimal inference on trees have appeared in the literature of various fields, including coding theory [6], artificial intelligence [1], and system theory [39]. In broad overview, such algorithms consist of a recursive series of updates, in which "messages" are passed from node to node.

[4]In Section VI, we shall consider more general Markov random fields that may include higher order cliques.

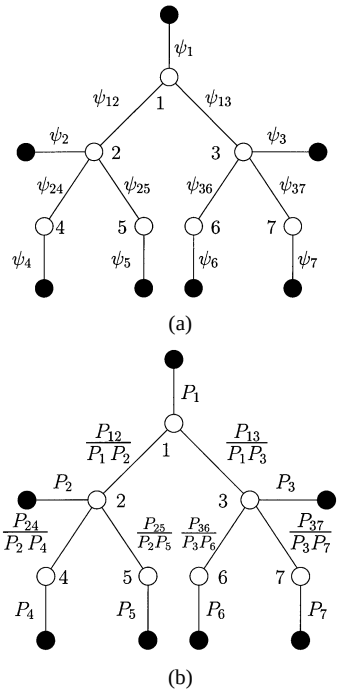



Fig. 2. A simple example of a tree-structured graphical model. (a) Original parameterization of distribution as in (1). (b) Alternative parameterization in terms of marginal distributions $P_s$ and $P_{st}$ as in (2).

The most efficient implementation of these algorithms follows a two-pass form, first sweeping upwards from the leaves to a node designated as the root, and then downwards from the root to leaves, with an overall computational complexity of $\mathcal{O}(m^2 N)$.

It is worth noting that tree inference algorithms can, in principle, be applied to any graph by clustering nodes so as to form a so-called *junction tree* (e.g., [27], [28]). This junction tree procedure is discussed in more detail in Section VI-B. However, in many cases of interest, the aggregated nodes of the junction tree have exponentially large state cardinalities, meaning that applying tree algorithms is prohibitively complex. This explosion in the state cardinality is another demonstration of the intrinsic complexity of exact computations for graphs with cycles.

An important observation that arises from the junction tree perspective (e.g., [27], [28]) is that any exact algorithm for optimal inference on trees actually computes marginal distributions for pairs $(s, t)$ of neighboring nodes. In doing so, it produces an alternative factorization of the distribution $p(\boldsymbol{x})$, namely,

$$p(\boldsymbol{x}) = \prod_{s \in V} P_s(x_s) \prod_{(s,\,t) \in E} \frac{P_{st}(x_s, x_t)}{P_s(x_s) P_t(x_t)} \tag{2}$$

where $P_s(x_s)$ is the marginal distribution of the variable $x_s$, and $P_{st}(x_s, x_t)$ is the joint marginal over $x_s$ and $x_t$. As an illustration, Fig. 2(a) shows a simple example of a tree-structured distribution, specified in terms of compatibility functions $\psi_s$ and $\psi_{st}$, as in the factorization of (1). Fig. 2(b) shows this same tree-structured distribution, now *reparameterized* in terms of the local marginal distributions $P_s$ and $P_{st}$. The representation of (2) can be deduced from a more general factorization result on junction trees (e.g., [28], [27]). More concretely, (2)

can be seen as a symmetrized generalization of the well-known factorization(s) of Markov chains. For example, the variables at the three nodes $\{1, 2, 4\}$ in Fig. 2(b) form a simple Markov chain, meaning that the joint distribution can be written as

$$\begin{aligned} P_{124} &= P_1(P_{2|1})(P_{4|2}) \\ &= P_1(P_{12}/P_1)(P_{24}/P_2) \\ &= P_1P_2P_4(P_{12}/P_1P_2)(P_{24}/P_2P_4) \end{aligned}$$

where the last equality is precisely the form of (2). Note that the final line removes the asymmetry present in the preceding lines (which resulted from beginning the factorization from node 1, as opposed to node 2 or 4).

We thus arrive at an alternative interpretation of exact inference on trees: it entails computing a reparameterized factorization of the distribution $p(\boldsymbol{x})$ that explicitly exposes the local marginal distributions; and also does not require any additional normalization (i.e., with partition function $Z = 1$).

*2) Belief Propagation for Graphs With Cycles:* As we have indicated, the message-passing form of BP, in addition to being exact in application to trees, yields an iterative message-passing algorithm for graphs with cycles. In this subsection, we summarize for future reference the equations governing the BP dynamics. The message passed from node $s$ to node $t$, denoted by $M_{st}$, is an $m$-vector in which element $M_{st;\,k}$ gives its value when $x_t = k$. Let $\Gamma(s) = \{t \in V \,|\, (s, t) \in E\}$ be the set of neighbors of $s$ in $G$. With this notation, the message at iteration $(n + 1)$ is updated based on the messages at the previous iteration $n$ as follows:

$$M_{st;\,k}^{n+1} = \kappa \sum_{j=0}^{m-1} \psi_{st;\,jk}\psi_{s;\,j} \prod_{u\in\Gamma(s)/t} M_{us;\,j}^{n} \tag{3}$$

where $\kappa$ denotes a normalization constant.[5] At any iteration, the "beliefs"—that is, approximations to the marginal distributions—are given by

$$B_{s;\,j}^{n} = \kappa\psi_{s;\,j} \prod_{u\in\Gamma(s)} M_{us;\,j}^{n}. \tag{4}$$

## III. Tree-Based Reparameterization Framework

In this section, we introduce the class of TRP updates. Key to TRP is the concept of a tree-structured subgraph of an arbitrary graph $G$ with cycles—i.e., a tree formed by removing edges from the graph. A *spanning tree* is an acyclic subgraph that connects all the vertices of the original graph. Fig. 3 illustrates these definitions: Fig. 3(a) shows a nearest neighbor grid, whereas Fig.3 (b) illustrates a spanning tree. Of course, Fig.3 (b) is just one example of a spanning tree embedded in the original graph (a). Indeed, a graph generally has a (large) number of spanning trees,[6] and we exploit this fact in our work. Specifically, suppose that $\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}$ (with corresponding edge sets $E^0, \ldots, E^{L-1} \subset E$) is a given set of spanning trees for the graph $G$. Then, for any $i \in \{0, \ldots, L-1\}$, the distribution $p(\boldsymbol{x})$ can be factored as

$$p(\boldsymbol{x}) = p^i(\boldsymbol{x})r^i(\boldsymbol{x}) \tag{5}$$

where $p^i(\boldsymbol{x})$ includes the factors in (1) corresponding to cliques of $\mathcal{T}^i$, and $r^i(\boldsymbol{x})$ absorbs the remaining terms, corresponding to edges in $E\backslash E^i$ removed to form $\mathcal{T}^i$.

Because $\mathcal{T}^i$ is a tree, the reparameterization operation in (2) can be applied to the tree-structured distribution $p^i(\boldsymbol{x})$ in order to obtain an alternative factorization of the distribution $p^i(\boldsymbol{x})$. With reference to the full graph $G$ and distribution $p(\boldsymbol{x})$, this operation simply specifies an alternative choice of compatibility functions that give rise to the same distribution $p(\boldsymbol{x})$. In a subsequent update, using this new set of functions and choosing a *different* tree $\mathcal{T}^j$, we can write $p(\boldsymbol{x}) = p^j(\boldsymbol{x})r^j(\boldsymbol{x})$, where $p^j(\boldsymbol{x})$ includes compatibility functions over cliques in $\mathcal{T}^j$. We can then perform reparameterization for $p^j(\boldsymbol{x})$, and repeat the process, choosing one of the $\mathcal{T}^i$ at each step of the iteration.

The basic steps of this procedure are illustrated for a simple graph in Fig. 4. Fig. 4(a) shows the original parameterization of $p(\boldsymbol{x})$ in terms of compatibility functions $\psi_s$ and $\psi_{st}$, as in (1). A spanning tree, formed by removing edges $(4, 5)$ and $(5, 6)$, is shown in Fig.4(b); that is,

$$r^i(\boldsymbol{x}) = \psi_{45}(x_4, x_5)\,\psi_{56}(x_5, x_6)$$

in this case. The tree distribution $p^i(\boldsymbol{x})$, corresponding to the product of all the other compatibility functions, is reparameterized in terms of marginals $T_s$ and $T_{st}$ computed from the tree $\mathcal{T}^i$, as shown in panel (c). Note that the quantities $\{T_s, T_{st}\}$ are exact marginals for the tree, but represent approximations to the true marginals $\{P_s, P_{st}\}$ of the full graph with cycles. The graph compatibility functions after this first update are shown in panel (d). In a subsequent update, a different tree is chosen over which reparameterization is to be performed.

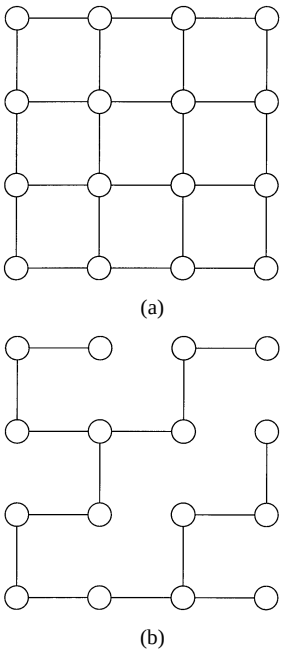


Fig. 3. A graph with cycles has a (typically large) number of spanning trees. (a) Original graph is a nearest neighbor grid. Panel (b) shows one of the 100 352 spanning trees of the graph in (a).

[5]Throughout this paper, we will use $\kappa$ to refer to an arbitrary normalization constant, the definition of which may change from line to line. In all cases, it is easy to determine $\kappa$ by local calculations.

[6]In general, the number of spanning trees can be computed by the matrix-tree theorem (e.g., [34]).

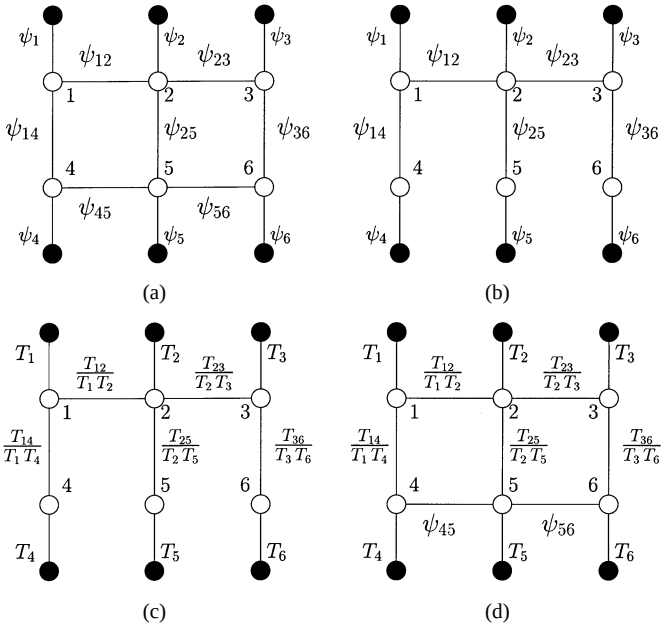



Fig. 4. Illustration of a TRP update. (a) Original parameterization in terms of compatibility functions $\psi_s$ and $\psi_{st}$. (b) Isolate terms corresponding to a tree $T^i$. (c) Reparameterize the tree-structured component $p^i(\boldsymbol{x})$ of the distribution in terms of the tree marginals $\{T_s,\ T_{st}\}$. (d) New parameterization of the full distribution after a single iteration.

At one level, the sequence of updates just described is equivalent to a particular tree-based schedule for BP message passing. In particular, each tree update consists of fixing all messages on edges not in the tree, and updating messages on the tree edges until convergence. However, thinking about reparameterization instead of message passing highlights an important property: each step of the algorithm[7] entails specifying an alternative factorization of the distribution $p(\boldsymbol{x})$, and therefore, leaves the full distribution intact. To formalize this basic idea, in this section, we introduce a particular parameterization of distributions $p(\boldsymbol{x};\ \theta)$, such that iterations of the type just described can be represented as explicit functional updates $\theta^n \mapsto \theta^{n+1}$ on these parameters. We also show that synchronous BP can be interpreted as performing reparameterization updates over especially simple (nonspanning) trees, and we present experimental results illustrating the potential advantages of tree-based updates over synchronous BP.

### A. Exponential Families of Distributions

Central to our work are exponential representations of distributions, which have been studied extensively in statistics and applied probability theory (e.g., [31], [40], [30], [41], [42]). Given an index set $\mathcal{A}$, we consider a collection of potential functions $\boldsymbol{\phi} = \{\phi_\alpha \,|\, \alpha \in \mathcal{A}\}$ associated with the graph $G$. We let $\theta = \{\theta_\alpha \,|\, \alpha \in \mathcal{A}\}$ denote a vector of parameters, and then consider following distribution:

$$p(\boldsymbol{x};\ \theta) = \exp\left\{\sum_\alpha \theta_\alpha \phi_\alpha(\boldsymbol{x}) - \Phi(\theta)\right\} \tag{6a}$$

[7]Here we have described an unrelaxed form of the updates; in the sequel, we present and analyze a suitably relaxed formulation.

$$\Phi(\theta) = \log\left(\sum_{\boldsymbol{x}\in\mathcal{X}^N} \exp\left\{\sum_\alpha \theta_\alpha \phi_\alpha(\boldsymbol{x})\right\}\right). \tag{6b}$$

Here, $\Phi$ is the *log partition function* that serves to normalize the distribution. As the exponential parameter $\theta$ ranges over $\mathbb{R}^{d(\theta)}$ where $d(\theta) = |\mathcal{A}|$, (6) specifies a family of distributions associated with the given graph.

The log partition function $\Phi$ has a number of useful properties that we will exploit in subsequent analysis. In particular, by straightforward calculations, we obtain

$$\frac{\partial \Phi}{\partial \theta_\alpha}(\theta) = \mathbb{E}_\theta[\phi_\alpha] \tag{7a}$$

$$\frac{\partial^2 \Phi}{\partial \theta_\alpha\, \partial \theta_\beta}(\theta) = \operatorname{cov}_\theta\{\phi_\alpha,\ \phi_\beta\} \triangleq \mathbb{E}_\theta[\phi_\alpha\ \phi_\beta] - \mathbb{E}_\theta[\phi_\alpha]\mathbb{E}_\theta[\phi_\beta] \tag{7b}$$

where $\mathbb{E}_\theta$ denotes the expectation taken over $\boldsymbol{x}$ with respect to the distribution $p(\boldsymbol{x};\ \theta)$; that is, $\mathbb{E}_\theta[\phi_\alpha] = \sum_{\boldsymbol{x}} p(\boldsymbol{x};\ \theta)\phi_\alpha(\boldsymbol{x})$. We note that the quantity in (7b) is an element of the Fisher information matrix $(-\mathbb{E}_\theta\{\frac{\partial^2 \log p(\boldsymbol{x};\,\theta)}{\partial \theta^2}\})$. Therefore, the Hessian $\nabla^2\Phi$ is positive semidefinite, so that $\Phi$ is a convex function of $\theta$.

In addition, the exponential parameterization of (6) induces a certain form for the KL divergence [43] that will be useful in the sequel. Given two parameter vectors $\theta$ and $\theta^*$, we denote by $D(\theta\,||\,\theta^*)$ the KL divergence between the distributions $p(\boldsymbol{x};\ \theta)$ and $p(\boldsymbol{x};\ \theta^*)$. This divergence can be written in the following form:

$$D(\theta\,||\,\theta^*) = \sum_\alpha \mathbb{E}_\theta[\phi_\alpha(\boldsymbol{x})]\,(\theta_\alpha - \theta^*_\alpha) + \Phi(\theta^*) - \Phi(\theta). \tag{8}$$

Note that this definition entails a minor abuse of notation, since the divergence applies to distributions $p(\boldsymbol{x};\ \theta)$ and $p(\boldsymbol{x};\ \theta^*)$, and not the parameters $\theta$ and $\theta^*$ themselves.

It remains to specify a choice of functions $\boldsymbol{\phi} = \{\phi_\alpha\}$. Let $s,\ t \in V$ be indexes parameterizing the nodes of the graph, and let the indexes $j,\ k$ run over the $m$ possible states of the discrete random variables. We then take the index set $\mathcal{A}$ to be the set of pairs $(s;\ j)$ or 4-tuples $(st;\ jk)$, and choose the potentials $\phi_\alpha$ as indicator functions for $x$ to take on the indicated value (or values) at the indicated node (or pair of nodes). That is,

$$\phi_\alpha(\boldsymbol{x}) = \delta_{s;\,j}(x_s), \qquad \text{for } \alpha = (s;\ j) \tag{9a}$$

$$\phi_\alpha(\boldsymbol{x}) = \delta_{s;\,j}(x_s)\delta_{t;\,k}(x_t), \qquad \text{for } \alpha = (st;\ jk). \tag{9b}$$

Here, the indicator function $\delta_{s;\,j}(x_s)$ is equal to 1 when node $x_s$ takes the state value $j$, and 0 otherwise. With this choice of $\boldsymbol{\phi}$, the length of $\theta$ is given by

$$d(\theta) = mN + m^2|E| \tag{10}$$

where $|E|$ is the number of edges in $G$. Moreover, note that the exponential representation is related to the compatibility functions (as in, e.g., (1)) via the relation

$$\theta_{s;\,j}\phi_{s;\,j}(x_s) = \log \psi_s(x_s = j)$$

with a similar relation for $\theta_{st;\,jk}$ and $\psi_{st}$.

It is typical to choose a linearly independent collection of functions, which gives rise to a so-called *minimal* representation (e.g., [40]). In this context, the exponential parameterization of

(9) is unorthodox because it is *overcomplete* (i.e., there are affine relations among the functions $\boldsymbol{\phi} = \{\phi_\alpha\}$). As an example, for any edge $(s, t) \in E$, we have the linear dependence

$$\sum_j \delta_{s;\,j}(x_s)\delta_{t;\,k}(x_t) = \delta_{t;\,k}(x_t), \quad \text{for all } k = 0, \ldots, m-1.$$

An important consequence of overcompleteness is the existence of distinct parameter vectors $\theta \neq \theta^*$ that induce the same distribution (i.e., $p(\boldsymbol{x};\ \theta) = p(\boldsymbol{x},\ \theta^*)$). This many-to-one correspondence between parameters and distributions is of paramount importance to our analysis because it permits reparameterization operations that leave the overall distribution unchanged.

### B. Basic Operators

Given a distribution $p(\boldsymbol{x};\ \theta)$ defined by a graph $G$, the quantities that we wish to compute are elements of the $d(\theta)$-dimensional *marginal probability vector* $P = \{P_\alpha \,|\, \alpha \in \mathcal{A}\}$. The elements of this marginal probability vector are given by

$$\begin{aligned} P_{s;\,j} &= p(x_s = j;\ \theta), && \text{for } s \in V,\ j \in \mathcal{X} \\ P_{st;\,jk} &= p(x_s = j,\ x_t = k;\ \theta), && \text{for } (s, t) \in E,\ j, k \in \mathcal{X} \end{aligned}$$

corresponding to the single-node and joint pairwise marginals, respectively. We use $P_s$ to denote the $m$-vector of values $\{P_{s;\,j},\ \ j = 1, \ldots, m\}$, and define the $m^2$-vector $P_{st}$ in an analogous way. Such vectors can be viewed as an alternative set of parameters for a graphical distribution. More precisely, the quantities $P$ and $\theta$ are a dual set of parameters, related via the Legendre transform applied to the log partition function (see [41], [44], [45]).

We will frequently need to consider mappings between these two parameterizations. In particular, the computation of the marginals can be expressed compactly as a map acting on the parameter vector $\theta$

$$P = \Lambda(\theta). \tag{11}$$

Note that the range of $\Lambda$ is a highly constrained set. Its elements correspond to *realizable* marginal vectors, so that we use $\mathrm{MARG}(G)$ to denote the range of $\Lambda$. First of all, any realizable marginal vector must belong to the unit hypercube $[0, 1]^{d(\theta)}$. Second, there are normalization constraints (single-node and joint marginal probabilities must sum to one); and marginalization constraints (pairwise joint distributions, when marginalized, must be consistent with the single node marginals). That is, $\mathrm{MARG}(G)$ is contained within the constraint set $\mathrm{TREE}(G)$, defined as follows:

$$\mathrm{TREE}(G) = \left\{ T \in [0, 1]^{d(\theta)} \;\middle|\; \sum_j T_{s;\,j} = 1,\ \sum_k T_{st;\,jk} = T_{s;\,j} \right\}. \tag{12}$$

Our choice of notation is motivated by the fact that for a tree-structured graph $G = \mathcal{T}$, we have the equivalence $\mathrm{MARG}(\mathcal{T}) = \mathrm{TREE}(\mathcal{T})$. More specifically, given some element $T$ of $\mathrm{TREE}(\mathcal{T})$, the local constraints imposed on $T$ are sufficient to guarantee the existence of a unique distribution $p(\boldsymbol{x};\ \theta)$ such that $T = \Lambda(\theta)$. This fact—that local consistency implies global consistency for trees—is the essence of the junction tree theorem (see, e.g., [28]).

For a graph $G$ with cycles, in contrast, there exist elements of $\mathrm{TREE}(G)$ that cannot be realized as the marginals of any distribution (see [45]), so that $\mathrm{MARG}(G)$ is strictly contained within $\mathrm{TREE}(G)$. This strict containment reflects the fact that for a graph with cycles, local consistency is no longer sufficient to guarantee the existence of a globally consistent distribution.

For a general graph with cycles, of course, the computation of $\Lambda(\theta)$ in (11) is very difficult. Indeed, algorithms like BP and TRP can be formulated as iteratively generating approximations to $\Lambda(\theta)$. To make a sharp distinction from exact marginal vectors $P \in \mathrm{MARG}(G) \subset \mathrm{TREE}(G)$, we use the symbol $T$ to denote such *pseudomarginals*. Moreover, when the graph $G$ is clear from the context, we will simply write MARG and TREE to denote $\mathrm{MARG}(G)$ and $\mathrm{TREE}(G)$, respectively.

We also make use of the following mapping that is defined for any $T \in (0, 1)^{d(\theta)}$:

$$[\Theta(T)]_\alpha = \begin{cases} \log T_{s;\,j}, & \text{if } \alpha = (s;\ j) \in \mathcal{A} \\ \log \dfrac{T_{st;\,jk}}{\left(\sum_j T_{st;\,jk}\right)\left(\sum_k T_{st;\,jk}\right)}, & \text{if } \alpha = (st;\ jk) \in \mathcal{A}. \end{cases} \tag{13}$$

The quantity $\Theta(T)$ can be viewed as an exponential parameter vector that indexes a distribution $p(\boldsymbol{x};\ \Theta(T))$ on the graph $G$. In fact, consider a marginal vector $P \in \mathrm{MARG}(G)$. If $G$ is a tree, then not only is the computation of (11) simple, but we are also guaranteed $\Theta(P)$ indexes the same graphical distribution as that corresponding to the marginal vector $P$; that is,

$$\Lambda(\Theta(P)) = P. \tag{14}$$

This equality is simply a restatement of the factorization of (2) for any tree-structured distribution in terms of its single-node and joint pairwise marginals. However, if $G$ has cycles, then, in general, the marginal distributions of $p(\boldsymbol{x};\ \Theta(P))$ need not agree with the original marginals $P$ (i.e., the equality of (14) does not hold). In fact, determining the exponential parameters corresponding to $P$ for a graph with cycles is as difficult as the computation of $\Lambda(\theta)$ in (11). Thus, the composition of operators $\Lambda \circ \Theta$, mapping one marginal vector to another, is the identity for trees but not for general graphs.

Alternatively, we can consider composing $\Theta$ and $\Lambda$ in the other order

$$\mathcal{R}(\theta) = \big(\Theta \circ \Lambda\big)(\theta) \tag{15}$$

which defines a mapping from one exponential parameter vector to another. For a general graph, the operator $\mathcal{R}$ will alter the distribution (that is, $p(\boldsymbol{x};\ \theta) \neq p(\boldsymbol{x};\ \mathcal{R}(\theta))$). For a tree-structured graph, while $\mathcal{R}$ is not the identity mapping, it does leave the probability distribution unchanged; indeed, applying $\mathcal{R}$ corresponds to shifting from the original parameterization of the tree distribution in terms of $\theta$ to a new exponential parameter $\mathcal{R}(\theta)$ that corresponds directly to the factorization of (2). As a

result, in application to trees, the operator $\mathcal{R}$ is idempotent (i.e., $\mathcal{R} \circ \mathcal{R} = \mathcal{R}$).

### C. TRP Updates

The basic idea of TRP is to perform reparameterization updates on a set of spanning trees in succession. We assume that each edge in the graph belongs to at least on spanning tree in the collection $\{\mathcal{T}^0, \ldots, \mathcal{T}^{L-1}\}$. The update on any given spanning tree $\mathcal{T}^i$ involves only a subset

$$\mathcal{A}^i = \{(s;\ j),\ (st;\ jk) \,|\, s \in V,\ (s,\ t) \in E^i\}$$

of all the elements of $\theta$. To move back and forth between parameter vectors on the full graph and those on spanning tree $\mathcal{T}^i$, we define projection and injection operators

$$\Pi^i(\theta) = \{\theta_\alpha \,|\, \alpha \in \mathcal{A}^i\} \tag{16a}$$

$$\mathcal{I}^i(\Pi^i(\theta)) = \begin{cases} \theta_\alpha, & \text{if } \alpha \in \mathcal{A}^i \\ 0, & \text{if } \alpha \notin \mathcal{A}^i. \end{cases} \tag{16b}$$

We let $\Lambda^i$, $\Theta^i$, and $\mathcal{R}^i$ denote operators analogous to those in (11), (13), and (15), respectively, but as defined for $\mathcal{T}^i$.

Each TRP update acts on the full-dimensional vector $\theta$, but changes only the lower dimensional subvector $\Pi^i(\theta) = \{\theta_\alpha \,|\, \alpha \in \mathcal{A}^i\}$. For this reason, it is convenient to use the underbar notation to define operators of the following type:

$$\underline{\mathcal{R}}^i(\theta) = \mathcal{I}^i(\mathcal{R}^i(\Pi^i(\theta))) \tag{17a}$$

$$\underline{\Lambda}^i(\theta) = \mathcal{I}^i(\Lambda^i(\Pi^i(\theta))). \tag{17b}$$

For instance, $\underline{\Lambda}^i$ projects the exponential parameter vector $\theta$ onto spanning tree $\mathcal{T}^i$, computes the corresponding marginal vector for the distribution $p(\boldsymbol{x};\ \Pi^i(\theta))$ induced on the tree, and then injects back to the higher dimensional space by inserting zeroes for elements of edges not in $\mathcal{T}^i$ (i.e., for indexes $\alpha \in \mathcal{A} \backslash \mathcal{A}^i$). Moreover, analogous to $\mathrm{TREE}$, we define a constraint set $\mathrm{TREE}^i$ by imposing marginalization constraints only for edges in the spanning tree (i.e., as in (12) with $E$ replaced by $E^i$). Note that $\mathrm{TREE}^i \supseteq \mathrm{TREE}$, and since every edge is included in at least one spanning tree, we have that $\cap_i \mathrm{TREE}^i = \mathrm{TREE}$.

Using this notation, the operation of performing tree reparameterization on spanning tree $\mathcal{T}^i$ can be written compactly as transforming a parameter vector $\theta$ into the new vector given by

$$\mathcal{Q}^i(\theta) = \underline{\mathcal{R}}^i(\theta) + \left[I - \mathcal{I}^i \circ \Pi^i\right](\theta) \tag{18a}$$

$$= \theta + \left[\underline{\mathcal{R}}^i(\theta) - \mathcal{I}^i(\Pi^i(\theta))\right] \tag{18b}$$

where $I$ is the identity operator. The two terms in (18a) parallel the decomposition of (5): namely, the operator $\underline{\mathcal{R}}^i$ performs reparameterization of the distribution $p^i(\boldsymbol{x})$, whereas the operator $[I - \mathcal{I}^i \circ \Pi^i]$ corresponds to leaving the residual term $r^i(\boldsymbol{x})$ unchanged. Thus, (18a) is a precise statement of a spanning tree update (as illustrated in Fig. 4), specified in terms of the exponential parameter $\theta$.

Given a parameter vector $\theta$, computing $\mathcal{Q}^i(\theta)$ is straightforward, since it only involves operations on the spanning tree $\mathcal{T}^i$. The TRP algorithm generates a sequence of parameter vectors $\{\theta^n\}$ by successive application of these operators $\mathcal{Q}^i$. The sequence is initialized[8] at $\theta^0$ using the original set of compatibility functions $\{\psi_s\}$ and $\{\psi_{st}\}$ as follows:

$$\theta^0_\alpha = \begin{cases} \log \psi_{s;\,j}, & \text{if } \alpha = (s;\ j) \\ \log \psi_{st;\,jk}, & \text{if } \alpha = (st;\ jk). \end{cases}$$

At each iteration $n$, we choose some spanning tree index $i(n)$ from $\{0, \ldots, L-1\}$, and then update using the operator on spanning tree $\mathcal{T}^i$

$$\theta^{n+1} = \mathcal{Q}^{i(n)}(\theta^n). \tag{19}$$

In the sequel, we also consider a relaxed iteration, involving a step size $\lambda^n \in (0,\ 1]$ for each iteration

$$\theta^{n+1} = \lambda^n \mathcal{Q}^{i(n)}(\theta^n) + (1 - \lambda^n)\theta^n \tag{20}$$

where $\lambda^n = 1$ recovers the unrelaxed version.

The only restriction that we impose on the set of spanning trees is that each edge of the full graph $G$ is included in at least one spanning tree (i.e., $\cup_i \mathcal{A}^i = \mathcal{A}$). It is also necessary to specify an order in which to apply the spanning trees—that is, how to choose the index $i(n)$. A natural choice is the *cyclic ordering*, in which we set $i(n) \equiv n (\mathrm{mod}\, L)$. More generally, any ordering—possibly random—in which each spanning tree occurs infinitely often is acceptable. A variety of possible orderings for successive projection algorithms are discussed in [32].

In certain applications (e.g., graphical codes), some of the compatibility functions may include zeroes that reflect deterministic constraints (e.g., parity checks). In this case, we can either think of the reparameterization updates as taking place over exponential parameters $\theta_\alpha$ in the extended reals $\mathbb{R} \cup \{-\infty\}$, or more naturally, as operating directly on the compatibility functions $\{\psi_s,\ \psi_{st}\}$ themselves.

### D. Belief Propagation as Reparameterization

In this subsection, we show how to reformulate synchronous BP in "message-free" manner as a sequence of local rather than global reparameterization operations. Specifically, in each step, new compatibility functions are determined by performing exact calculations over extremely simple (nonspanning) trees formed of two nodes and the corresponding edge joining them.

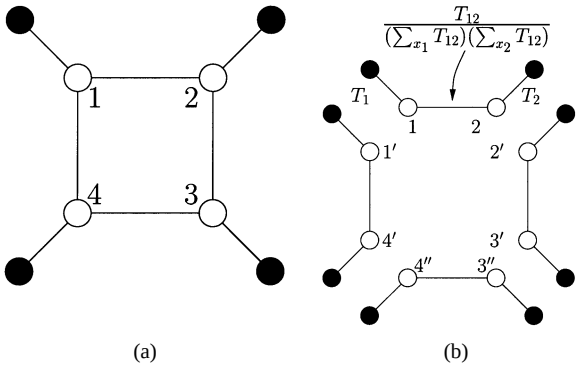
We denote by $M^0_{st}$ the $m$-vector corresponding to the chosen initialization of the messages. This choice is often the vector of all ones, but any initialization with strictly positive components is permissible. The message-free version of BP iteratively updates approximations to the collection of exact marginals $P = \{P_s,\ P_{st}\}$. Initial values of the pseudomarginals $T = \{T_s,\ T_{st}\}$ are determined from the initial messages $M^0_{st}$ and the original compatibility functions of the graphical model as follows:

$$T^0_{s;\,j} = \kappa \psi_{s;\,j} \prod_{u \in \Gamma(s)} M^0_{us;\,j} \tag{21a}$$

$$T^0_{st;\,jk} = \kappa \psi_{st;\,jk} \psi_{s;\,j} \psi_{t;\,k} \prod_{u \in \Gamma(s) \backslash t} M^0_{us;\,j} \prod_{u \in \Gamma(t) \backslash s} M^0_{ut;\,k} \tag{21b}$$

where $\kappa$ denotes a normalization factor.

[8]Other initializations are also possible. More generally, $\theta^0$ can be chosen as any exponential parameter that induces the same distribution as the original compatibility functions $\{\psi_s\}$ and $\{\psi_{st}\}$.




(a) (b)

Fig. 5. (a) Single cycle graph. (b) Two-node trees used for updates in message-free version of belief propagation. Computations are performed exactly on each two-node tree formed by a single edge and the two associated observation potentials as in (22b). The node marginals from each two-node tree are merged via (22a).

At iteration $n$, these pseudomarginals are updated according to the following recursions:

$$T^n_{s;\,j} = \kappa T^{n-1}_{s;\,j} \prod_{t\in\Gamma(s)} \frac{1}{T^{n-1}_{s;\,j}} \sum_{k=0}^{m-1} T^{n-1}_{st;\,jk} \tag{22a}$$

$$T^n_{st;\,jk} = \kappa \frac{T^{n-1}_{st;\,jk}}{\left(\sum_{j=0}^{m-1} T^{n-1}_{st;\,jk}\right)\left(\sum_{k=0}^{m-1} T^{n-1}_{st;\,jk}\right)} T^n_{s;\,j} T^n_{t;\,k}. \tag{22b}$$

The update in (22b) is especially noteworthy: it corresponds to performing optimal estimation on the very simple two-node tree formed by edge $(s, t)$. As an illustration, Fig. 5(b) shows the decomposition of a single-cycle graph into such two-node trees. This simple reparameterization algorithm operates by performing optimal estimation on this set of nonspanning trees, one for each edge in the graph, as in (22b). The single-node marginals from each such tree are merged via (22a).

We now claim that this reparameterization algorithm is equivalent to belief propagation, summarizing the result as follows.

*Proposition 1:* The reparameterization algorithm specified by (21) and (22) is equivalent to the message-passing form of BP given in (3) and (4). In particular, for each iteration $n = 0, 1, \ldots$ and initial message vector $M^0_{st}$, we have the following relations:

$$M^{n+1}_{st;\,k} = \kappa M^0_{st;\,k} \prod_{i=0}^{n} \frac{1}{T^i_{t;\,k}} \sum_{j=0}^{m-1} T^i_{st;\,jk}, \quad \forall (s,t)\in E \tag{23a}$$

$$B^n_{s;\,j} = T^n_{s;\,j}, \qquad \forall s\in V \tag{23b}$$

where $\kappa$ denotes a normalization factor.

*Proof:* These equations can be established by induction on iteration $n$, using the BP equations (3) and (4), and the reparameterization equations (21) and (22). □

### *E. Empirical Comparisons of Local Versus Tree-Based Updates*

Given that a spanning tree reaches every node of the graph, one might expect tree-based updates, such as those of TRP, to have convergence properties superior to those of local updates such as synchronous BP. As stated previously, a single TRP update on a given spanning tree can be performed by fixing all the messages on edges not in tree, and updating messages on edges in the tree until convergence. Such tree-based message updating schedules are used in certain applications of BP like turbo decoding [2], for which there are natural choices of trees over which to perform updates. In this subsection, we provide experimental evidence supporting the claim that tree-based updates can have superior convergence properties for other problems. An interesting but open question raised by these experiments is how to optimize the choice of trees (not necessarily spanning) over which to perform the updates.

*1) Convergence Rates:* In this subsection, we report the results of experiments on the convergence rates of TRP and BP on three graphs: a single cycle with 15 nodes, a $7 \times 7$ nearest neighbor grid, and a larger $40 \times 40$ grid. At first sight, the more global nature of TRP might suggest that each TRP iteration is more complex computationally than the corresponding BP iteration. In fact, the opposite statement is true. Each TRP update corresponds to solve a tree problem exactly, and therefore requires $\mathcal{O}(m^2(N-1))$ operations.[9] In contrast, each iteration of synchronous BP requires $\mathcal{O}(m^2|E|)$ operations, where $|E| \geq N-1$ is the number of edges in the graph. In order to make comparisons fair in terms of actual computation required, the iteration numbers that we report are rescaled in terms of relative cost (i.e., for each graph, TRP iterations are rescaled by the ratio $(N-1)/|E| < 1$). In all cases, we used unrelaxed updates for both BP and TRP.

For each graph, we performed simulations under three conditions: edge potentials that are *repulsive* (i.e., that encourage neighboring nodes to take opposite values); *attractive* (that encourage neighbors to take the same value); and *mixed* (in which some potentials are attractive, while others are repulsive). For each of these experimental conditions, each run involved a random selection of the initial parameter vector $\theta^0$ defining the distribution $p(\boldsymbol{x};\ \theta^0)$. In all experiments reported here, we generated the single-node parameters $\theta_{s;\,j}$ as follows:[10] for each node $s \in V$, sample $a_s \sim \mathcal{N}(0, (0.25)^2)$, and set $[\theta_{s;\,0}\ \theta_{s;\,1}] = [a_s\ -a_s]$. To generate the edge potential components $\theta_{st;\,jk}$, we began by sampling $b_{st} \sim \mathcal{N}(0, 1)$ for each edge $(s, t)$, With $\delta_{jk}$ denoting the Kronecker delta for $j, k$, we set the edge potential components in one of three ways depending on the experimental condition. For the *repulsive condition*, we set $\theta_{st;\,jk} = -(2\delta_{jk} - 1)|b_{st}|$; for the *attractive condition*, $\theta_{st;\,jk} = (2\delta_{jk} - 1)|b_{st}|$; whereas for the *mixed condition*, $\theta_{st;\,jk} = (2\delta_{jk} - 1)b_{st}$.

For each experimental condition, we performed a total of 500 trials for each of the single cycle and $7 \times 7$ grid, comparing the performance of TRP to BP. On any given run, an algorithm was deemed to converge when the mean $L^2$ difference between successive node elements $(\frac{1}{n}\sum_s \|\theta^{n+1}_s - \theta^n_s\|^2)$ reached a threshold of $\epsilon = 1 \times 10^{-16}$. A run in which a given algorithm failed to reach this threshold within 3000 iterations was classified as a failure to converge. In each condition, we report the total number of convergent trials (out of 500); and also the mean number of iterations required to converge, rescaled by the ratio $(N-1)/|E|$ and based only on trials where both TRP and BP converged.

[9]Here we are using the fact that a tree problem can be solved efficiently by a two-pass sweep, where exactly two messages are passed along each edge of the graph.

[10]The notation $\mathcal{N}(0,\ \sigma^2)$ denotes a zero-mean Gaussian with variance $\sigma^2$.

TABLE I
COMPARISON OF CONVERGENCE BEHAVIOR OF TRP VERSUS BP FOR A SINGLE CYCLE OF 15 NODES; AND A $7 \times 7$ GRID. POTENTIALS WERE CHOSEN RANDOMLY IN EACH OF THE THREE CONDITIONS: REPULSIVE POTENTIALS ($R$), ATTRACTIVE POTENTIALS ($A$), MIXED POTENTIALS ($M$). FIRST AND SECOND NUMBERS IN EACH BOX DENOTE THE NUMBER OF CONVERGENT RUNS OUT OF 500; AND THE MEAN NUMBER OF ITERATIONS (RESCALED BY RELATIVE COST AND COMPUTED USING ONLY RUNS WHERE BOTH TRP AND BP CONVERGED)

| Graph | Single 15-cycle | | | | | | $7 \times 7$ grid | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | R | | A | | M | | R | | A | | M | |
| BP | 500 | 23.2 | 500 | 23.4 | 500 | 23.6 | 455 | 62.3 | 457 | 65.8 | 267 | 310.1 |
| TRP | 500 | 8.1 | 500 | 8.0 | 500 | 8.2 | 500 | 30.5 | 500 | 30.8 | 282 | 103.2 |

Table I shows some summary statistics for the two graphs used in these experiments. For the single cycle, we implemented TRP with two spanning trees, whereas we used four spanning trees for the grid. Although both algorithms converged on all trials for the single cycle, the rate of TRP convergence was significantly (roughly three times) faster. For the grid, algorithm behavior depends more on the experimental condition. The repulsive and attractive conditions are relatively easy,[11] though still difficult enough for BP that it failed to converge on roughly 10% of the trials, in contrast to the perfect convergence percentage of TRP. In terms of mean convergence rates, TRP converged more than twice as quickly as BP. The mixed condition is difficult for suitably strong edge potentials on a grid: in this case both algorithms failed to converge on almost half the trials, although TRP converged more frequently than BP. Moreover, on runs where both algorithms converged, the TRP mean rate of convergence was roughly three times faster than BP. Although mean convergence rates were faster, we did find individual problems on the grid for which the version of TRP with four trees converged more slowly than BP. However, one possibility (which we did not take advantage of here) is to optimize the choice of trees in an adaptive manner.

We also examined convergence behavior for the $40 \times 40$ grid with 1600 nodes, using a version of TRP updates over two spanning trees. Fig. 6 provides an illustration of the convergence behavior of the two algorithms. Plotted on a log scale is the $L^2$ distance between the single-node elements of $\theta^n$ and $\theta^*$ at each iteration, where $\theta^*$ is a fixed point common to BP and TRP, versus the iteration number. Again, the TRP iteration are rescaled by their cost relative to BP iterations ($(N-1)/|E|$), which for this large grid is very close to $0.5$. Fig. 6 (a) illustrates a case with repulsive potentials, for which the TRP updates converge quite a bit faster than BP updates. Examples in the attractive condition show similar convergence behavior. Fig. 6 (b) and (c) shows two different examples, each with a mixed set of potentials. The mixed condition on the grid is especially difficult due to the possibility of conflicting or frustrated interactions between the nodes. For the problem in Fig. 6 (b), the two spanning trees used for this particular version of TRP are a good choice, and again lead to faster convergence. The potentials for the problem in Fig. 6 (c), in contrast, cause difficulties for this pair of spanning trees; note the erratic convergence behavior of TRP.

Each TRP update ignores some local interactions corresponding to the edges removed to form the spanning tree. These edges are covered by other spanning trees in the set used; however, it remains an open question how to choose trees so as to maximize the rate of convergence. In this context, one could imagine a hybrid algorithm in which pure synchronous BP iterations are interspersed with iterations over more global structures like trees (not necessarily spanning). The exploration of such issues remains for future research.

*2) Domain of Convergence:* We have also found that tree-based updates can converge for a wider range of potentials than synchronous BP. The simple five-node graph shown in Fig. 7(a) serves to illustrate this phenomenon. We simulated a binary process over a range of potential strengths $\mu$ ranging from $-0.3$ to $-1.0$. Explicitly, for each value of $\mu$, we made a deterministic assignment of the potential for each edge $(s, t)$ of the graph as $\theta_{st;\,jk} = (2\delta_{jk} - 1)\mu$. For each potential strength, we conducted 100 trials, where on each trial the single-node potentials were set randomly by sampling $a_s \sim \mathcal{N}(0, (0.25)^2)$ and setting $[\theta_{s;\,0} \;\; \theta_{s;\,1}] = [a_s \;\; -a_s]$. On any given trial, the convergence of a given algorithm was assessed as in Section III-E1. Plotted in Fig. 7(b) is the percentage of successfully converged trials versus potential strength for TRP and BP. Both algorithms exhibit a type of threshold behavior, in which they converge with 100% success up to a certain potential strength, after which their performance degrades rapidly. However, the tree-based updates extend the effective range of convergence.[12] To be fair, recently proposed alternatives to BP for minimizing the Bethe free energy (e.g., [23], [22]), though they entail greater computational cost than the updates considered here, are guaranteed to converge to a stationary point.

## IV. ANALYSIS OF GEOMETRY AND FIXED POINTS

In this section, we present a number of results related to the geometry and fixed points of reparameterization algorithms like TRP and BP. The defining characteristic of a reparameterization algorithm is that the original distribution is never altered. Accordingly, we begin in Section IV-A with a formal statement of this property in terms of exponential parameters, and then establish its validity for the more general class of relaxed updates. We also develop the geometric interpretation of this result: all iterates are confined to an affine subspace of the exponential parameters (i.e., an $e$-flat manifold in information geometry [30], [40]). Motivated by this geometric view, we show that a TRP update can be viewed as a projection onto the tree constraint set $\mathrm{TREE}^i$. This projection is defined by a particular cost function,

[11] In fact, on a bipartite graph like the nearest neighbor grid, the repulsive and attractive conditions are equivalent.

[12] This result is not dependent on the symmetry of the problem induced by our choice of edge potentials; for instance, the results are similar if edge potentials are perturbed from their nominal strengths by small random quantities.

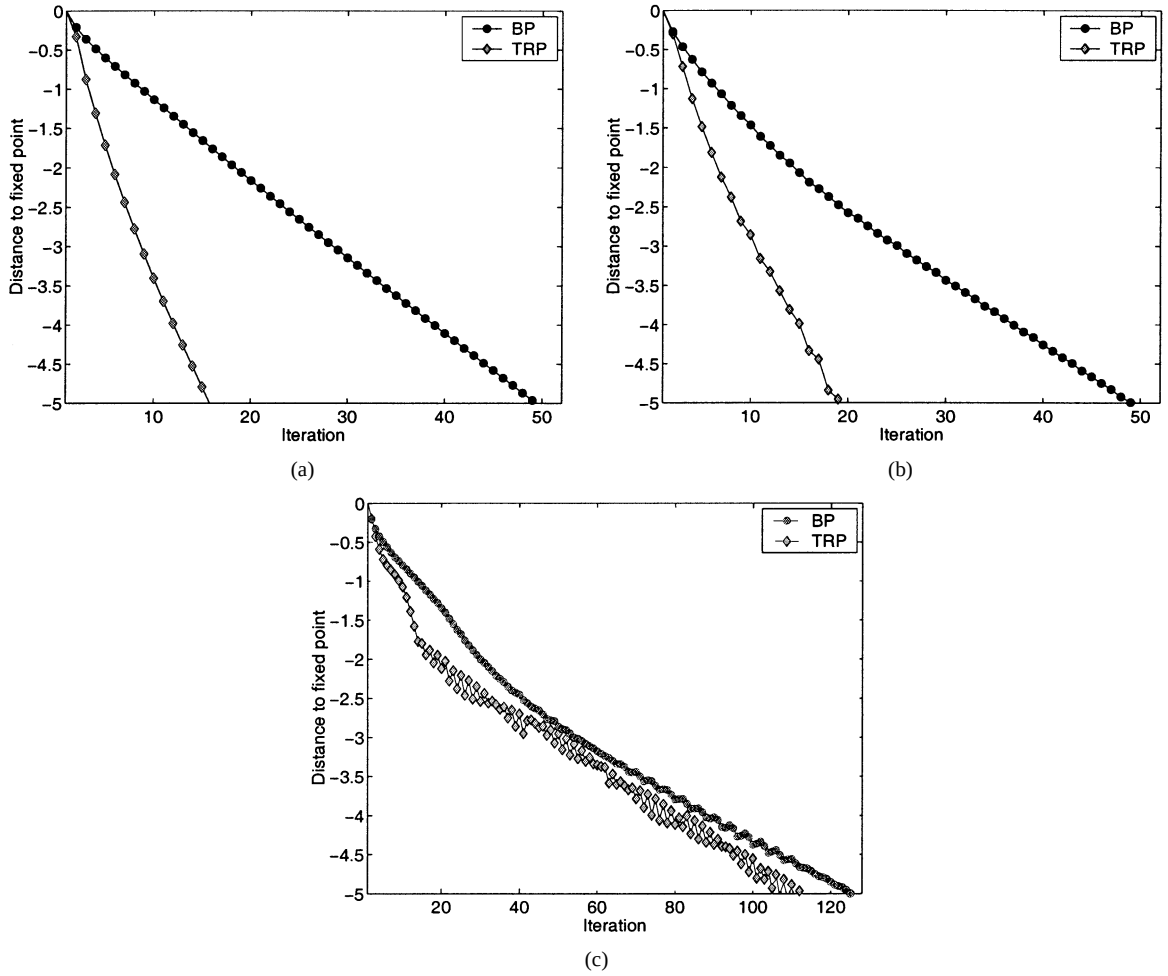



Fig. 6. Convergence rates for TRP versus BP on a $40 \times 40$ grid. Plotted on a log scale is the $L^2$ distance $(\sum_{s,\,j} |\theta^n_{s;\,j} - \theta^*_{s;\,j}|^2)$ from current iterate $\theta^n$ to fixed point $\theta^*$ versus iteration number $n$. In all cases, both BP and TRP converge to the same fixed point $\theta^*$. (a) Repulsive potentials. (b) Mixed potentials. (c) Particular choice of mixed potentials that causes difficulty for TRP.

defined in Section IV-B, that arises as an approximation to the KL divergence and agrees with the Bethe free energy [3] on the constraint set. In Section IV-C, we show that successive TRP iterates satisfy a Pythagorean relation with respect to this cost function. This result is of independent interest because it establishes links to successive projection techniques for constrained minimization of Bregman distances (e.g., [32]). In Section IV-D, we use this Pythagorean relation to prove that fixed points of the TRP algorithm satisfy necessary conditions to be a constrained minimum of this cost function. By combining our results with those of Yedidia *et al.* [3], we conclude that fixed points of the TRP algorithm coincide with those of BP. The Pythagorean result also allows us to formulate a set of sufficient conditions for convergence of TRP in the case of two spanning trees, which we briefly discuss in Section IV-E. In Section IV-F, we provide an elementary proof of the result originally developed in [19], [18] concerning the behavior of BP for jointly Gaussian distributions.

### A. Geometry and Invariance of TRP Updates

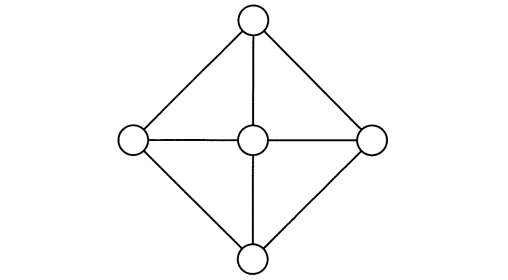
Highlighted by our informal setup in Section III is an important property of reparameterization algorithms like TRP and BP—namely, they do not change the original distribution on the graph with cycles. In this section, we formalize this notion of invariance, and show that it also holds for the more general class of relaxed updates in (20). On the basis of this invariance, we provide an illustration of the TRP updates in terms of information geometry [41], [31], [46]. This geometric perspective provides intuition and guides our subsequent analysis of reparameterization algorithms and their fixed points.

From the perspective of reparameterization, a crucial feature of the exponential parameterization defined in (9) is its overcompleteness. For this reason, given a fixed exponential parameter $\tilde{\theta}$, it is interesting to consider the following subset of $\mathbb{R}^{d(\theta)}$:

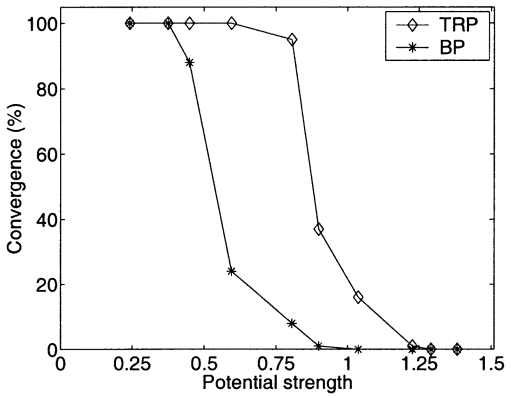
$$\mathcal{M}(\tilde{\theta}) \triangleq \left\{\theta \in \mathbb{R}^{d(\theta)} \,|\, p(\boldsymbol{x};\ \theta) \equiv p(\boldsymbol{x};\ \tilde{\theta})\right\} \tag{24}$$

where $d(\theta)$ denotes the length of $\theta$ as defined in (10). This set can be seen to be a closed submanifold of $\mathbb{R}^{d(\theta)}$—in particular, note that it is the inverse image of the point $\tilde{\theta}$ under the continuous mapping $\theta \mapsto p(\boldsymbol{x};\ \theta)$.

In order to further understand the structure of $\mathcal{M}(\tilde{\theta})$, we need to link the overcomplete parameterization to a minimal parameterization, specified by a linearly independent collection of functions. To illustrate the basic intuition, we begin with the special case of binary-valued nodes $(m = 2)$. In this case, the indicator functions of the $\theta$-representation can be expressed as



(a)




(b)

Fig. 7. (a) Simple five-node graph. (b) Comparison of BP and TRP convergence percentages versus function of potential strength on graph in (a). Plotted along the abscissa as a measure of potential strength is the multi-information $D(p(\boldsymbol{x};\,\theta)\,\|\,\prod_{s=1}^{N} p(x_s;\,\theta))$. Both TRP and BP exhibit a threshold phenomenon, with TRP converging for a wider range of potentials.

linear combinations of the functions $x_s$ and $x_s x_t$. For example, we have $\delta_{s;\,0}(x_s)\delta_{t;\,1}(x_t) = (1 - x_s)x_t$. Thus, in the binary case, a minimal parameterization is given by

$$p(\boldsymbol{x};\,\gamma) = \exp\left\{\sum_s \gamma_s x_s + \sum_{s,\,t\in E} \gamma_{st} x_s x_t - \Phi(\gamma)\right\}. \quad (25)$$

Such a model is known as the Ising model in statistical physics (e.g., [8]).

These ideas are extended readily to discrete processes with $m > 2$ states. In general, for a graph with pairwise cliques, the following collection of functions constitute a minimal representation:

$$\mathcal{R}(s) \triangleq \{x_s^a \,|\, a = 1,\,\ldots,\,m-1\}, \qquad \forall\, s \in V \quad (26a)$$

$$\mathcal{R}(s,\,t) \triangleq \{x_s^a x_t^b \,|\, a,\,b = 1,\,\ldots,\,m-1\}, \qquad \forall\,(s,\,t). \quad (26b)$$

As in the binary case illustrated above, we let $\gamma$ be a parameter vector of weights on these functions.

In contrast to the overcomplete case, the minimal representation induces a one-to-one correspondence between parameter vectors $\gamma$ and distributions $p(\boldsymbol{x};\,\gamma)$. Therefore, associated with the distribution $p(\boldsymbol{x};\,\tilde{\theta})$ is a unique vector $\tilde{\gamma}$ such that $p(\boldsymbol{x};\,\tilde{\theta}) \equiv p(\boldsymbol{x};\,\tilde{\gamma})$. The dimension of the exponential family (see [30]) is given by the length of $\gamma$, which we denote by $d(\gamma)$. From (26), we see that this dimension is

$$d(\gamma) = [(m-1)N + (m-1)^2|E|]$$

where $|E|$ is the number of edges in the graph. On the basis of these equivalent representations, the set $\mathcal{M}(\tilde{\theta})$ can be characterized as follows.

*Lemma 1:* The set $\mathcal{M}(\tilde{\theta})$ of (24) is an affine subspace ($e$-flat submanifold) of $\mathbb{R}^{d(\theta)}$ of dimension $d(\theta) - d(\gamma)$. It has the form $\{\theta \in \mathbb{R}^{d(\theta)} \,|\, A\theta = \tilde{\gamma}\}$, where $A$ is an appropriately defined $d(\gamma) \times d(\theta)$ matrix of constraints.

*Proof:* See Appendix A. □

Based on this lemma, we can provide a geometric statement and proof of the invariance of TRP updates, as well as message-passing algorithms.

*Theorem 2—Invariance of Distribution:*

a) Any sequence of TRP iterates $\{\theta^n\}$, whether relaxed or unrelaxed, specifies a sequence of reparameterizations of the original distribution on the graph with cycles. More specifically, for any $n \in \mathbb{N}$, the iterate $\theta^n$ belongs to the set

$$\mathcal{M}(\theta^0) = \{\theta \in \mathbb{R}^{d(\theta)} \,|\, p(\boldsymbol{x};\,\theta) \equiv p(\boldsymbol{x};\,\theta^0)\}.$$

b) Similarly, any form of BP message passing, when suitably reformulated (see Section III-D), specifies a sequence of reparameterizations.

*Proof:*

a) As previously described, the unrelaxed TRP update of (19) does indeed leave the distribution unchanged, so that $\mathcal{Q}^i(\theta) \in \mathcal{M}(\theta)$ for all $\theta$. The relaxed update of (20) is nothing more than a convex combination of two exponential vectors ($\theta^n$ and $\mathcal{Q}^{i(n)}(\theta^n)$) that parameterize the same distribution, so that by recourse to Lemma 1, the proof of the first statement is complete. As noted earlier, $\mathcal{M}(\theta^0)$ is a closed submanifold, so that any limit point of the sequence $\{\theta^n\}$ must also belong to $\mathcal{M}(\theta^0)$.

b) Given a message-passing algorithm, the messages $M^n = \{M_{st}^n\}$ at iteration $n$ define a set of pseudomarginals as follows:

$$T_{s;\,j}^n = \kappa\psi_{s;\,j} \prod_{u\in\Gamma(s)} M_{us;\,j}^n \quad (27a)$$

$$T_{st;\,jk}^n = \kappa\psi_{st;\,jk}\psi_{s;\,j}\psi_{t;\,k} \prod_{u\in\Gamma(s)\setminus t} M_{us;\,j}^n \prod_{u\in\Gamma(t)\setminus s} M_{ut;\,k}^n. \quad (27b)$$

Note that from these definitions, it follows that for any edge $(s,\,t)$, the ratio $T_{st;\,jk}^n / [T_{s;\,j}^n T_{t;\,k}^n]$ is proportional to $\psi_{st;\,jk} / [M_{ts;\,j}^n M_{st;\,k}^n]$. Using this observation, we see that the product

$$\prod_{s\in V} T_{s;\,j}^n \prod_{(s,\,t)\in E} \frac{T_{st;\,jk}^n}{T_{s;\,j}^n T_{t;\,k}^n}$$

is proportional to the following expression:

$$\prod_{s\in V}\left[\psi_{s;\,j} \prod_{u\in\Gamma(s)} M_{us;\,j}^n\right] \prod_{(s,\,t)\in E} \frac{\psi_{st;\,jk}}{M_{ts;\,j}^n M_{st;\,k}^n}.$$

Now for any edge $(w,\,v)$, consider the message $M_{wv;\,j}^n$. Note that it appears once in the numerator (in the definition of $T_v^n$), and once in the denominator (in the $(w,\,v)$ edge term). Therefore, all messages cancel out in the product, and we are left with the assertion that the pseudomarginals specify a reparameterization of the original distribution. □

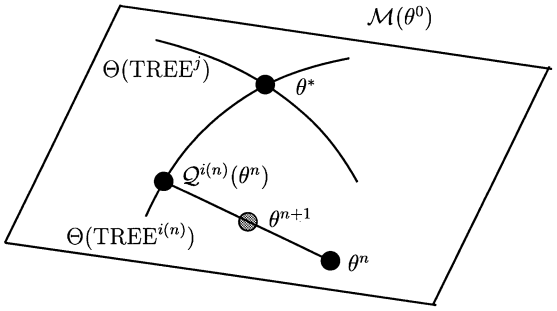



Fig. 8. Geometry of tree-reparameterization updates in the exponential domain. Iterates are confined to the linear manifold $\mathcal{M}(\theta^0)$. Curved lines within $\mathcal{M}(\theta^0)$ correspond to the intersection $\Theta^i(\mathrm{TREE}^i) \cap \mathcal{M}(\theta^0)$, for a particular spanning tree constraint set $\Theta(\mathrm{TREE}^i)$. Each update entails moving along the line between $\theta^n$ and the point $\mathcal{Q}^{i(n)}(\theta^n)$ on $\Theta(\mathrm{TREE}^{i(n)})$. Any fixed point $\theta^*$ belongs to the intersection of $\Theta(\mathrm{TREE}) = \cap_i \Theta(\mathrm{TREE}^i)$ with $\mathcal{M}(\theta^0)$.

Theorem 2a) and Lemma 1 together lead to a geometric understanding of the TRP updates in the exponential domain (i.e., in terms of the parameter vector $\theta$). In order to describe this geometry, consider the image of the constraint set $\mathrm{TREE}$ under the mapping $\Theta$; it is a subset of the exponential domain, which we denote by $\Theta(\mathrm{TREE})$. Any vector $\theta \in \Theta(\mathrm{TREE})$ must satisfy certain nonlinear convex constraints (e.g., $\log[\sum_j \exp(\theta_{s;\,j})] = 0$ for all $s \in V$; and $\log[\sum_j \exp(\theta_{st;\,jk} + \theta_{s;\,j})] = 0$ for all $(s,\,t) \in E$). For each spanning tree constraint set $\mathrm{TREE}^i$, we also define the image $\Theta(\mathrm{TREE}^i)$ in an analogous manner. Note that for any $\theta$, the updated $\mathcal{Q}^i(\theta)$ is guaranteed to belong to $\Theta(\mathrm{TREE}^i)$. Moreover, the setup of the algorithm ensures that $\Theta(\mathrm{TREE}) = \cap_i \Theta(\mathrm{TREE}^i)$.

Fig. 8 illustrates the geometry of TRP updates. The sequence of iterates $\{\theta^n\}$ remains within the linear manifold $\mathcal{M}(\theta^0)$. In terms of information geometry [30], this manifold is $e$-flat, since it is linear in the exponential parameters. Note that $\mathrm{TREE}$ and each $\mathrm{TREE}^i$ are defined by linear constraints in terms of the (pseudo)marginal $T$, and so are $m$-flat manifolds. Each set $\Theta(\mathrm{TREE}^i)$ is a curved manifold in the space of exponential parameters. Therefore, the intersection $\Theta(\mathrm{TREE}^i) \cap \mathcal{M}(\theta^0)$ forms a curved line, as illustrated in Fig. 8. Each update consists of moving along the straight line between the current iterate $\theta^n$, and the point $\mathcal{Q}^{i(n)}(\theta^n)$ obtained by applying the tree reparameterization operator $\mathcal{Q}^{i(n)}$. By construction, the vector $\mathcal{Q}^{i(n)}(\theta^n)$ belongs to the constraint set $\Theta(\mathrm{TREE}^{i(n)})$. The ultimate goal of a reparameterization algorithm is to obtain a point $\theta^*$ in the intersection $\cap_i \Theta(\mathrm{TREE}^i)$ of all the tree constraint sets.

### *B. Approximation to the KL Divergence*

Based on the geometric view illustrated in Fig. 8, an unrelaxed TRP update corresponds to moving from $\theta^n$ to the point $\mathcal{Q}^{i(n)}(\theta^n)$ in the constraint set $\Theta(\mathrm{TREE}^{i(n)})$. We now show that this operation shares certain properties of a projection operation—that is, it can be formulated as finding a point in $\mathrm{TREE}^{i(n)}$ that is closest to $\theta^n$ in a certain sense. The cost function $\mathcal{G}$, central to our analysis, arises as an approximation to the KL divergence [43], one which is exact for a tree.

Let $T \in (0,\,1)^{d(\theta)}$ be a pseudomarginal vector, and let $\theta$ be a parameter vector for the original graph $G$ with cycles. As building blocks for defining the full cost function, we define functions for each node $s$ and edge $(s,\,t)$ as follows:

$$\mathcal{G}_{st}(T_{st};\,\theta_{st}) = \sum_{j,\,k} T_{st;\,jk}\left[\log \frac{T_{st;\,jk}}{\left(\sum_j T_{st;\,jk}\right)\left(\sum_k T_{st;\,jk}\right)} - \theta_{st;\,jk}\right]$$

$$\mathcal{G}_s(T_s;\,\theta_s) = \sum_j T_{s;\,j}\left[\log T_{s;\,j} - \theta_{s;\,j}\right]. \tag{28}$$

We then define the cost function as

$$\mathcal{G}(T;\,\theta) = \sum_{s\in V} \mathcal{G}_s(T_s;\,\theta_s) + \sum_{(s,\,t)\in E} \mathcal{G}_{st}(T_{st};\,\theta_{st}) = \sum_{\alpha\in\mathcal{A}} T_\alpha\left[\Theta(T) - \theta\right]_\alpha. \tag{29}$$

This cost function is equivalent to the Bethe free energy [3] when $T$ belongs to the constraint set $\mathrm{TREE}$, but distinct for vectors $T$ that do not satisfy the marginalization constraints defining membership in $\mathrm{TREE}$.

To see how $\mathcal{G}$ is related to the KL divergence, as defined in (8), consider the analogous function defined on spanning tree $\mathcal{T}^i$ for a vector $T \in \mathrm{TREE}^i$

$$\mathcal{G}^i(\Pi^i(T);\,\Pi^i(\theta)) = \sum_{s\in V} \mathcal{G}_s(T_s;\,\theta_s) + \sum_{(s,\,t)\in E^i} \mathcal{G}_{st}(T_{st};\,\theta_{st}) = \sum_{\alpha\in\mathcal{A}^i} T_\alpha\left[\Theta^i(\Pi^i(T)) - \theta\right]_\alpha \tag{30}$$

where $\Pi^i(\theta)$ and $\Pi^i(T)$ are exponential parameter vectors and marginal vectors, respectively, defined on $\mathcal{T}^i$. With the exponential parameterization of (9) applied to any tree, we have $T_\alpha = \mathbb{E}_{\Theta^i(\Pi^i(T))}[\phi_\alpha(\boldsymbol{x})]$ for all indexes $\alpha \in \mathcal{A}^i$. As a result, the function $\mathcal{G}^i$ is related to the KL divergence as follows:

$$D(\Theta^i(\Pi^i(T))\,\|\,\Pi^i(\theta)) = \mathcal{G}^i(\Pi^i(T);\,\Pi^i(\theta)) + \Phi(\Pi^i(\theta)). \tag{31}$$

In establishing this equivalence, we have used the fact that the partition function of the factorization in (2) is unity, so that the corresponding log partition function is zero (i.e., $\Phi(\Theta^i(\Pi^i(T))) = 0$). Therefore, aside from an additive constant $\Phi(\Pi^i(\theta))$ independent of $T$, the quantity $\mathcal{G}^i(\Pi^i(T);\,\Pi^i(\theta))$, when viewed as a function of $\Pi^i(T)$, is equivalent to the KL divergence.

Now consider the problem of minimizing the KL divergence as a function of $T$, subject to the constraint $T \in \mathrm{TREE}^i$. The KL divergence in (31) assumes its minimum value of zero at the vector of correct marginals on the spanning tree—namely

$$P^i = \underline{\Lambda}^i(\Pi^i(\theta)) \in \mathrm{TREE}^i.$$

By the equivalence shown in (31), minimizing the function $\mathcal{G}^i(\Pi^i(T);\,\Pi^i(\theta))$ over $T \in \mathrm{TREE}^i$ will also yield the same minimizing argument $P^i$.

For the original graph with cycles, the cost function $\mathcal{G}$ of (29) is not equivalent to the KL divergence. The argument leading up to (31) cannot be applied because $\Lambda(\Theta(T)) \neq T$ for a general graph with cycles. Nevertheless, this cost function lies at the core of our analysis of TRP. Indeed, we show in Section IV-C how the TRP algorithm shares certain properties of a successive projection technique for constrained minimization of the cost function $\mathcal{G}$, in which the reparameterization update on spanning tree $\mathcal{T}^i$ as in (19) corresponds to a projection onto constraint set $\mathrm{TREE}^i$. Moreover, since $\mathcal{G}$ agrees with the Bethe free energy [3] on the constraint set $\mathrm{TREE}$, this allows us to establish equivalence of TRP fixed points with those of BP.

### C. Tree Reparameterization Updates as Projections

Given a linear subspace $\mathcal{L} \subset \mathbb{R}^n$ and a vector $\boldsymbol{y} \in \mathbb{R}^n$, it is well known [47] that the projection $\hat{\boldsymbol{x}}$ under the Euclidean norm (i.e., $\hat{\boldsymbol{x}} \triangleq \arg\min_{\boldsymbol{x} \in \mathcal{L}} ||\boldsymbol{x} - \boldsymbol{y}||$) is characterized by an orthogonality condition, or equivalently a Pythagorean relation. The main result of this subsection is to show that a similar geometric picture holds for TRP updates with respect to the cost function $\mathcal{G}$. In stating the result, we let $T^{n+1}$ denote a pseudomarginal vector[13] such that $\Theta(T^{n+1}) = \mathcal{Q}^{i(n)}(\theta^n)$.

*Proposition 2—Pythagorean Relation:* Assume that the sequence $\{\theta^n\}$ generated by (19) remains bounded. Let $i = i(n)$ be the tree index used at iteration $n$. Then, for all $U \in \mathrm{TREE}^i$

$$\mathcal{G}(U;\ \theta^n) = \mathcal{G}(U;\ \theta^{n+1}) + \mathcal{G}(T^{n+1};\ \theta^n). \tag{32}$$

*Proof:* See Appendix C. □

A result analogous to (32) holds for the minimum of a Bregman distance over a linear constraint set (e.g., [32]). Well-known examples of Bregman distances include the Euclidean norm, as well as the KL divergence. Choosing the KL divergence as the Bregman distance leads to the I-projection in information geometry (e.g., [41]), [31], [46].

Even when the distance is not the Euclidean norm, results of the form in (32) are still called Pythagorean, because the function $\mathcal{G}$ plays the role (in a loose sense) of the squared Euclidean distance. This geometric interpretation is illustrated in Fig. 9. For the unrelaxed updates, we use $T^n$ to denote the pseudomarginal satisfying $\theta^n = \Theta(T^n)$. The three points $T^n$, $T^{n+1}$, and $U$ are analogous to the vertices of a right triangle, as drawn in Fig. 9. We project the point $T^n$ onto the constraint set $\mathrm{TREE}^i$, where the function $\mathcal{G}^i$ serves as the distance measure. This projection yields the point $T^{n+1} \in \mathrm{TREE}^i$, and we have depicted its relation to an arbitrary $U$ also in $\mathrm{TREE}^i$.

It is worthwhile comparing Fig. 9 to Fig. 8, which represent the same geometry in the two different coordinate systems. Fig. 9 gives a picture of a single TRP update in terms of pseudomarginal vectors $T$; in this coordinate system, the constraint set $\mathrm{TREE}^i$ is affine and hence illustrated as a plane. Fig. 8 provides a similar picture in terms of the exponential parameters. The nonlinear mapping $\Theta$ transforms this constraint set $\mathrm{TREE}^i$ to its analog $\Theta(\mathrm{TREE}^i)$ in the exponential domain. As a consequence of the nonlinearity, the sets $\Theta(\mathrm{TREE}^i)$ in Fig. 8 are represented by curved lines in exponential coordinates. In Fig. 9, a single TRP update corresponds to moving along the straight line in exponential parameters between $\theta^n = \Theta(T^n)$ and the point $\mathcal{Q}^{i(n)}(\theta^n)$ that belongs in $\Theta(\mathrm{TREE}^i) \cap \mathcal{M}(\theta^0)$. Conversely, in Fig. 9, this same update is represented by moving along the curved line between $T^n$ and $T^{n+1}$.

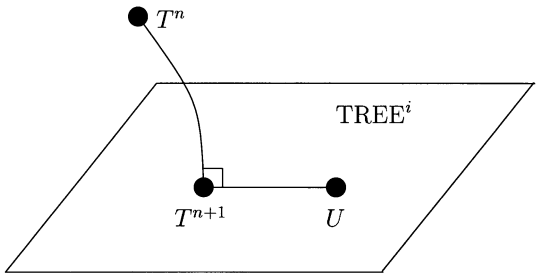



Fig. 9. Illustration of the geometry of Proposition 2. The pseudomarginal vector $T^n$ is projected onto the linear constraint set $\mathrm{TREE}^i$. This yields the point $T^{n+1}$ that minimizes the cost function $\mathcal{G}^i$ over the constraint set $\mathrm{TREE}^i$.

### D. Characterization of Fixed Points

Returning to the Euclidean projection example at the start of Section IV-C, consider again the problem of projecting $\boldsymbol{y} \in \mathbb{R}^n$ onto the linear constraint set $\mathcal{L} \subset \mathbb{R}^n$. Suppose that constraint set $\mathcal{L}$ can be decomposed as the intersection $\mathcal{L} = \cap_i \mathcal{L}^i$. Whereas it may be difficult to compute directly the projection $\hat{\boldsymbol{x}} \in \mathcal{L}$, performing projections onto the larger linear constraint sets $\mathcal{L}^i$ is often easier. In this scenario, one possible strategy for finding the optimal projection $\hat{\boldsymbol{x}} \in \mathcal{L}$ is to start at $\boldsymbol{y}$, and then perform a series of projections onto the constraint sets $\{\mathcal{L}^i\}$ in succession. In fact, such a sequence of projections is guaranteed [32] to converge to the optimal approximation $\hat{\boldsymbol{x}} \in \mathcal{L}$.

More generally, a wide class of algorithms can be formulated as successive projection techniques for minimizing a Bregman distance over a set formed by an intersection of linear constraints [32]. An example that involves a Bregman distance other than the Euclidean norm is the generalized iterative scaling algorithm [48], used to compute projections involving the KL divergence. A Pythagorean relation analogous to (32) is instrumental in establishing the convergence of such techniques [46], [32].

The problem of interest here is similar, since we are interested in finding a point belonging to a constraint set formed as an intersection of linear constraint sets (i.e., $\mathrm{TREE} = \cap_i \mathrm{TREE}^i$). However, the function $\mathcal{G}$ is certainly not a Bregman distance since, for instance, it can assume negative values; moreover, the TRP update at iteration $n$ minimizes $\mathcal{G}^{i(n)}$, as opposed to the full $\mathcal{G}$. Nonetheless, the Pythagorean result in Proposition 2 allows us to show that any bounded fixed point $\theta^*$ of the TRP algorithm satisfies the necessary conditions for it to be a local minimum of $\mathcal{G}(T;\ \theta^0)$ over the constraint set $\mathrm{TREE}$.

*Theorem 3—Characterization of Fixed Points:*

a) Fixed points of the TRP updates in (20) exist, and coincide with those of BP. Any fixed point $\theta^*$ of TRP is associated

[13]For an arbitrary exponential parameter $\theta^n$, this will not always be possible. For instance, observe that the image of the open unit hypercube $(0,\ 1)^{d(\theta)}$ under the map $\Theta$ is not all of $\mathbb{R}^{d(\theta)}$, since, for example, given any pseudomarginal $T \in (0,\ 1)^{d(\theta)}$, we have $[\Theta(T)]_{s;\,j} = \log T_{s;\,j} < 0$. Nonetheless, for unrelaxed updates producing iterates $\theta^n$, it can be seen that the inverse image of a point $\theta^n$ under $\Theta$ will be nonempty as soon as each edge has been updated at least once, in which case we can construct the desired $T^{n+1}$.

with a unique pseudomarginal vector $T^* \in \mathrm{TREE}$ that satisfies the necessary first-order condition to be a local minimum of $\mathcal{G}(T;\ \theta^0)$ over the constraint set TREE: that is, we have

$$\sum_{\alpha \in \mathcal{A}} \frac{\partial \mathcal{G}}{\partial T_\alpha}(T^*;\ \theta^0)\big[U - T^*\big]_\alpha = 0$$

for all $U$ in the constraint set TREE.

b) The pseudomarginal vector $T^*$ specified by any fixed point of TRP or, more generally, any algorithm that solves the Bethe variational problem, is *tree consistent* for every tree of the graph. Specifically, given a tree $\mathcal{T} = (V(\mathcal{T}),\ E(\mathcal{T}))$, the subcollection of pseudomarginals

$$\Pi^{\mathcal{T}}(T^*) = \{T_s^*,\ s \in V(\mathcal{T})\} \cup \{T_{st}^*,\ (s,\ t) \in E(\mathcal{T})\}$$

are the correct marginals for the tree-structured distribution defined as follows:

$$p^{\mathcal{T}}(\boldsymbol{x};\ T^*) = \prod_{s \in V(\mathcal{T})} T_s^*(x_s) \prod_{(s,\ t) \in E(\mathcal{T})} \frac{T_{st}^*(x_s,\ x_t)}{T_s^*(x_s) T_t^*(x_t)}. \tag{33}$$

*Proof:* See Appendix C. □

A few remarks about Theorem 3 are in order. First, with reference to statement a), the unique pseudomarginal vector $T^*$ associated with $\theta^*$ can be constructed explicitly as follows. For an arbitrary index $\alpha$, pick a spanning tree $\mathcal{T}^i$ such that $\alpha \in \mathcal{A}^i$. Then define $T_\alpha^* = [\Lambda^i(\Pi^i(\theta^*))]_\alpha$; that is, $T_\alpha^*$ is the value of this (single-node or pairwise) marginal for the tree distribution on $\mathcal{T}^i$ specified by $\Pi^i(\theta^*)$. Note that this is a consistent definition of $T_\alpha^*$, because the condition of part a) means that $[\Lambda^i(\Pi^i(\theta^*))]_\alpha$ is the same for all spanning tree indexes $i \in \{0,\ \ldots,\ L-1\}$ such that $\alpha \in \mathcal{A}^i$. Moreover, this construction ensures that $T^* \in \mathrm{TREE}$, since it must satisfy the normalization and marginalization constraints associated with every node and edge. With reference to any message-passing algorithm, any fixed point $M^* = \{M_{st}^*\}$ can be used to construct the unique pseudomarginal vector $T^*$ of Theorem 3 via (27a) and (27b).

Fig. 10 provides a graphical illustration of the tree-consistency statement of Theorem 3b). Shown in Fig. 10 (a) is an example of a graph $G$ with cycles, parameterized according to the approximate marginals $T_{st}^*$ and $T_s^*$. Fig. 10 (b) shows the tree obtained by removing edges $(4,\ 5)$ and $(5,\ 6)$ from the graph in Fig. 10 (a). Consider the associated tree-structured distribution $p^{\mathcal{T}}(\boldsymbol{x};\ T^*)$—that is, formed by removing the functions $T_{45}^*/T_4^*T_5^*$ and $T_{56}^*/T_5^*T_6^*$ from the original distribution $p(\boldsymbol{x})$. The consistency condition of Theorem 3b) guarantees that all the single-node pseudomarginals $T_s^*$, and all of the pairwise pseudomarginals $T_{st}^*$ except for the nontree edges $(4,\ 5)$ and $(5,\ 6)$, are *exact* marginals for this tree-structured distribution. For this reason, we say that the pseudomarginal vector $T^*$ is tree consistent with respect to the tree $\mathcal{T}$. In the context of the TRP algorithm, it is clear from definition of the updates (see Fig. 4) that this tree consistency must hold for any tree included in the set $\{\mathcal{T}^0,\ \ldots,\ \mathcal{T}^{L-1}\}$. In fact, tree consistency holds for *any* acyclic substructure embedded within the full graph with cycles—not just the spanning trees used to implement the algorithm.

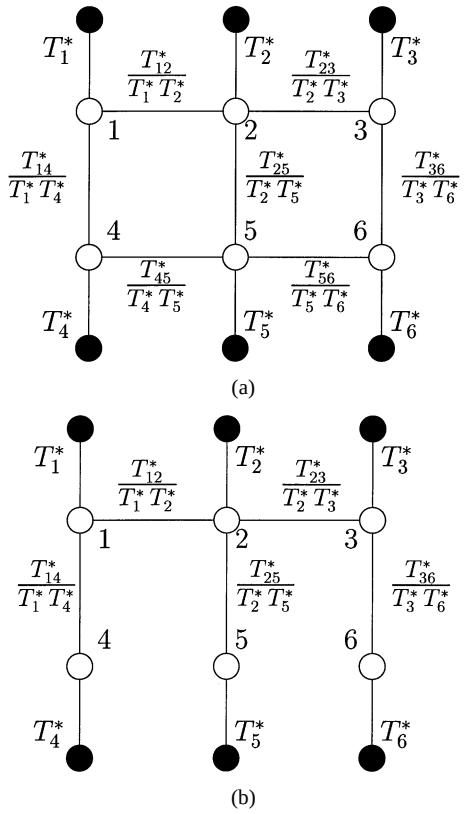



Fig. 10. Illustration of fixed-point consistency condition. (a) Fixed point $T^* = \{T_s^*,\ T_{st}^*\}$ that reparameterizes the original distribution on the full graph with cycles. (b) A tree $\mathcal{T}$ with edge set $E(\mathcal{T})$ formed by removing edges $(4,\ 5)$ and $(5,\ 6)$. The corresponding tree-structured distribution $p^{\mathcal{T}}(\boldsymbol{x};\ T^*)$ is formed by removing the reparameterized functions $T_{45}^*/(T_4^*T_5^*)$ and $T_{56}^*/(T_5^*T_6^*)$. The subset of pseudomarginals $\{T_s^* \mid s \in V\} \cup \{T_{st}^* \mid (s,\ t) \in E(\mathcal{T})\}$ are a consistent set of marginals for this tree-structured distribution $p^{\mathcal{T}}(\boldsymbol{x};\ T^*)$ constructed as in (33).

Overall, Theorems 2 and 3 in conjunction provide an alternative and very intuitive view of BP, TRP , and more generally, any algorithm for solving the Bethe variational problem (e.g., [23], [22]). This class of algorithms can be understood as seeking a reparameterization of the distribution on a graph with cycles (as in Theorem 2) that is consistent with respect to every tree of the graph (as in Theorem 3b)). Since these algorithms have fixed points, one consequence of our results is that any positive distribution on any graph can be reparameterized in terms of a set of pseudomarginals $T^*$ that satisfy the tree-based consistency condition of Theorem 3b). Although the existence of such a reparameterization is well known for trees [37], it is by no means obvious for an arbitrary graph with cycles.

### *E. Sufficient Conditions for Convergence for Two Spanning Trees*

Proposition 2 can also be used to derive a set of conditions that are sufficient to guarantee the convergence in the case of two spanning trees. To convey the intuition underlying the result, suppose that it were possible to interpret the cost function $\mathcal{G}$ as a distance function. Moreover, suppose $U$ were an arbitrary element of $\mathrm{TREE} = \cap_i \mathrm{TREE}^i$, so that we could apply Proposition 2 for each index $i$. Then (32) would show that the "distance" between $\theta^n$ and an arbitrary element $U \in \mathrm{TREE}$, as measured

by $\mathcal{G}$, decreases at each iteration. As with proofs on the convergence of successive projection techniques for Bregman distances (e.g., [32]), [46], this property would allow us to establish convergence of the algorithm.

Of course, there are two problems with the use of $\mathcal{G}$ as a type of distance: it is not necessarily nonnegative, and it is possible that $\mathcal{G}(\Lambda^i(\mathcal{Q}^i(\theta));\ \theta) = 0$ for some $\theta \neq \mathcal{Q}^i(\theta)$. With respect to the first issue, we are able to show in general that an appropriate choice of step size will ensure the nonnegativity of $\mathcal{G}(\Lambda^i(\mathcal{Q}^i(\theta));\ \theta)$. We can then derive sufficient conditions (including assuming that the second problem does not arise along TRP trajectories) for convergence in the case of two spanning trees. A detailed statement and proof of this result can be found in the thesis [45].

### *F. Implications for Continuous Variables*

Although our focus to this point has been on random vectors taking discrete values, the idea of reparameterization can applied to continuous variables as well. In particular, by extension to the Gaussian case, we obtain an elementary proof of the result [19], [18] that the means computed by BP, when it converges, are correct. To establish this result, consider the Gaussian analog of a reparameterization algorithm. For simplicity in notation, we treat the case of scalar Gaussian random variables at each node (though the ideas extend easily to the vector case). In the scalar Gaussian case, the pseudomarginal $T_s(x_s)$ at each node $s \in V$ is parameterized by a mean $\mu_s$ and variance $\sigma_s^2$. Similarly, the joint pseudomarginal $T_{st}(x_s, x_t)$ can be parameterized by a mean vector $\nu_{st} \triangleq [\nu_{st,1} \quad \nu_{st,2}]'$ and a covariance matrix. At any iteration, associated with the edge $(s, t)$ is the quotient $T_{st}/[T_{st\to s}T_{st\to t}]$, where

$$T_{st\to s}(x_s) = \int_{-\infty}^{\infty} T_{st}(x_s, x_t)\, dx_t$$

is the marginal distribution over $x_s$ induced by $T_{st}$. This edge function is parameterized by the mean vector $\nu_{st}$, and a $2 \times 2$ matrix $Q_{st}$. With this setup, we have the following proposition.

*Proposition 3:* Consider the Gaussian analog of BP, and suppose that it converges. Then the computed means are exact, whereas in general the error covariances are incorrect.

*Proof:* The original parameterization on the graph with cycles is of the form

$$-\log p(\boldsymbol{x}) = 1/2(\boldsymbol{x} - \hat{\mu})^T P^{-1}(\boldsymbol{x} - \hat{\mu}) + C. \qquad (34)$$

Here, $P^{-1}$ is the inverse covariance, $C$ is a constant independent of $\boldsymbol{x}$, and $\hat{\mu}$ is the exact mean vector on the graph with cycles.

We begin by noting that the Gaussian analog of Theorem 2 guarantees that the distribution will remain invariant under the reparameterization updates of TRP (or BP). At any iteration, the quantity $-\log p(\boldsymbol{x})$ is reparameterized in terms of $T_s$ and the edge functions as follows:

$$\frac{1}{2}\sum_{(s,t)\in E}\{[(x_s, x_t) - \nu_{st}]' Q_{st}[(x_s, x_t) - \nu_{st}]\} + \frac{1}{2}\sum_s (x_s - \mu_s)^2/\sigma_s^2 + C. \qquad (35)$$

Note that the pseudomarginal vector $\{T_s, T_{st}\}$ need not be consistent so that, for example, $T_{st\to s}(x_s)$ need not equal $T_s(x_s)$. However, suppose that TRP (or BP) converges so that these quantities are equal, which, in particular, implies that $\mu_s = \nu_{st;\,1}$ for all $(s, t)$ such that $t \in \Gamma(s)$. In words, the means parameterizing the edge functions must agree with the means at the node marginals. In this case, (34) and (35) are two alternative representations of the same quadratic form, so that we must have $\hat{\mu}_s = \mu_s$ for each node $s \in V$. Therefore, the means computed by TRP or BP must be exact. In contrast to the means, there is no reason to expect that the error covariances in a graph with cycles need be exact. □

It is worth remarking that there exist highly efficient techniques from numerical linear algebra (e.g., conjugate gradient) for computing the means of a linear Gaussian problem on a graph. Therefore, although algorithms like BP compute the correct means (if they converge), there is little reason to apply them in practice. There remains, however, the interesting problem of computing correct error covariances at each node: we refer the reader to [49], [50] for description of an embedded spanning tree method that efficiently computes both means and error covariances for a linear Gaussian problem on a graph with cycles.

## V. Analysis of the Approximation Error

An important but very difficult problem is analysis of the error in the BP approximation—that is, the difference between the exact marginals and the BP approximate marginals. As discussed in Section I, results on this error have been obtained for the special cases of single cycle graphs [11], and for turbo codes [13]. A more general understanding of this error is desirable in assessing the accuracy of the BP approximation. In this section, we make a contribution to this problem by deriving an exact expression for the error in the BP approximation for an arbitrary graph, as well as upper and lower bounds.

Our analysis of the BP approximation error is based on two key properties of a fixed point $\theta^*$. First, by the invariance stated in Theorem 2, the distribution $p(\boldsymbol{x};\ \theta^*)$ induced by the fixed point $\theta^*$ is equivalent to the original distribution $p(\boldsymbol{x};\ \theta^0)$. Second, part b) of Theorem 3 dictates that for an arbitrary spanning tree $\mathcal{T}$, the single-node elements $\theta^*_{s;\,j} = \log T^*_{s;\,j}$ correspond to exact marginal distributions on the spanning tree. Consequently, the quantities $T^*_s = \{T^*_{s;\,j} \,|\, j \in \mathcal{X}\}$ have two distinct interpretations:

a) as the BP approximations to the exact marginals on the graph with cycles;
b) as the exact single-node marginals of a distribution defined by the spanning tree $\mathcal{T}$.

The basic intuition is captured in Fig. 10. Fig. 10(a) shows the original distribution on the graph with cycles, now reparameterized in an alternative but equivalent manner by the set of pseudomarginals $T^* = \{T^*_s, T^*_{st}\}$. As a consequence, if it were possible to perform exact computations for the distribution illustrated in Fig. 10(a), the result would be the desired exact marginals $P_s$ of the original distribution. On the other hand, given a tree $\mathcal{T}$ like that shown in Fig. 10(b), suppose that we form a tree-structured distribution $p^{\mathcal{T}}(\boldsymbol{x};\ T^*)$, as in (33), by

removing the functions $T^*_{st}/(T^*_s T^*_t)$ on non-tree edges. Then, the tree-consistency condition of Theorem 3 ensures that the quantities $T^*_s$ are a consistent set of single-node marginals for $p^{\mathcal{T}}(\boldsymbol{x};\ T^*)$. Therefore, the exact marginals $P_s$ on the graph with cycles are related to the approximations $T^*_s$ by a relatively simply perturbation—namely, removing functions on edges to form a spanning tree. In the analysis that follows, we make this basic intuition more precise.

### A. Exact Expression

Our treatment begins at a slightly more general level, before specializing to the case of marginal distributions and the BP approximations. Consider a function $f\colon \mathcal{X}^N \to \mathbb{R}$, and two distributions $p(\boldsymbol{x};\ \tilde{\theta})$ and $p(\boldsymbol{x};\ \theta)$. Suppose that we wish to express the expectation $\mathbb{E}_{\tilde{\theta}}[f(\boldsymbol{x})]$ in terms of an expectation under the distribution $p(\boldsymbol{x};\ \theta)$. Using the exponential representation of (6), it is straightforward to re-express $\mathbb{E}_{\tilde{\theta}}[f(\boldsymbol{x})]$ as follows:

$$\mathbb{E}_{\theta}\left[\exp\left\{\sum_{\alpha}(\tilde{\theta}_\alpha - \theta_\alpha)\phi_\alpha(\boldsymbol{x}) + \Phi(\theta) - \Phi(\tilde{\theta})\right\} f(\boldsymbol{x})\right]. \tag{36}$$

Note that this is a change of measure formula, where the exponentiated quantity can be viewed as the Radon–Nikodym derivative.

We will use (36) to derive an expression for marginals on the graph with cycles in terms of an expectation over a tree-structured distribution. In particular, we denote the true single-node marginal $p(x_s = j;\ \theta^0)$ on the graph with cycles by

$$P_{s;\,j} \triangleq \mathbb{E}_{\theta^0}[\delta_{s;\,j}(x_s)] \overset{(a)}{=} \mathbb{E}_{\theta^*}[\delta_{s;\,j}(x_s)]. \tag{37}$$

To assert equality $(a)$, we have used the invariance property (i.e., $p(\boldsymbol{x};\ \theta^0) = p(\boldsymbol{x};\ \theta^*)$) of Theorem 2, as applied to the fixed-point $\theta^*$.

Given an arbitrary spanning tree $\mathcal{T} = (V,\ E(\mathcal{T}))$, let

$$\mathcal{A}(\mathcal{T}) = \{(s;\ j),\ (st;\ jk)\,|\,s \in V,\ (s,\ t) \in E(\mathcal{T})\}$$

be the set of tree indexes, and let $\Pi^{\mathcal{T}}(\theta^*) = \{\theta^*_\alpha\,|\,\alpha \in \mathcal{A}(\mathcal{T})\}$ be the projection of $\theta^*$ onto these tree indexes. By Theorem 3b), any fixed point $\theta^*$ is associated with a tree-consistent pseudomarginal vector $T^*$. More specifically, the tree consistency guarantees that the single-node elements of $T^*$ can be interpreted as the following expectation:

$$T^*_{s;\,j} \triangleq \mathbb{E}_{\Pi^{\mathcal{T}}(\theta^*)}[\delta_{s;\,j}(x_s)]. \tag{38}$$

We now make the assignments $\tilde{\theta} = \theta^*$, $\theta = \Pi^{\mathcal{T}}(\theta^*)$, and $f(\boldsymbol{x}) = \delta_{s;\,j}(x_s)$ in (36) and rearrange to obtain the exact expression for the error $\mathrm{Err}_{s;\,j} = \log T^*_{s;\,j} - \log P_{s;\,j}$ in the log domain shown in (39) at the bottom of the page. In deriving this expression, we have used the fact that $\Phi(\Pi^{\mathcal{T}}(\theta^*)) = 0$ for any tree $\mathcal{T}$. Equation (39) is an exact expression for the error $\mathrm{Err}_{s;\,j}$ in terms of an expectations involving the tree-structured distribution $p(\boldsymbol{x};\ \Pi^{\mathcal{T}}(\theta^*))$. Note that (39) holds for any spanning tree $\mathcal{T}$.

### B. Error bounds

It is important to observe that (39), though conceptually interesting, will typically be difficult to compute. A major obstacle arises from the presence of the term $\exp\{\sum_{\alpha\in\mathcal{A}\setminus\mathcal{A}(\mathcal{T})}\theta^*_\alpha\phi_\alpha(\boldsymbol{x})\}$ in the denominator. For most problems, this computation will not be tractable, since it involves all the potential functions on edges removed to form spanning tree $\mathcal{T}$. Indeed, if the computation of (39) were easy for a particular graph, this would imply that we could compute the *exact* marginals, thereby obviating the need for an approximate algorithm such as BP/TRP.

This intractability motivates the idea of bounding the approximation error. In order to do so, we begin by considering the following problem: given distributions $p(\boldsymbol{x};\ \tilde{\theta})$ and $p(\boldsymbol{x};\ \theta)$, and a function $f\colon \mathcal{X}^N \to \mathbb{R}$, give a bound for the expectation $\mathbb{E}_{\tilde{\theta}}[f(\boldsymbol{x})]$ in terms of quantities computed using the distribution $p(\boldsymbol{x};\ \theta)$. The linearity of expectation allows us to assume without loss of generality that $f(\boldsymbol{x}) \in [0,\ 1]$ for all $\boldsymbol{x} \in \mathcal{X}^N$. If $g$ is another function, the covariance of $f$ and $g$ under $p(\boldsymbol{x};\ \theta)$ is defined as

$$\mathrm{cov}_\theta\{f(\boldsymbol{x}),\ g(\boldsymbol{x})\} = \mathbb{E}_\theta[f(\boldsymbol{x})g(\boldsymbol{x})] - \mathbb{E}_\theta[f(\boldsymbol{x})]\mathbb{E}_\theta[g(\boldsymbol{x})]. \tag{40}$$

With these definitions, we have the following result.

*Lemma 2:* Let $\tilde{\theta}$ and $\theta$ be two arbitrary parameter vectors, and let $f$ be a function from $\mathcal{X}^N$ to the nonnegative reals. Then we have the lower bound

$$\mathbb{E}_{\tilde{\theta}}[f(\boldsymbol{x})] \geq \mathbb{E}_\theta[f(\boldsymbol{x})]\exp\left\{-D(\theta\,\|\,\tilde{\theta}) + \frac{1}{\mathbb{E}_\theta[f(\boldsymbol{x})]}\sum_{\alpha\in\mathcal{A}}(\tilde{\theta}_\alpha - \theta_\alpha)\mathrm{cov}_\theta\{f(\boldsymbol{x}),\ \phi_\alpha(\boldsymbol{x})\}\right\}. \tag{41}$$

*Proof:* See Appendix D. □

The bound in Lemma 2 is first order, based on the convexity of a (tilted) log-partition function; we note that tighter bounds of this nature can be derived by including higher order terms (see, e.g., [51]). We now use Lemma 2 to develop bounds on the approximation error in the single-node marginals. In order to state these bounds, it is convenient to define a quantity that,

$$\log\left\{\frac{\mathbb{E}_{\Pi^{\mathcal{T}}(\theta^*)}[\delta_{s;\,j}(x_s)]}{\mathbb{E}_{\Pi^{\mathcal{T}}(\theta^*)}\left[\left(\exp\left\{\sum_{\alpha\in\mathcal{A}\setminus\mathcal{A}(\mathcal{T})}\theta^*_\alpha\phi_\alpha(\boldsymbol{x})\right\} - \Phi(\theta^*)\right)\delta_{s;\,j}(x_s)\right]}\right\}. \tag{39}$$

for each spanning tree $\mathcal{T}$, measures the difference between the exact node marginal $P_{s;\,j}$ (corresponding to $p(\boldsymbol{x};\ \theta^*)$) and the approximation $T^*_{s;\,j}$ (corresponding to the tree-structured distribution $p(\boldsymbol{x};\ \Pi^{\mathcal{T}}(\theta^*))$). Consider the set $\{\phi_\alpha \,|\, \alpha \in \mathcal{A}\backslash\mathcal{A}(\mathcal{T})\}$, corresponding to the set of potentials that must be removed from the full graph $G$ in order to form the spanning tree $\mathcal{T}$. Associated with this tree, we define a quantity $\Delta^{\mathcal{T}}_{s;\,j}$ as a sum over the removed potentials

$$\Delta^{\mathcal{T}}_{s;\,j} \triangleq \sum_{\alpha\in\mathcal{A}\backslash\mathcal{A}(\mathcal{T})} \theta^*_\alpha \mathrm{cov}_{\Pi^{\mathcal{T}}(\theta^*)}\{\delta_{s;\,j}(x_s),\, \phi_\alpha(\boldsymbol{x})\}. \quad (42)$$

With this definition, we have the following error bounds on the approximation error.

*Theorem 4—Error Bounds:* Let $\theta^*$ be a fixed point of TRP/BP, giving rise to approximate single-node marginals $T^*_{s;\,j}$, and let $P_{s;\,j}$ be the true marginal distributions on the graph with cycles. For any spanning tree $\mathcal{T}$ of the graph, the error $\mathrm{Err}_{s;\,j} \triangleq \log T^*_{s;\,j} - \log P_{s;\,j}$ is bounded above and below as follows:

$$\mathrm{Err}_{s;\,j} \le D(\Pi^{\mathcal{T}}(\theta^*)\,\|\,\theta^*) - \frac{\Delta^{\mathcal{T}}_{s;\,j}}{T^*_{s;\,j}} \quad (43a)$$

$$\mathrm{Err}_{s;\,j} \ge \log T^*_{s;\,j} - \Upsilon^{\mathcal{T}}_{s;\,j} \quad (43b)$$

where

$$\Upsilon^{\mathcal{T}}_{s;\,j} \triangleq \log\left[1 - (1 - T^*_{s;\,j}) \times \exp\left\{-D(\Pi^{\mathcal{T}}(\theta^*)\,\|\,\theta^*) - \frac{\Delta^{\mathcal{T}}_{s;\,j}}{1 - T^*_{s;\,j}}\right\}\right].$$

*Proof:* We first make the identifications $\tilde{\theta} = \theta^*$ and $\theta = \Pi^{\mathcal{T}}(\theta^*)$, and then set $f(\boldsymbol{x}) = \delta_{s;\,j}(x_s)$, a choice which satisfies the assumptions of Lemma 2. Equation (43a) then follows by applying Lemma 2, and then performing by some algebraic manipulation. The lower bound follows via the same argument applied to $f(\boldsymbol{x}) = 1 - \delta_{s;\,j}(x_s)$, which also satisfies the hypotheses of Lemma 2. □

On the conceptual side, Theorem 4 highlights three factors that control the accuracy of the TRP/BP approximation. For the sake of concreteness, consider the upper bound of (43a).

a) The covariance terms in the definition of $\Delta^{\mathcal{T}}_{s;\,j}$ (see (42)) reflect the strength of the interaction, as measured under the tree distribution $p(\boldsymbol{x};\ \Pi^{\mathcal{T}}(\theta^*))$, between the delta function $\delta_{s;\,j}(x_s)$ and the clique potential $\phi_\alpha(\boldsymbol{x})$. When the removed clique potential interacts only weakly with the delta function, then this covariance term will be small and so have little effect.

b) The parameter $\theta^*_\alpha$ in the definition of $\Delta^{\mathcal{T}}_{s;\,j}$ is the strength of the clique potential $\phi_\alpha$ that was removed to form the spanning tree.

c) The KL divergence $D(\Pi^{\mathcal{T}}(\theta^*)\,\|\,\theta^*)$ measures the discrepancy between the tree-structured distribution $p(\boldsymbol{x};\ \Pi^{\mathcal{T}}(\theta^*))$ and the distribution $p(\boldsymbol{x};\ \theta^*)$ on the graph with cycles. It will be small when the distribution $p(\boldsymbol{x};\ \theta^*)$ is well approximated by a tree. The presence of this term reflects the empirical finding that BP performs well on graphs that are approximately tree-like (e.g., graphs with fairly long cycles).

On the practical side, there is a caveat associated with the computation of the upper and lower bounds in Theorem 4. On the one hand, the summations appearing in (43a) and (43b) are tractable. In particular, each of the covariances can be calculated by taking expectations over tree-structured distributions, and their weighted summation is even simpler. On the other hand, within the KL divergence $D(\Pi^{\mathcal{T}}(\theta^*)\,\|\,\theta^*)$ lurks a log-partition function $\Phi(\theta^*)$ associated with the graph with cycles. In general, computing this quantity is as costly as performing inference on the original graph. What is required, in order to compute the expressions in Theorem 4, are upper bounds on the log-partition function. A class of upper bounds are available for the Ising model [52]; in related work [53], [45], we have developed a technique for upper bounding the log partition function of an arbitrary undirected graphical model. Such methods allow upper bounds on the expressions in Theorem 4 to be computed.

### C. Illustrative Examples of Bounds

The tightness of the bounds given in Theorem 4 varies, depending on a number of factors, including the choice of spanning tree, the graph topology, as well as the strength and type of clique potentials. In this subsection, we provide some examples to illustrate the role of these factors.

For the purposes of illustration, it is more convenient to display bounds on the difference $T^*_{s;\,j} - P_{s;\,j} = \mathrm{Diff}_{s;\,j}$. Using Theorem 4, it is straightforward to derive the following bounds on this difference:

$$\mathrm{Diff}_{s;\,j} \le \left\{1 - \exp\left(-D(\Pi^{\mathcal{T}}(\theta^*)\,\|\,\theta^*) + \frac{\Delta^{\mathcal{T}}_{s;\,j}}{T^*_{s;\,j}}\right)\right\} \times \left[T^*_{s;\,j}\right] \quad (44a)$$

$$\mathrm{Diff}_{s;\,j} \ge \left\{\exp\left(-D(\Pi^{\mathcal{T}}(\theta^*)\,\|\,\theta^*) - \frac{\Delta^{\mathcal{T}}_{s;\,j}}{1 - T^*_{s;\,j}}\right) - 1\right\} \times \left[1 - T^*_{s;\,j}\right]. \quad (44b)$$

As a difference of marginal probabilities, the quantity $\mathrm{Diff}_{s;\,j}$ belongs to the interval $[-1,\,1]$. It can be seen that the upper and lower bounds are also confined to this interval. Since the primary goal of this subsection is illustrative, the bounds displayed here are computed using the exact value of $\Phi(\theta^*)$.

*1) Choice of Spanning Tree:* Note that a bound of the form in Theorem 4 (or (44)) holds for any singly connected subgraph embedded within the graph (not just the spanning trees used to implement the algorithm). This allows us to choose the *tightest* of all the spanning tree bounds for a given index $(s;\ j)$. Here we provide a simple example demonstrating the effect of tree choice on the quality of the bounds.

Using a binary distribution $p(\boldsymbol{x};\ \theta^*)$ on the complete graph $K_9$ with $N = 9$ nodes, we chose spanning trees according to the following heuristic. For a given (overcomplete) exponential parameter $\theta^*$, we first computed the minimal exponential

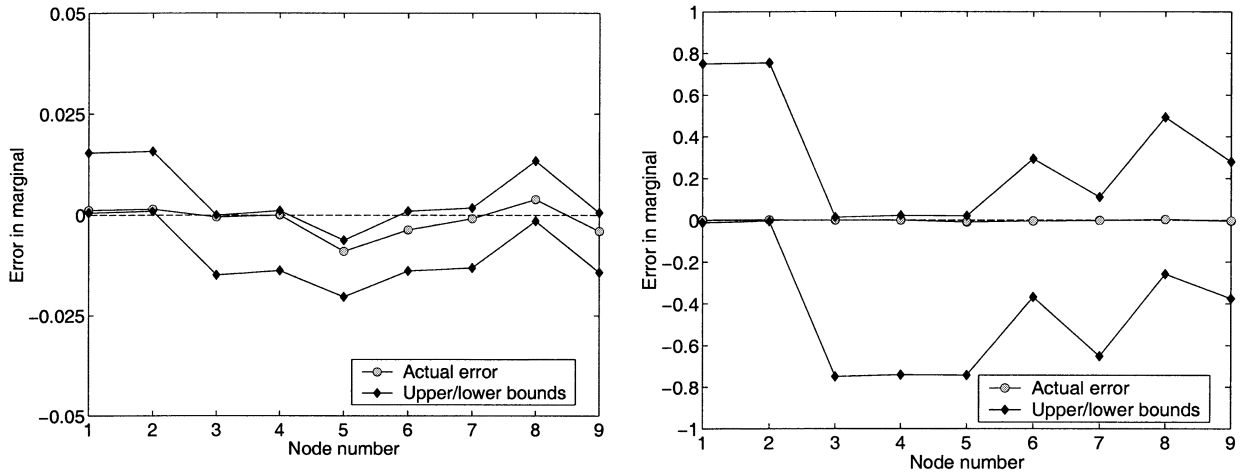



Fig. 11. Effect of spanning tree choice on the bounds of Theorem 4. Each panel shows the error $P_{s;\,0} - T^*_{s;\,0}$ versus node number for the complete graph $K_9$ on nine nodes. (a) Upper and lower bounds on the error computed using the maximum-weight spanning tree. (b) Error bounds for the *same problem* computed using the minimum-weight spanning tree. Note the difference in vertical scales between parts (a) and (b).

parameter $\gamma^*$, as in (25). Using $|\gamma^*_{st}|$ as the weight on edge $(s,\, t)$, we then computed the maximum- and minimum-weight spanning trees using Kruskal's algorithm (e.g., [54]). Fig. 11(a) and (b) shows the upper and lower bounds on the error $T^*_{s;\,j} - P_{s;\,j}$ obtained from (44a) and (44b), using the maximum- and minimum-weight spanning trees, respectively. The disparity between the two sets of bounds is striking: bounds on the error from the maximum-weight spanning tree (Fig.11 (a)) are very tight, whereas bounds from the minimum-weight tree are quite loose (Fig. 11(b)). Further work should examine principled techniques for optimizing the choice of spanning tree so as to obtain the tightest possible error bounds at a particular node.

*2) Varying Potentials:* Second, for a fixed graph, the bounds depend on both the strength and type of potentials. Here, we examine the behavior of the bounds for binary variables with attractive, repulsive, and mixed potentials; see Section III-E1 for the relevant definitions. Fig. 12 illustrates the behavior for a binary-valued process on a $10 \times 10$ grid under various conditions. Each panel shows the error $T^*_{s;\,0} - P_{s;\,0}$ for a randomly chosen subset of 20 nodes, as well as the lower and upper bounds on this error in (44). Fig.12(a) corresponds to a problem with very weak mixed potentials, for which both the BP approximation and the corresponding bounds are very tight. In contrast, Fig.12(b) shows a case with strong repulsive potentials, for which the BP approximation is poor. The error bounds are correspondingly weaker, but nonetheless track the general trend of the error. Fig. 12(c) and (d) shows two cases with attractive potentials: medium strength in Fig.12(c), and very strong in Fig. 12(d). In Fig.12(c), the BP approximation is mediocre, and the bounds are relatively loose. For the strong potentials in Fig. 12(d), the BP approximation is heavily skewed, and the lower bound is nearly met with equality for many nodes.

*3) Uses of Error Analysis:* The preceding examples serve to illustrate the general behavior of the bounds of Theorem 4 for a variety of problems. We reiterate that these bounds were computed using the exact value of $\Phi(\theta^*)$; in general, it will be necessary to upper-bound this quantity (see, e.g., [53]), which will tend to weaken the bounds. For sufficiently strong potentials on very large graphs, the upper and lower bounds in (44) may tend toward $+1$ and $-1$, respectively, in which case the bounds would no longer provide useful quantitative information. Nonetheless, the error analysis itself can still be useful. First, one direction to explore is the possibility of using our exact error analysis to derive correction terms to the BP approximation. For instance, the term $\Delta^{\mathcal{T}}_{s;\,j}$ defined in (42) is a computable first-order correction term to the BP approximation. Understanding when such corrections are useful is an interesting open problem. Second, our error analysis can be applied to the problem of assessing the relative accuracy of different approximations. As we discuss in Section VI, various extensions to BP (e.g., [33], [29], [25], [4]) can be analyzed from a reparameterization perspective, and a similar error analysis is applicable. Since the (intractable) partition function of the original model is the same regardless of the approximation, it plays no role in the relative error between the accuracy of different approximations. Therefore, these bounds could be useful in assessing when a more structured approximation, like generalized BP [33], yields more accurate results.

## VI. Extensions of Tree-Based Reparameterization

A number of researchers have proposed and studied extensions to BP that involve operating over higher order clusters of nodes (e.g., [33], [4], [25], [29]). In this section, we describe how such extensions of BP can also be interpreted as performing reparameterization.

We begin our development with background on *hypergraphs* (e.g., [55]), which represent a generalization of ordinary graphs and provide a useful framework in which to discuss generalized forms of reparameterization. In this context, the natural generalization of trees are known as hypertrees (i.e., acyclic hypergraphs). As will be clear, the notion of a hypertree is intimately tied to that of a junction tree [37]. Equipped with this background, we then describe how to perform reparameterization updates over hypertrees. As a specific illustration, we show how generalized BP (GBP) updates over Kikuchi clusters on

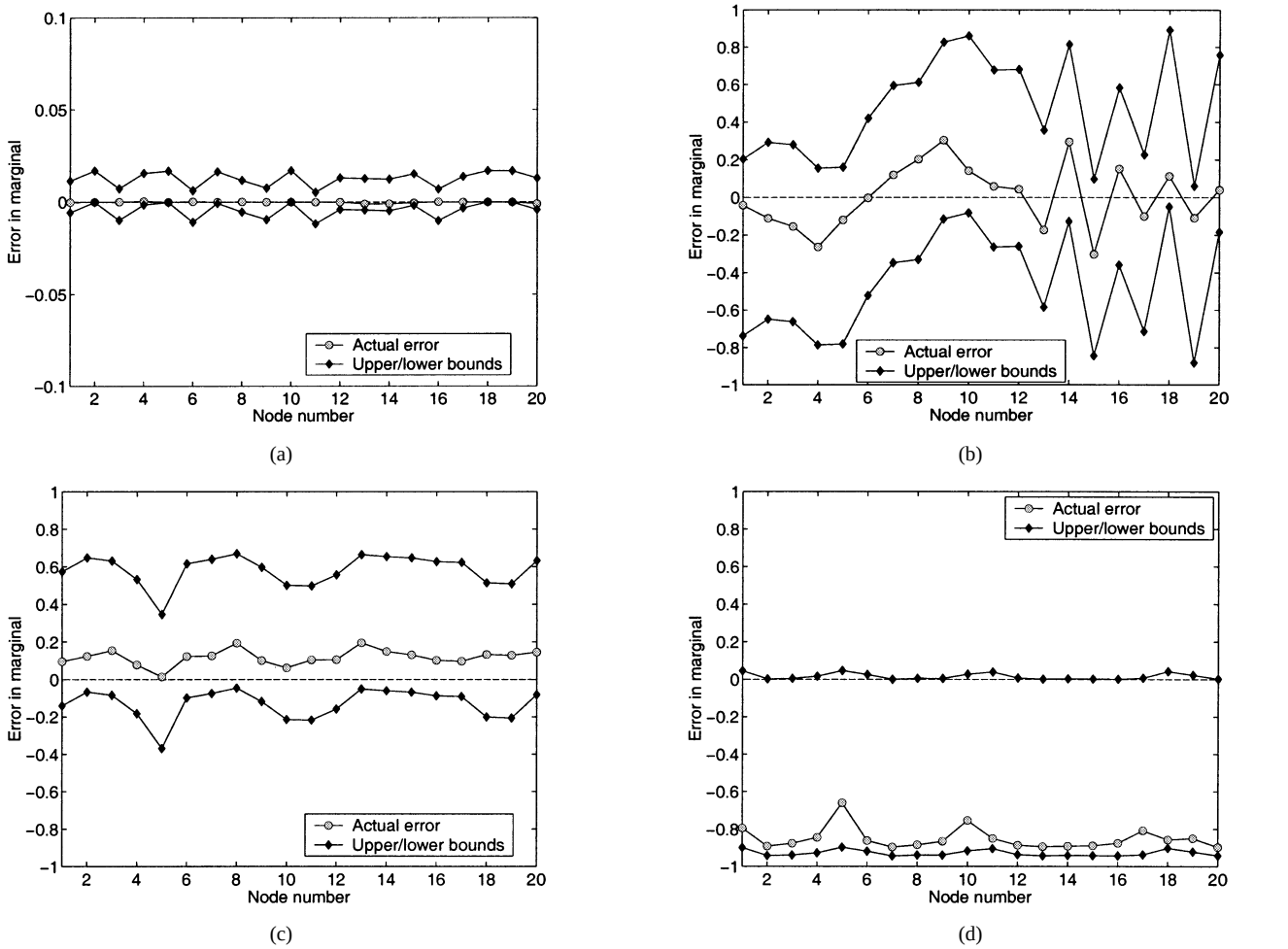



Fig. 12. Behavior of the bounds of Theorem 4 for various types of problem on a $10 \times 10$ grid. Each panel shows the error $T^*_{s;\,0} - P_{s;\,0}$ in the BP approximation versus node index, for a randomly chosen subset of 20 nodes. (a) For weak potentials, the BP approximation is excellent, and the bounds are quite tight. (b) For strong repulsive potentials, the BP approximation is poor; the error bounds are considerably looser, but still track the error. (c) Medium-strength attractive potentials, for which the approximation is mediocre and the bounds are relatively loose. (d) For strong attractive potentials, the BP approximation is very poor, and the difference between lower and upper bounds is relatively loose.

the two-dimensional grid can be understood as reparameterization over hypertrees. The reparameterization perspective allows much of our earlier analysis, including the geometry, characterization of fixed points, and error analysis, to be extended to generalizations of BP in a natural way. More details on this reparameterization analysis are provided in [45].

### A. Hypergraphs

A hypergraph $G_{\mathrm{HYP}} = (V, E)$ consists of a vertex set $V = \{1, \ldots, N\}$ and a set of hyperedges $E$, where each *hyperedge* $h$ is a particular subset of $V$ (i.e., an element of the power set of $V$). The set of hyperedges can be viewed as a partially ordered set, where the partial ordering is specified by inclusion. More details on hypergraphs can be found in Berge [55], whereas Stanley [56] provides more information on partially ordered sets (also known as posets).

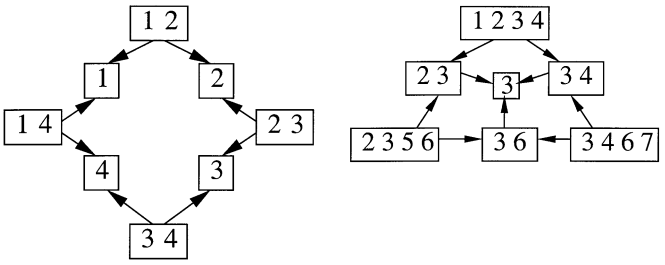
Given two hyperedges $g$ and $h$, one of three possibilities can hold: i) the hyperedge $g$ is contained within $h$, in which case we write $g < h$; ii) conversely, when the hyperedge $h$ is contained within $g$, we write $h < g$; iii) finally, if neither containment relation holds, then $g$ and $h$ are incomparable. We say that a hyperedge is *maximal* if it is not contained within any other hyperedge. Given any hyperedge $h$, we define the sets of its *descendants* and *ancestors* in the following way:

$$\mathcal{D}(h) = \{g \in E \,|\, g < h\} \tag{45a}$$

$$\mathcal{A}(h) = \{g \in E \,|\, g > h\}. \tag{45b}$$

With these definitions, an ordinary graph is a special case of a hypergraph, in which each maximal hyperedge consists of a pair of vertices (i.e., an ordinary edge of the graph). Note that for hypergraphs (unlike graphs), the set of hyperedges may include (a subset of) the individual vertices.

A convenient graphical representation of a hypergraph is in terms of a diagram of its hyperedges, with (directed) edges representing the inclusion relations. Diagrams of this nature have been used by Yedidia *et al.* [3], who refer to them as region graphs; other researchers [4], [25] have adopted the term Hasse diagram from poset terminology. Fig. 13 provides some simple graphical illustrations of hypergraphs. As a special case, any ordinary graph can be drawn as a hypergraph; in particular, Fig. 13(a) shows the hypergraph representation of a single cycle on four nodes. Shown in Fig. 13(b) is a more complex hypergraph that does not correspond to an ordinary graph.




(a) (b)

Fig. 13. Graphical representations of hypergraphs. Subsets of nodes corresponding to hyperedges are shown in rectangles, whereas the arrows represent inclusion relations among hyperedges. (a) An ordinary single-cycle graph represented as a hypergraph. (b) A more complex hypergraph that does not correspond to an ordinary graph.

### B. Hypertrees

Of particular importance are acyclic hypergraphs, which are also known as hypertrees. In order to define these objects, we require the notions of tree decomposition and running intersection, which are well known in the context of junction trees (see [57], [37]). Given a hypergraph $G_{\mathrm{HYP}}$, a *tree decomposition* is an acyclic graph in which the nodes are formed by the maximal hyperedges of $G_{\mathrm{HYP}}$. Any intersection $g \cap h$ of two maximal hyperedges that are adjacent in the tree is known as a *separator set*. The tree decomposition has the *running intersection property* if for any two nodes $g$ and $h$ in the tree, all nodes on the unique path joining them contain the intersection $g \cap h$; such a tree decomposition is known as a *junction tree*. A hypergraph is *acyclic* if it possesses a tree decomposition with the running intersection property. The *width* of an acyclic hypergraph is the size of the largest hyperedge minus one; we use the term *$k$-hypertree* to mean a singly connected acyclic hypergraph of width $k$.

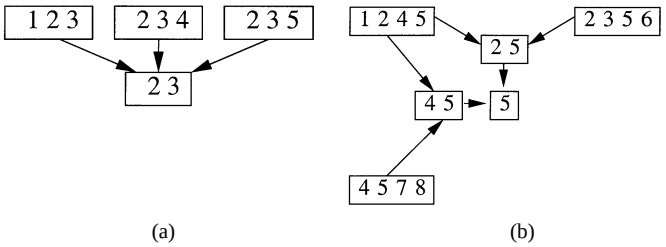
A simple illustration is provided by any tree of an ordinary graph: it is a 1-hypertree, because its maximal hyperedges (i.e., ordinary edges) all have size two. As a second example, the hypergraph of Fig. 14(a) has maximal hyperedges of size three. This hypergraph is acyclic with width two, since it is in direct correspondence with the junction tree formed by the three maximal hyperedges, and using the separator set (23) twice. Fig. 14(b) shows another hypertree with three maximal hyperedges of size four. The junction tree in this case is formed of these three maximal hyperedges, using the two hyperedges of size two as separator sets. In this case, including the extra hyperedge (5) in the hypergraph diagram turns out to be superfluous.

### C. Hypertree Factorization

At the heart of the TRP updates is the factorization of any tree-structured distribution in terms of its marginals, as specified in (2). In fact, this representation is a special case of a more general junction tree representation (e.g., [37])

$$p(\boldsymbol{x}) = \frac{\prod_{h \in E_{\max}} P_h(\boldsymbol{x}_h)}{\prod_{g \in E_{\mathrm{sep}}} \left[P_g(\boldsymbol{x}_g)\right]^{d(g)-1}}. \tag{46}$$

In this equation, $E_{\max}$ denotes the set of maximal hyperedges, $E_{\mathrm{sep}}$ denotes the set of separator sets in a tree decomposition,




(a) (b)

Fig. 14. Two examples of acyclic hypergraphs or hypertrees. (a) A hypertree of width two. The hyperedge (23) will appear twice as a separator set in any tree decomposition. (b) A hypertree of width 3. Hyperedges (25) and (45) are separator sets, and node 5 plays no role in the tree decomposition.

and $d(g)$ denotes the number of maximal hyperedges that contain the separator set $g$.

Instead of this junction tree form, it is convenient for our purposes to use an alternative but equivalent factorization, one which is based directly on the hypertree structure. To state this factorization, we first define, for each hyperedge $h \in E$, the following function:

$$\varphi_h(\boldsymbol{x}) \triangleq \frac{P_h(\boldsymbol{x}_h)}{\prod_{g \in \mathcal{D}(h)} \varphi_g(\boldsymbol{x}_g)}. \tag{47}$$

This definition is closely tied to the Möbius function associated with a partially ordered set (see [56]). With this notation, the factorization of a hypertree-structured distribution is very simple

$$p(\boldsymbol{x}) = \prod_{h \in E} \varphi_h(\boldsymbol{x}_h). \tag{48}$$

Note that the product is taken over all hyperedges (not just maximal ones) in the hypertree.

To illustrate (48), suppose first that the hypertree is an ordinary tree, in which case the hyperedge set consists of the union of the vertex set with the (ordinary) edge set. For any vertex $s$, we have $\varphi_s(x_s) = P_s(x_s)$, whereas we have

$$\varphi_{st}(x_s, x_t) = P_{st}(x_s, x_t)/[P_s(x_s)\, P_t(x_t)]$$

for any edge $(s, t)$. Therefore, in this special case, (48) reduces to the tree factorization in (2). As a second illustration, consider the hypertree shown in Fig. 14(a), which has maximal hyperedges $\{(123), (234), (235)\}$. By explicit computation, it can be verified that the factorization of (48) reduces, after canceling terms and simplifying, to $[P_{123}\ P_{234}\ P_{235}]/[P_{23}^2]$. Here we have omitted the dependence on $\boldsymbol{x}$ for notational simplicity. This finding agrees with the junction tree representation in (46) when applied to the appropriate tree decomposition. A third example is provided by the hypertree of Fig. 14(b). We first calculate $\varphi_{1245}$ as follows:

$$\varphi_{1245} = \frac{P_{1245}}{\varphi_{25}\varphi_{45}\varphi_5} = \frac{P_{1245}}{\frac{P_{25}}{P_5}\frac{P_{45}}{P_5}P_5} = \frac{P_{1245}P_5}{P_{25}P_{45}}.$$

Similarly, we compute $\varphi_{4578} = P_{4578}/P_{45}$ (with an analogous expression for $\varphi_{2356}$), $\varphi_{45} = P_{45}/P_5$ (with an analogous expression for $\varphi_{25}$), and $\varphi_5 = P_5$. Finally, taking the product $\prod_{h \in E} \varphi_h$ yields the familiar junction tree factorization for this particular case—viz. $[P_{1245}\ P_{4578}\ P_{2356}]/[P_{45}\ P_{25}]$.

### D. Reparameterization Over Hypertrees

With this setup, we are now ready to describe the extension of reparameterization to hypertrees. Our starting point is a distribution $p(\boldsymbol{x})$ formed as a product of compatibility functions associated with the hyperedges of some (hyper)graph $\tilde{G}_{\mathrm{HYP}}$. Rather than performing inference directly on this hypergraph, we may wish to perform some clustering (e.g., as in Kikuchi methods) so as to form an augmented hypergraph $G_{\mathrm{HYP}}$. As emphasized by Yedidia *et al.* [33], it is important to choose the hyperedges in the augmented hypergraph with care, so as to ensure that every compatibility function in the original hypergraph is counted exactly once; in other words, the single counting criteria must be satisfied. As an illustration, suppose that our original hyperedge $\tilde{G}_{\mathrm{HYP}}$ is the two-dimensional nearest neighbor grid shown in Fig. 15(a) (i.e., simply an ordinary graph). Illustrated in Fig. 15(b) is a particular grouping of the nodes, known as Kikuchi 4-plaque clustering in statistical physics [3]. Fig. 15(c) shows one possible choice of augmented hypergraph $G_{\mathrm{HYP}}$ that satisfies the single counting conditions, obtained by Kikuchi clustering [33].

We will take as given a hypergraph $G_{\mathrm{HYP}}$ that satisfies the single counting criteria, and such that there is an upper bound $k+1$ on the size of the maximal hyperedges. The implicit assumption is that $k$ is sufficiently small so that exact inference can be performed for hypertrees of this width. The idea is to perform a sequence of reparameterization updates, entirely analogous to those of TRP updates, except over a collection of hypertrees of width $k$ embedded within the hypergraph. The updates involve a collection $T = \{T_h \,|\, h \in E\}$ of pseudomarginals on hyperedges of the hypergraph. At any iteration, the collection $T^n$ specifies a particular parameterization $p(\boldsymbol{x};\, T^n)$ of the original distribution $p(\boldsymbol{x})$, analogous to the hypertree factorization of (48). (See Appendix E for more details on this parameterization.) Given any hypertree $\mathcal{T}$, the distribution $p(\boldsymbol{x})$ can be decomposed into a product of two terms $p^i(\boldsymbol{x})$ and $r^i(\boldsymbol{x})$, where $p^i(\boldsymbol{x})$ includes those factors corresponding to the hyperedges in the hypertree, while $r^i(\boldsymbol{x})$ absorbs the remaining terms, corresponding to hyperedges removed to form the hypertree. Now the hypertree distribution $p^i(\boldsymbol{x})$ can be reparameterized in terms of a collection $T$ of pseudomarginals on its hyperedges, as in (48). As with ordinary tree reparameterization, this operation simply specifies an alternative but equivalent choice for the compatibility functions corresponding to those hyperedges in the hypertree. A subsequent update entails choosing a *different* hypertree $\mathcal{T}^j$, and performing reparameterization for it. As before, we iterate over a set of hypertrees that covers all hyperedges in the hypergraph. With specific choices of hypergraphs, it can be shown that fixed points of hypertree reparameterization are equivalent to fixed points of various forms of GBP [3], [33].

As one illustration, consider again the Kikuchi approximation of Fig. 15, and more specifically the augmented hypergraph $G_{\mathrm{HYP}}$ in Fig. 15(c). We can form an acyclic hypergraph (though not spanning) of $G_{\mathrm{HYP}}$ by removing the hyperedge $(5689)$. The resulting hypertree of width three is shown in Fig. 15(d). Hypertree updates entail performing reparameterization on a collection of hypertrees that cover the hypergraph of Fig. 15(c). As we show in more detail in Appendix F, doing so yields fixed points equivalent to GBP applied to the Kikuchi approximation associated with the hypergraph in Fig. 15(c).

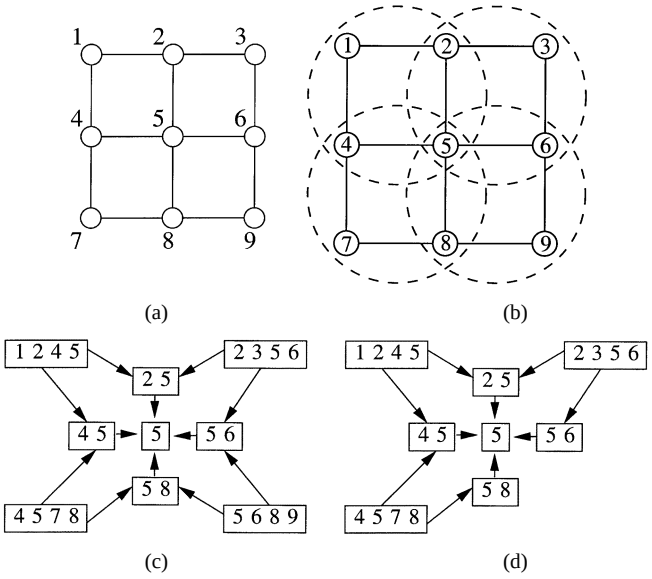



Fig. 15. Kikuchi clustering on the grid. (a) Original two-dimensional grid. (b) Clustering into groups of four. (c) Associated hypergraph with maximal hyperedges of size four. (d) One acyclic hypergraph of width three (not spanning) embedded within (c).

Since hypertree updates are a natural generalization of TRP updates, they inherit the important properties of TRP , including the fact that the original distribution $p(\boldsymbol{x})$ is not altered. Rather, any fixed points specify an alternative parameterization such that the pseudomarginals $T^*$ are consistent on every hypertree (of width $\leq k$) embedded within the graph. Based on this invariance and the fixed-point characterization, it is again possible to derive an exact expression, as well as bounds, for the difference between the exact marginals and the approximate marginals computed by any hypertree reparameterization algorithm. More details on the notion of reparameterization and its properties can be found in [45].

## VII. Discussion

This paper introduced a new conceptual framework for understanding sum-product and related algorithms, based on their interpretation as performing a sequence of reparameterization updates. Each step of reparameterization entails specifying a new set of compatibility functions for an acyclic subgraph (i.e., a (hyper)tree) of the full graph. The key property of these algorithms is that the original distribution is never changed, but simply reparameterized. The ultimate goal is to obtain a new factorization in which the functions on cliques are required to satisfy certain local consistency conditions, and represent approximations to the exact marginals. In particular, the pseudomarginals computed by a tree-reparameterization algorithm must be consistent on every tree embedded within the graph with cycles, which strongly constrains the nature of the approximation.

One important consequence of the reparameterization view is the insight that it provides into the nature of the approximation error associated with BP, and more broadly, with any algorithm that minimizes the Bethe free energy. In particular, the

error arises from the reparameterized functions sitting on those edges that must be removed in order to form a spanning tree. Based on this intuition, we derived an exact expression for the approximation error for an arbitrary graph with cycles. We also developed upper and lower bounds on this approximation error, and provided illustrations of their behavior. Computing these bounds for large problems requires an upper bound on the log partition function (e.g., [52], [53], [45]).

Overall, the class of algorithms that can be interpreted as performing reparameterization is large, including not only BP, but also various extensions to BP, among them generalized belief propagation [33]. For any algorithm that has this reparameterization interpretation, the same basic intuition is applicable, and there exist results (analogous to those of this paper) about their fixed points and the approximation error [45]. Finally, we note that a related concept of reparameterization also gives insight into the behavior of the closely related max-product algorithm [58].

## Appendix A
## Proof of Lemma 1

We begin by expressing the delta function $\delta_{s;\,j}(x_s)$ as a linear combination of the monomials in the set $\mathcal{R}(s)$ defined in (26a) as follows:

$$\delta_{s;\,j}(x_s) = \prod_{k \neq j} \frac{(k - x_s)}{(k - j)}. \tag{49}$$

This decomposition is extended readily to pairwise delta functions, which are defined by products $\delta_{s;\,j}(x_s)\delta_{t;\,k}(x_t)$; in particular, they can be written as linear combinations of elements in the sets $\mathcal{R}(s)$ and $\mathcal{R}(s,\,t)$, as defined in (26a) and (26b), respectively. Now suppose that $\theta \in \mathcal{M}(\theta^0)$, so that $\log p(\boldsymbol{x};\,\gamma^0) = \log p(\boldsymbol{x};\,\theta)$ for all $\boldsymbol{x} \in \mathcal{X}^N$. By construction, both the left-hand side (LHS) and right-hand side (RHS) are linear combination of the elements $\mathcal{R}(s)$ and $\mathcal{R}(s,\,t)$. Equating the coefficients of these terms yields a set of $d(\gamma) = (m-1)N + (m-1)^2|E|$ linear equations. We write these equations compactly in matrix form as $A\theta = \gamma^0$.

This establishes the necessity of the affine manifold constraints. To establish their sufficiency, we need only check that the linear constraints ensure the constant terms in $\log p(\boldsymbol{x};\,\gamma^0)$ and $\log p(\boldsymbol{x};\,\theta)$ are also equal. This is a straightforward verification.

## Appendix B
## Proof of Proposition 2

We begin with some preliminary definitions and lemmas. For a closed and convex set $X$, we say that $x^*$ is an *algebraic interior point* [32] if for all $x \neq x^*$ in $X$ there exists $x' \in X$, and $\lambda \in (0,\,1)$ such that $x^* = \lambda x + (1-\lambda)x'$. Otherwise, $x$ is an exterior point. The following lemma characterizes the nature of a constrained local minimum over $X$.

*Lemma 3:* Let $f\colon X \to \mathbb{R}$ be a $C^1$ function, where $X$ is a closed, convex, and nonempty set. Suppose that $x^*$ is a local minimum of $f$ over $X$. Then

$$\nabla f(x^*)^T(x - x^*) \geq 0$$

for all $x \in X$. Moreover, if $x^*$ is an algebraic interior point, then $\nabla f(x^*)^T(x - x^*) = 0$ for all $x \in X$.

*Proof:* See Bertsekas [47] for a proof of the first statement. To prove the second statement, assume that $x^*$ is an algebraic interior point, so that for an arbitrary $x \in X$ we can write $x^* = \lambda x + (1-\lambda)x'$ for some $x'$ and $\lambda \in (0,\,1)$. Then

$$\begin{aligned}\nabla f(x^*)^T(x' - x^*) &= \lambda \nabla f(x^*)^T(x' - x) \geq 0\\ \nabla f(x^*)^T(x - x^*) &= (1-\lambda)\nabla f(x^*)^T(x - x') \geq 0.\end{aligned}$$

Since $\lambda \in (0,\,1)$, this establishes that $\nabla f(x^*)^T(x' - x) = 0$, and hence, also $\nabla f(x^*)^T(x - x^*) = 0$. □

*Lemma 4:* Let $U \in \mathrm{TREE}^i$ be arbitrary. Then for any $\theta$ such that $\tilde{\theta} = \underline{\mathcal{R}}^i(\theta)$ is bounded

$$\sum_{\alpha \in \mathcal{A}^i} \left\{U_\alpha - \Lambda^i(\Pi^i(\theta))_\alpha\right\}\left[\tilde{\theta}_\alpha - \theta_\alpha\right] = 0. \tag{50}$$

*Proof:* We established in Section IV-B that the point $\Lambda^i(\Pi^i(\theta))$ is the minimizing argument of the function $\mathcal{G}^i$ defined in (30) over the linear and hence convex set $\mathrm{TREE}^i$. This point will be an exterior point only if some element is equal to zero or one, a possibility that is prevented by the assumption that $\underline{\mathcal{R}}^i(\theta) = \mathcal{I}^i(\Theta^i(\Lambda^i(\Pi^i(\theta))))$ is bounded. Therefore, $\Lambda^i(\Pi^i(\theta))$ is an algebraic interior point, meaning that we can apply Lemma 3 to conclude that for all $U \in \mathrm{TREE}^i$, we have

$$\sum_{\alpha \in \mathcal{A}^i} \left\{U_\alpha - \Lambda^i(\Pi^i(\theta))_\alpha\right\} \frac{\partial \mathcal{G}^i}{\partial T_\alpha}\left(\Lambda^i(\Pi^i(\theta))\theta\right) = 0. \tag{51}$$

It remains to calculate the necessary partial derivatives of $\mathcal{G}^i$. We begin with the decomposition

$$\mathcal{G}^i(T;\,\theta) = \sum_{(s,t)\in E^i} \mathcal{G}_{st}(T_{st};\,\theta_{st}) + \sum_{s \in V} \mathcal{G}_s(T_s;\,\theta_s)$$

where $\mathcal{G}_{st}$ and $\mathcal{G}_s$ are defined in (28). We then calculate

$$\frac{\partial \mathcal{G}^i}{\partial T_\alpha}(T;\,\theta) = \begin{cases} \Theta(T)_{s;\,j} - \theta_{s;\,j} + 1, & \text{for } \alpha = (s;\,j)\\ \Theta(T)_{st;\,jk} - \theta_{st;\,jk} - 1, & \text{for } \alpha = (st;\,jk).\end{cases}$$

We evaluate these partial derivatives at $\tilde{T} = \Lambda^i(\Pi^i(\theta))$, and use the notation $\tilde{\theta}_\alpha = \Theta^i(\tilde{T})_\alpha$ for each $\alpha \in \mathcal{A}^i$. Substituting the results into equation (51), we conclude that the sum

$$\begin{aligned}&\sum_{s \in V}\sum_{j}\left[U_{s;\,j} - \tilde{T}_{s;\,j}\right]\left[\tilde{\theta}_{s;\,j} - \theta_{s;\,j} + 1\right]\\ &\quad + \sum_{(s,t)\in E^i}\sum_{j,\,k}\left[U_{st;\,jk} - \tilde{T}_{st;\,jk}\right]\left[\tilde{\theta}_{st;\,jk} - \theta_{st;\,jk} - 1\right]\end{aligned} \tag{52}$$

is equal to zero. Now, since both $U$ and $\tilde{T}$ belong to $\mathrm{TREE}^i$, we have

$$\sum_j [U - \tilde{T}]_{s;\,j} = 0, \qquad \text{for all } s \in V$$

and

$$\sum_{j,k} [U_{st;\,jk} - \tilde{T}_{st;\,jk}] = 0, \qquad \text{for all } (s,\,t) \in E^i.$$

As a result, the constants 1 or $-1$ in (52) vanish in the sums over $j$ or $\{j,\,k\}$, and we are left with the desired statement in (50). □

Equipped with these lemmas, we first establish (32). For a given $\theta$, let $T$ denote the pseudomarginal such that $\Theta(T) = \mathcal{Q}^i(\theta)$. (Of particular importance is the fact that $T_\alpha = \Lambda^i(\Pi^i(\theta))_\alpha$ for all $\alpha \in \mathcal{A}^i$.) We then write

$$\begin{aligned}\mathcal{G}(U;\,\theta) - \mathcal{G}(U;\,\mathcal{Q}^i(\theta)) - \mathcal{G}(T;\,\theta) &= \sum_{\alpha\in\mathcal{A}} [U - T]_\alpha [\mathcal{Q}^i(\theta) - \theta]_\alpha \\ &= \sum_{\alpha\in\mathcal{A}^i} [U - T]_\alpha [\underline{\mathcal{R}}^i(\theta) - \theta]_\alpha\end{aligned}$$

where we used the definition

$$\mathcal{Q}^i(\theta)_\alpha = \begin{cases} \theta_\alpha, & \text{for all } \alpha \in \mathcal{A}\backslash\mathcal{A}^i \\ \underline{\mathcal{R}}^i(\theta)_\alpha, & \text{for all } \alpha \in \mathcal{A}^i. \end{cases}$$

Now $\underline{\mathcal{R}}^i(\theta)$ is bounded by assumption. Moreover, by construction, we have $T_\alpha = \Lambda^i(\Pi^i(\theta))_\alpha$ for all $\alpha \in \mathcal{A}^i$. Thus, we can apply Lemma 4 to conclude that

$$\mathcal{G}(U;\,\theta) - \mathcal{G}(U;\,\mathcal{Q}^i(\theta)) - \mathcal{G}(T;\,\theta) = 0 \tag{53}$$

thereby establishing (32) with the identifications $\theta \equiv \theta^n$, $T \equiv T^{n+1}$, and $i = i(n)$.

For use in the proof of Theorem 3, it is convenient to extend this result to the relaxed updates of (20), in which the step size $\lambda^n \in [0,\,1]$. In order to do so, use the definition of $\theta^{n+1}$ given in (20) to write the difference $\mathcal{G}(U;\,\theta^n) - \mathcal{G}(U;\,\theta^{n+1})$ as

$$\begin{aligned}\sum_{\alpha\in\mathcal{A}} U_\alpha [\theta^{n+1} - \theta^n]_\alpha &= \lambda^n \sum_{\alpha\in\mathcal{A}} U_\alpha [\mathcal{Q}^{i(n)}(\theta^n) - \theta^n]_\alpha \\ &= \lambda^n \left\{\mathcal{G}(U;\,\theta^n) - \mathcal{G}(U;\,\mathcal{Q}^{i(n)}(\theta^n))\right\} \\ &= \lambda^n \mathcal{G}(T^{n+1};\,\theta^n)\end{aligned}$$

where we have obtained the final line using (53). We have thus established

$$\mathcal{G}(U;\,\theta^n) = \mathcal{G}(U;\,\theta^{n+1}) + \lambda^n \mathcal{G}(T^{n+1};\,\theta^n). \tag{54}$$

## Appendix C
## Proof of Theorem 3

a) See the text following Theorem 3 for discussion of how to construct the unique pseudomarginal $T^* \in \mathrm{TREE}$ associated with any exponential parameter fixed point $\theta^*$. Now let $U \in \mathrm{TREE} = \cap_i \mathrm{TREE}^i$ be arbitrary. By applying (54) repeatedly, we obtain

$$\mathcal{G}(U;\,\theta^0) = \mathcal{G}(U;\,\theta^*) + \sum_{n=1}^{\infty} W_n \tag{55}$$

where

$$W_n = \lambda^n \mathcal{G}(T^{n+1};\,\theta^n) = \lambda^n \mathcal{G}^{i(n)}(\underline{\Lambda}^{i(n)}(\theta^{n+1});\,\theta^n).$$

We then apply (55) with $U = T^*$ (i.e., the pseudomarginal corresponding to $\theta^*$). By construction, we have $\mathcal{G}(T^*;\,\theta^*) = 0$, so that (55) yields $\mathcal{G}(T^*;\,\theta^0) = \sum_{n=1}^{\infty} W_n$. Substituting this result back into (55), we find that

$$\mathcal{G}(U;\,\theta^0) = \mathcal{G}(U;\,\theta^*) + \mathcal{G}(T^*;\,\theta^0) \tag{56}$$

for all $U \in \mathrm{TREE}$. To prove that $T^*$ satisfies the necessary conditions to be a local minimum, we note that (56) is equivalent to the statement

$$\sum_{\alpha\in\mathcal{A}} \frac{\partial \mathcal{G}}{\partial T_\alpha}(T^*;\,\theta^0)[U - T^*]_\alpha = 0 \tag{57}$$

where we have used a sequence of steps similar to the proof of Proposition 2. Since the cost function $\mathcal{G}$ is bounded below and the constraint set is nonempty, the problem has at least one minimum, which must satisfy these first-order stationarity conditions.

To establish equivalence with BP fixed points, recall that the cost function $\mathcal{G}$ agrees with the Bethe free energy on this constraint set. Yedidia *et al.* [3] have shown that BP fixed points correspond to points that satisfy the Lagrangian conditions for an extremum of the Bethe free energy over TREE. Moreover, because the constraint sets are linear, the existence of Lagrange multipliers is guaranteed for any local minimum [47]. By recourse to Farkas' lemma [47], these Lagrangian conditions are equivalent to the first-order condition (57). Therefore, TRP fixed points coincide with those of BP.

b) By definition, the elements of any pseudomarginal vector $T^* \in \mathrm{TREE}$ satisfy local normalization ($\sum_j T^*_{s;\,j} = 1$), and marginalization $\sum_k T^*_{st;\,jk} = T^*_{s;\,j}$) constraints. Given some tree $\mathcal{T} = (V(\mathcal{T}),\, E(\mathcal{T}))$, consider the subcollection of pseudomarginals

$$\Pi^{\mathcal{T}}(T^*) = \{T^*_s,\, s \in V(\mathcal{T})\} \cup \{T^*_{st},\, (s,\,t) \in E(\mathcal{T})\}.$$

By the junction tree theorem [37], the local consistency conditions are sufficient to guarantee global consistency for the subcollection $\Pi^{\mathcal{T}}(T^*)$. In particular, they are the exact single-node and pairwise marginal distributions of the tree-structured distribution formed as in (33).

## Appendix D
## Proof of Lemma 2

Consider the function

$$F(\theta) \triangleq \log \sum_{\boldsymbol{x}\in\mathcal{X}^N} \left[\exp\left\{\sum_\alpha \theta_\alpha \phi_\alpha(\boldsymbol{x})\right\} f(\boldsymbol{x})\right].$$

By interpreting $F$ as the log-partition function corresponding to the distribution $q(\boldsymbol{x};\ \theta) \propto \exp\{\sum_\alpha \theta_\alpha \phi_\alpha(\boldsymbol{x})\} f(\boldsymbol{x})$, we see that it is a convex function of $\theta$. For any parameter vectors $\tilde{\theta}$ and $\theta$, this convexity allows to write

$$F(\tilde{\theta}) \geq F(\theta) + \sum_\alpha \left.\frac{\partial F}{\partial \theta_\alpha}\right|_\theta (\tilde{\theta} - \theta)_\alpha. \tag{58}$$

We take derivatives of $F$ with respect to $\theta_\beta$ to find

$$\frac{\partial F}{\partial \theta_\beta}(\theta) = \mathbb{E}_\theta[f(\boldsymbol{x})\phi_\beta(\boldsymbol{x})]/\mathbb{E}_\theta[f(\boldsymbol{x})].$$

We also observe the relation $F(\theta) = \log \mathbb{E}_\theta[f(\boldsymbol{x})] + \Phi(\theta)$. Substituting these relations into (58), rearranging, and taking exponentials yields the bound of (41).

## Appendix E
## Hypergraph Parameterization

This appendix gives details of the hypergraph parameterization. Suppose that we are given a pseudomarginal $T_h$ for each hyperedge $h$ of the hypergraph (not just the maximal ones). Given the pseudomarginal $T_h$ and any hyperedge $g$ that is contained within $h$, we define

$$T_{g<h}(\boldsymbol{x}_g) = \sum_{\{\boldsymbol{x}'_h \mid \boldsymbol{x}'_g = \boldsymbol{x}_g\}} T_h(\boldsymbol{x}'_h). \tag{59}$$

That is, $T_{g<h}$ is the pseudomarginal on $g$ induced by $T_h$. At this initial stage, we are *not* assuming that the collection of pseudomarginals is fully consistent. For example, if we consider two distinct hyperedges $h$ and $f$ that both contain $g$, the pseudomarginal $T_{g<h}$ need not be equal to $T_{g<f}$. Moreover, neither of these quantities has to be equal to the pseudomarginal $T_g$ defined independently for hyperedge $g$. However, when the hypertree reparameterization updates converge, all of these consistency conditions will hold.

Now for each hyperedge $h$ with pseudomarginal $T_h$, we define the following quantity:

$$\varphi_h(\boldsymbol{x}_h;\ T_h) = \frac{T_h(\boldsymbol{x}_h)}{\prod_{g \in \mathcal{D}(h)} \varphi_g(\boldsymbol{x}_g;\ T_{g<h})} \tag{60}$$

where $\mathcal{D}(h)$ is defined in (45a).

This quantity $\varphi_h(\boldsymbol{x}_h;\ T_h)$ should be viewed as a function of both $\boldsymbol{x}_h$ and the pseudomarginal $T_h$. In addition, note that for any hyperedge $g < h$, it is the quantity $T_{g<h}$ (which need not agree with the pseudomarginal $T_g$ at intermediate stages of the algorithm) that is used in the recursive definition of (60). With these definitions, the collection $T$ specifies an alternative parameterization of the original distribution $p(\boldsymbol{x})$ on the full hypergraph (which need not be acyclic in general) as follows:

$$p(\boldsymbol{x};\ T) \propto \prod_{h \in E} \varphi_h(\boldsymbol{x}_h;\ T_h). \tag{61}$$

As one illustration, if the hypergraph arises from an ordinary graph $G$ with pairwise maximal cliques, then hyperedges consist of ordinary edges $(s,\ t)$ and vertices. For an ordinary edge $(s,\ t)$, we have

$$\varphi_{st}(x_s,\ x_t;\ T_{st}) = T_{st}(x_s,\ x_t) \Bigg/ \left[ \left\{ \sum_{x'_s} T_{st}(x'_s,\ x_t) \right\} \left\{ \sum_{x'_t} T_{st}(x_s,\ x'_t) \right\} \right]$$

whereas for any vertex $s$, we have $\varphi_s(x_s;\ T_s) = T_s(x_s)$. In this special case, then, the parameterization of (61) is equivalent to the $\{T_s,\ T_{st}\}$ parameterization that is updated by TRP/BP.

## Appendix F
## Hypertrees for Kikuchi Approximation

In this appendix, we show how generalized belief propagation [3], [33] with the Kikuchi approximation of Fig. 15(b) can be interpreted as a type of hypertree reparameterization. The augmented hypergraph in Fig. 15(c) has four maximal hyperedges, each of size four, namely: $(1245)$, $(2356)$, $(4578)$, and $(5689)$. Our hyperedge set consists of these hyperedges, as well as the (smaller) hyperedges contained within them. With a collection $T$ of pseudomarginals over this hyperedge set, the original distribution $p(\boldsymbol{x})$ is given the new parameterization $p(\boldsymbol{x};\ T)$, as in (61).

Our ultimate goal is a reparameterization of this form in which the hyperedge pseudomarginals are all locally consistent, and hence (by the junction tree theorem [37]) consistent on each hypertree. In order to find such a reparameterization, we perform a sequence of hypertree reparameterization updates, in which each step entails choosing some hypertree, and reparameterizing the corresponding hypertree distribution $p^i(\boldsymbol{x};\ T)$. For example, Fig. 15(d) shows the hypertree $\mathcal{T}$ (not spanning) obtained by removing the maximal hyperedge $(5689)$. On the analytical side, we form the distribution $p^i(\boldsymbol{x};\ T)$ by removing from $p(\boldsymbol{x};\ T)$, as defined in (61), the term $\varphi_{5689}(\boldsymbol{x};\ T_{5689})$ defined in (60).

We now perform reparameterization over a set of hypertrees that cover the hypergraph. If the updates converge, then the fixed point $T^*$ will be hypertree-consistent with respect to each of these hypertrees. Since the hypertrees cover the hypergraph, for any pair $g < h$, we are guaranteed the consistency condition $T^*_g(\boldsymbol{x}_g) = T^*_{g<h}(\boldsymbol{x}_g)$, where $T^*_{g<h}$ is defined in (59).

Suppose that we have a hypertree-consistent reparameterization for the particular Kikuchi approximation of Fig. 15. When the hyperedge consistency conditions are satisfied, there is no longer any need to distinguish between $T^*_g$ and $T^*_{g<h}$. Accordingly, the reparameterization $p(\boldsymbol{x};\ T^*)$ of (61) can be expanded and simplified in the following way. Omitting explicit dependence on $\boldsymbol{x}$, we first compute

$$\varphi_{1245} = \frac{T^*_{1245}}{\varphi_{25}\varphi_{45}\varphi_5} = \frac{T^*_{1245}}{\frac{T^*_{25}}{T^*_5}\frac{T^*_{45}}{T^*_5}T^*_5} = \frac{T^*_{1245}T^*_5}{T^*_{25}T^*_{45}}.$$

We can compute analogous expressions for the other maximal hyperedges. Next, we have $\varphi_{25} = T^*_{25}/T^*_5$, with analogous expressions for the other hyperedges of size two; and, finally,

$\varphi_5 = T_5^*$. We take the product $\prod_{h \in E} \varphi_h(\boldsymbol{x};\ T^*)$ over all the hyperedges, and then simplify to obtain

$$\frac{T^*_{1245}T^*_5}{T^*_{25}T^*_{45}}\ \frac{T^*_{2356}T^*_5}{T^*_{25}T^*_{56}}\ \frac{T^*_{5689}T^*_5}{T^*_{56}T^*_{58}}\ \frac{T^*_{4578}T^*_5}{T^*_{45}T^*_{58}} \times \frac{T^*_{25}}{T^*_5}\ \frac{T^*_{45}}{T^*_5}\ \frac{T^*_{56}}{T^*_5}\ \frac{T^*_{58}}{T^*_5} \times T^*_5 = \frac{T^*_{1245}T^*_{2356}T^*_{5689}T^*_{4578}T^*_5}{T^*_{25}T^*_{45}T^*_{56}T^*_{58}}.$$

The final line follows after some algebra, in which terms that are common to both the numerator and denominator cancel each other out. In canceling terms, we have used the consistency conditions satisfied by $T^*$, as described earlier. Finally, suppose that we take the logarithm of this factorization, and then take expectations with respect to the local pseudomarginals $T^*_h$. It can be seen that the resulting approximation to the entropy entropy agrees with the associated Kikuchi approximation for this problem, as discussed by Yedidia *et al.* [33]. Moreover, it can be shown that any local optimum of the associated Kikuchi variational problem is a hypertree-consistent reparameterization of this form.

## REFERENCES


[1] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Mateo, CA: Morgan Kaufman, 1988.

[2] R. J. McEliece, D. J. C. McKay, and J. F. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 140–152, Feb. 1998.

[3] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Generalized belief propagation," in *NIPS 13*. Cambridge, MA: MIT Press, 2001, pp. 686–695.

[4] R. J. McEliece and M. Yildirim, "Belief propagation on partially ordered sets," in *Mathematical Theory of Systems and Networks*. Minneapolis, MN: Inst. Math. and Applic., Aug. 2002.

[5] R. G. Gallager, *Low-Density Parity Check Codes*. Cambridge, MA: MIT Press, 1963.

[6] F. Kschischang and B. Frey, "Iterative decoding of compound codes by probability propagation in graphical models," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 219–230, Feb. 1998.

[7] M. Jordan, *Learning in Graphical Models*. Cambridge, MA: MIT Press, 1999.

[8] R. J. Baxter, *Exactly Solved Models in Statistical Mechanics*. New York: Academic, 1982.

[9] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *Intl. J. Computer Vision*, vol. 40, no. 1, pp. 25–40, 2000.

[10] G. Cooper, "The computational complexity of probabilistic inference using Bayesian belief networks," *Artificial Intell.*, vol. 42, pp. 393–405, 1990.

[11] Y. Weiss, "Correctness of local probability propagation in graphical models with loops," *Neural Comput.*, vol. 12, pp. 1–4, 2000.

[12] S. M. Aji and R. J. McEliece, "The generalized distributive law," *IEEE Trans. Inform. Theory*, vol. 46, pp. 325–343, Mar. 2000.

[13] T. Richardson, "The geometry of turbo-decoding dynamics," *IEEE Trans. Inform. Theory*, vol. 46, pp. 9–23, Jan. 2000.

[14] K. Murphy, Y. Weiss, and M. Jordan, "Loopy-belief propagation for approximate inference: An empirical study," *Uncertainty in Artificial Intell.*, vol. 15, pp. 467–475, 1999.

[15] J. B. Anderson and S. M. Hladnik, "Tailbiting map decoders," *IEEE J. Select. Areas Commun.*, vol. 16, pp. 297–302, Feb. 1998.

[16] S. M. Aji, G. Horn, and R. McEliece, "On the convergence of iterative decoding on graphs with a single cycle," in *Proc. IEEE Intl. Symp. Information Theory*, Cambridge, MA, Aug. 1998, p. 276.

[17] G. D. Forney, Jr., F. R. Kschischang, B. Marcus, and S. Tuncel, "Iterative decoding of tail-biting trellises and connections with symbolic dynamics," in *Codes, Systems and Graphical Models*. New York: Springer, 2001, pp. 239–264.

[18] Y. Weiss and W. T. Freeman, "Correctness of belief propagation in Gaussian graphical models of arbitrary topology," in *NIPS 12*. Cambridge, MA: MIT Press, 2000, pp. 673–679.

[19] P. Rusmevichientong and B. Van Roy, "An analysis of turbo decoding with Gaussian densities," in *NIPS 12*. Cambridge, MA: MIT Press, 2000, pp. 575–581.

[20] K. Tanaka and T. Morita, "Cluster variation method and image restoration problem," *Phys. Lett. A*, vol. 203, pp. 122–128, 1995.

[21] C. H. Wu and P. C. Doerschuk, "Tree approximations to Markov random fields," *IEEE Trans. Pattern Anal. Machine Inetll.*, vol. 17, pp. 391–402, Apr. 1995.

[22] M. Welling and Y. Teh, "Belief optimization: A stable alternative to loopy belief propagation," *Uncertainty in Artificial Intelligence*, vol. 17, pp. 554–561, July 2001.

[23] A. Yuille, "A double-loop algorithm to minimize the Bethe and Kikuchi free energies," Neural Comput., 2001, to be published.

[24] R. Kikuchi, "The theory of cooperative phenomena," *Phys. Rev.*, pp. 988–1003, 1951.

[25] P. Pakzad and V. Anantharam, "Belief propagation and statistical physics," in *Proc. Conf. Information Sciences and Systems*, Princeton, NJ, Mar. 2002.

[26] S. Tatikonda and M. I. Jordan, "Loopy belief propagation and Gibbs measures," *Proc. Uncertainty in Artificial Intell.*, vol. 18, pp. 493–500, Aug. 2002.

[27] S. L. Lauritzen and D. J. Spiegelhalter, "Local computations with probabilities on graphical structures and their application to expert systems (with discussion)," *J. Roy. Statist. Soc. B*, vol. 50, pp. 155–224, Jan. 1988.

[28] F. V. Jensen, *An Introduction to Bayesian Networks*. London, U.K.: UCL, 1996.

[29] T. P. Minka, "A family of algorithms for approximate Bayesian inference," Ph.D. dissertation, MIT, Cambridge, MA, Jan. 2001.

[30] S. Amari, "Differential geometry of curved exponential families—Curvatures and information loss," *Ann. Statist.*, vol. 10, no. 2, pp. 357–385, 1982.

[31] N. N. Chentsov, "Statistical decision rules and optimal inference," in *Translations of Mathematical Monographs*. Providence, RI: Amer. Math. Soc., 1982, vol. 53.

[32] Y. Censor and S. A. Zenios, "Parallel optimization: Theory, algorithms, and applications," in *Numerical Mathematics and Scientific Computation*. Oxford, U.K.: Oxford Univ. Press, 1988.

[33] J. S. Yedidia, W. T. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," Mitsubishi Elec. Res. Labs, Tech. Rep. TR2001–22, Jan. 2002.

[34] N. Biggs, *Algebraic Graph Theory*. Cambridge, U.K.: Cambridge Univ. Press, 1993.

[35] B. Bollobás, *Modern Graph Theory*. New York: Springer-Verlag, 1998.

[36] G. R. Grimmett, "A theorem about random fields," *Bull. London Math. Soc.*, vol. 5, pp. 81–84, 1973.

[37] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, "Probablistic networks and expert systems," *Statistics for Engineering and Information Science*, 1999.

[38] J. Besag, "Spatial interaction and the statistical analysis of lattice systems," *J. Roy. Statist. Soc. Ser. B*, vol. 36, pp. 192–236, 1974.

[39] M. Basseville, A. Benveniste, K. Chou, S. Golden, R. Nikoukhah, and A. Willsky, "Modeling and estimation of multiresolution stochastic processes," *IEEE Trans. Inform. Theory*, vol. 38, pp. 766–784, Mar. 1992.

[40] O. E. Barndorff-Nielson, *Information and Exponential Families*. Chichester, U.K.: Wiley, 1978.

[41] S. Amari, K. Kurata, and H. Nagaoka, "Information geometry of Boltzmann machines," *IEEE Trans. Neural Networks*, vol. 3, pp. 260–271, Mar. 1992.

[42] S. S. Gupta, Ed., "Differential geometry in statistical inference," in *Lecture Notes—Monograph Series*. Hayward, CA: Inst. Math. Statist., 1987, vol. 10.

[43] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[44] G. Rockafellar, *Convex Analysis*. Princeton, NJ: Princeton Univ. Press, 1970.

[45] M. J. Wainwright, "Stochastic processes on graphs with cycles: Geometric and variational approaches," Ph.D. dissertation, MIT, Cambridge, MA, Jan. 2002.

[46] I. Csiszár, "I-divergence geometry of probability distributions and minimization problems," *Ann. Probab.*, vol. 3, no. 1, pp. 146–158, Feb. 1975.

[47] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, 1995.

[48] J. N. Darroch and D. Ratcliff, "Generalized iterative scaling for log-linear models," *Ann. Math. Statist.*, vol. 43, pp. 1470–1480, 1972.

[49] M. J. Wainwright, E. B Sudderth, and A. S. Willsky, "Tree-based modeling and estimation of Gaussian processes on graphs with cycles," in *NIPS 13*. Cambridge, MA: MIT Press, 2001, pp. 661–667.

[50] E. Sudderth, "Embedded trees: Estimation of Gaussian processes on graphs with cycles," M.S. thesis, MIT, Cambridge, MA, Jan. 2002.
[51] M. A. R. Leisink and H. J. Kappen, "A tighter bound for graphical models," in *NIPS 13*. Cambridge, MA: MIT Press, 2001, pp. 266–272.
[52] T. S. Jaakkola and M. Jordan, "Recursive algorithms for approximating probabilities in graphical models," in *NIPS 9*. Cambridge, MA: MIT Press, 1996, pp. 487–493.
[53] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky, "A new class of upper bounds on the log partition function," in *Proc. Uncertainty in Artificial Intell.*, vol. 18, Aug. 2002, pp. 536–543.
[54] D. Jungnickel, *Graphs, Networks, and Algorithms*. New York: Springer-Verlag, 1999.
[55] C. Berge, *Hypergraphs*. Amsterdam, The Netherlands: North-Holland, 1989.
[56] R. P. Stanley, *Enumerative Combinatorics*. Cambridge, U.K.: Cambridge Univ. Press, 1997, vol. 1.
[57] S. L. Lauritzen, *Graphical Models*. Oxford, U.K.: Oxford Univ. Press, 1996.
[58] M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. (2002, July) Tree consistency and bounds on the max-product algorithm and its generalizations. MIT LIDS Tech. Rep. P-2554. [Online]. Available: http://www.eecs.berkeley.edu/~martinw