

Contents

Contents **1**

1 Linear programming and variants of Belief

Propagation **2**

- 1.1 Introduction 2
 - 1.1.1 Energy minimization and its Linear Programming Relaxation 3
 - 1.1.2 The need for special purpose LP solvers 6
- 1.2 Ordinary Sum-Product Belief Propagation and Linear Programming 6
 - 1.2.1 Convex BP and the LP Relaxation 9
- 1.3 Convex Max-Product BP 13
- 1.4 Discussion 16
- 1.5 Appendix - implementation details 17
 - 1.5.1 Implementation in log space 17
 - 1.5.2 Efficient message computation for Potts-model 17

Bibliography **19**

1 **Linear programming and variants of Belief Propagation**

Yair Weiss, CS HUJI, yweiss@cs.huji.ac.il

Chen Yanover, FHCRC, cyanover@fhcrc.org

Talya Meltzer, CS HUJI, talyam@cs.huji.ac.il

1.1 Introduction

The basic problem of energy minimization in an MRF comes up in many different application domains ranging from statistical physics [1] to error correcting codes [5] and protein folding [17]. *Linear Programming (LP) Relaxations* are a standard method for approximating combinatorial optimization problems in computer science [3] and have been used for energy minimization problems for some time [2,5,9]. LP relaxations have an advantage over other energy minimization schemes in that they come with an optimality guarantee – if the LP relaxation is “tight” i.e. the solution to the linear program is integer, then it is guaranteed to give the global optimum of the energy.

Despite this advantage, there have been very few applications of LP relaxations for solving MRF problems in vision. This can be traced to the computational complexity of LP solvers — the number of constraints and equations in LP relaxation of vision problems is simply too large. Instead, the typical algorithms used in MRF minimization for vision problems are either based on message-passing (in particular belief propagation (BP) and the tree reweighted version of belief propagation (TRW)) or on graph-cuts.

In the last five years, however, an intriguing connection has emerged between message passing algorithms, graph cut algorithms and LP relaxation. In this chapter, we give a short, introductory treatment of this intriguing connection (focusing on message passing algorithms). Specifically, we show that BP and its variants can be used to solve

LP relaxations that arise from vision problems, sometimes far more efficiently than using off the shelf LP software packages. Furthermore, we show that BP and its variants can give additional information that allows one to provably find the global minimum even when the LP relaxation is not tight.

1.1.1 Energy minimization and its Linear Programming Relaxation

The energy minimization problem and its LP relaxation were described in the introduction, and we briefly define them here again in a slightly different notation (that will make the connection to BP more transparent).

For simplicity, we discuss only MRFs with pairwise cliques in this chapter, but all statements can be generalized to higher-order cliques [15].

We work with MRFs of the form:

$$\Pr(x) = \frac{1}{Z} \prod_i \exp(-\Phi_i(x_i)) \prod_{\langle ij \rangle} \exp(-\Psi_{ij}(x_i, x_j)) \quad (1.1)$$

We wish to find the most probable configuration x^* that maximizes $\Pr(x)$ or equivalently, the one that minimizes the energy:

$$x^* = \arg \min \sum_i \Phi_i(x_i) + \sum_{ij} \Psi_{ij}(x_i, x_j) \quad (1.2)$$

For concreteness, let's focus on the stereo vision problem (figure 1.1). Here x_i will denote the disparity at a pixel i and $\Phi_i(x_i)$ will be the local, data term in the energy function while $\Psi_{ij}(x_i, x_j)$ is the pairwise, smoothness, term in the energy function. As shown in [4] for many widely used smoothness terms (e.g. the Potts model) exact minimization is NP hard. Figures 1.1b,c show the results of graph cuts and ordinary belief propagation on the Potts model energy function. In this display, the lighter a pixel is the further away it is calculated to be from the camera. Note that both graph cuts and BP calculate the depth of the hat and shirt to have "holes" - there are pixels inside the hat and the shirt whose disparity is calculated to be larger than the rest of the hat. Are these mistakes due to the energy function or the approximate minimization?

We can convert the minimization into an integer program, by introducing binary variables $q_i(x_i)$ for each pixel and $q_{ij}(x_i, x_j)$ for any pair

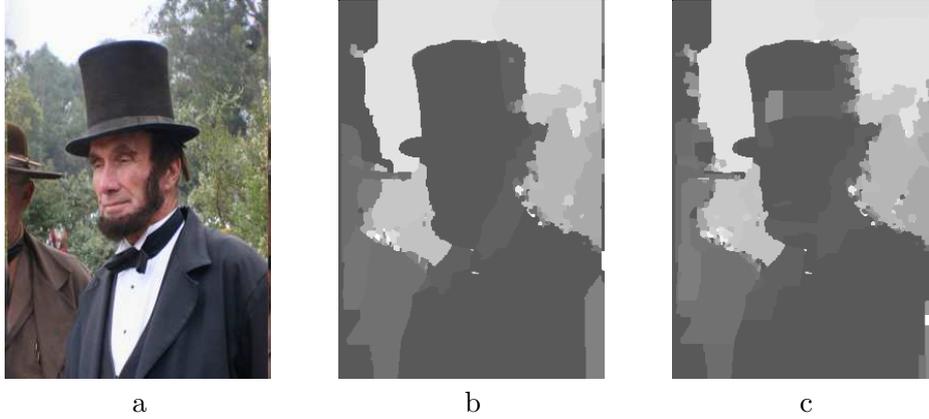


Figure 1.1 **a.** A single frame from a stereo pair. Finding the disparity is often done by minimizing an energy function. **b.** The results of graph cuts using a Potts model energy function. **c.** The results of ordinary BP using the same energy function.

of connected pixels. We can then rewrite the minimization problem as:

$$\{q_{ij}^*, q_i^*\} = \arg \min \sum_i \sum_{x_i} q_i(x_i) \Phi_i(x_i) + \sum_{\langle ij \rangle} \sum_{x_i, x_j} \Psi_{ij}(x_i, x_j) q_{ij}(x_i, x_j) \quad (1.3)$$

The minimization is done subject to the following constraints:

$$\begin{aligned} q_{ij}(x_i, x_j) &\in \{0, 1\} \\ \sum_{x_i, x_j} q_{ij}(x_i, x_j) &= 1 \\ \sum_{x_i} q_{ij}(x_i, x_j) &= q_j(x_j) \end{aligned}$$

where the last equation enforces the consistency of the pairwise indicator variables with the singleton indicator variable.

This integer program is completely equivalent to the original MAP problem, and is hence computationally intractable. We can obtain the linear programming relaxation by allowing the indicator variables to take on non-integer values. This leads to the following problem:

The LP relaxation of Pairwise Energy Minimization:

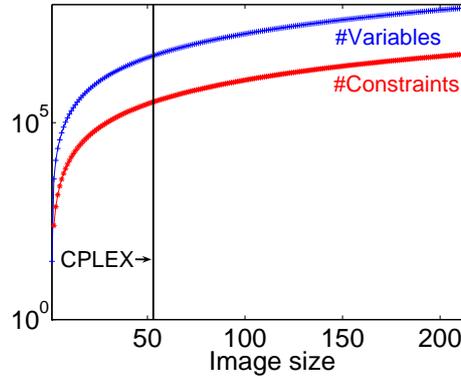


Figure 1.2 The number of variables and constraints in a stereo problem with 30 disparities as a function of image size. Even modestly sized images have millions of variables and constraints. The largest image that could be solved with commercial LP software on a machine with 4GB of memory in [16] is approximately 50×50 .

minimize:

$$J(\{q\}) = \sum_{\langle ij \rangle} \sum_{x_i, x_j} q_{ij}(x_i, x_j) \Psi_{ij}(x_i, x_j) + \sum_i \sum_{x_i} q_i(x_i) \Psi_i(x_i) \quad (1.4)$$

subject to:

$$q_{ij}(x_i, x_j) \in [0, 1] \quad (1.5)$$

$$\sum_{x_i, x_j} q_{ij}(x_i, x_j) = 1 \quad (1.6)$$

$$\sum_{x_i} q_{ij}(x_i, x_j) = q_j(x_j) \quad (1.7)$$

This is now a linear program – the cost and the constraints are linear. It can therefore be solved in polynomial time and we have the following guarantee:

Observation If the solutions $\{q_{ij}(x_i, x_j), q_i(x_i)\}$ to the MAP LP relaxation are all *integer*, that is $q_{ij}(x_i, x_j), q_i(x_i) \in \{0, 1\}$, then $x_i^* = \arg \max_{x_i} q_i(x_i)$ is the MAP.

1.1.2 The need for special purpose LP solvers

Having converted the energy minimization problem to a linear program (LP), it may seem that all we need is to use off the shelf LP solvers and apply them to computer vision problems. However, by relaxing the problem we have increased the size of the problem tremendously – there are much more variables in the LP than there are nodes in the original graph.

Formally, denote by k_i the number of possible states of node i . The number of variables and constraints in the LP relaxation is given by:

$$\begin{aligned} N_{variables} &= \sum_i k_i + \sum_{\langle i,j \rangle} k_i k_j \\ N_{constraints} &= \sum_{\langle i,j \rangle} (k_i + k_j + 1) \end{aligned}$$

The additional $\sum_{\langle i,j \rangle} 2k_i k_j$ bound constraints, derived from equation (1.5), are usually not considered part of the constraint matrix.

Figure 1.2 shows the number of variables and constraints as a function of image size for a stereo problem with 30 disparities. If the image is a modest 200×200 pixels and each disparity can take on 30 discrete values, then the LP relaxation will have over 72 million variables and four million constraints. The vertical line shows the largest size image that could be solved using a commercial powerful LP solver (CPLEX 9.0) using a desktop machine with 4GB of memory in [16]. Obviously, we need a solver that can somehow take advantage of the problem structure in order to deal with such a large-scale problem.

1.2 Ordinary Sum-Product Belief Propagation and Linear Programming

The sum-product belief propagation (BP) algorithm was introduced by Pearl [11] as a method for performing exact probabilistic calculations on singly connected MRFs. The algorithm receives as input a graph G and the functions $F_{ij}(x_i, x_j) = \exp(-\Psi(x_i, x_j))$, $F_i(x_i) = \exp(-\Psi_i(x_i))$. At each iteration, a node x_i sends a message $m_{ij}(x_j)$ to its neighbor in the graph x_j . The messages are updated as follows:

$$m_{ij}(x_j) \leftarrow \alpha_{ij} \sum_{x_i} F_{ij}(x_i, x_j) F_i(x_i) \prod_{k \in N_i \setminus j} m_{ki}(x_i) \quad (1.8)$$

where $N_i \setminus j$ refers to all neighbors of node x_i except x_j . The constant α_{ij} is a normalization constant typically chosen so that the messages sum to one (the normalization has no influence on the final beliefs). Given the messages, each node can form an estimate of its local “belief” defined as:

$$b_i(x_i) \propto F_i(x_i) \prod_{j \in N_i} m_{ji}(x_i) \quad (1.9)$$

and every pair of nodes can calculate their “pairwise belief”:

$$b_{ij}(x_i, x_j) \propto F_i(x_i) F_j(x_j) F_{ij}(x_i, x_j) \prod_{k \in N_i \setminus j} m_{ki}(x_i) \prod_{k \in N_j \setminus i} m_{kj}(x_j) \quad (1.10)$$

Pearl showed that when the MRF graph is singly-connected, the algorithm will converge and these pairwise beliefs and singleton beliefs will exactly equal the correct marginals of the MRF (i.e. $b_i(x_i) = \Pr(x_i)$, $b_{ij}(x_i, x_j) = \Pr(x_i, x_j)$). But when there are cycles in the graph, neither convergence nor correctness of the beliefs is guaranteed. Somewhat surprisingly, however, for any graph (with or without cycles) there is a simple relationship between the BP beliefs and the LP relaxation.

In order to show this relationship, we need to define the BP algorithm at temperature T . This is exactly the same algorithm defined above (equation 1.8) and the only difference is the definition of the local functions $F_{ij}(x_i, x_j)$, $F_i(x_i)$. The new definition depends both on the energy function parameters $\Psi_{ij}(x_i, x_j)$, $\Phi_i(x_i)$ as well as a new parameter T which we call “temperature”.

$$F_{ij}(x_i, x_j) = \exp -\frac{1}{T} \Psi_{ij}(x_i, x_j) \quad (1.11)$$

$$F_i(x_i) = \exp -\frac{1}{T} \Phi_i(x_i) \quad (1.12)$$

Observation: For any MRF as $T \rightarrow 0$ there exists a fixed-point of ordinary BP at temperature T whose beliefs approach the LP solution.

This observation follows directly from the connection between BP and the Bethe Free Energy [18]. As explained in the previous chapter, there is a one to one correspondence between the fixed points of BP at temperature T and stationary points of the following problem.

The Bethe Free Energy Minimization Problem: Minimize:

$$G(\{b\}; T) = \sum_{\langle ij \rangle} \sum_{x_i, x_j} b_{ij}(x_i, x_j) \Psi_{ij}(x_i, x_j) + \sum_i \sum_{x_i} b_i(x_i) \Phi_i(x_i) - T \left(\sum_{ij} H(b_{ij}) + \sum_i (1 - \text{deg}_i) H(b_i) \right) \quad (1.13)$$

subject to:

$$b_{ij}(x_i, x_j) \in [0, 1] \quad (1.14)$$

$$\sum_{x_i, x_j} b_{ij}(x_i, x_j) = 1 \quad (1.15)$$

$$\sum_{x_i} b_{ij}(x_i, x_j) = b_j(x_j) \quad (1.16)$$

where $H(b_i)$ is the Shannon Entropy of the belief $H(b_i) = -\sum_{x_i} b_i(x_i) \ln b_i(x_i)$ and deg_i is the degree of node i in the graph.

Comparing the Bethe Free Energy Minimization problem and the LP relaxation problem, we see that the constraints are the same, and the first term in the objective is also the same. The only difference is the existence of additional, entropy, terms in the Bethe Free Energy. But these terms are multiplied by T so that as $T \rightarrow 0$ the two problems coincide (recall that the Shannon entropy is bounded).

Figure 1.3 illustrates the convergence of the Bethe Free Energy to the LP relaxation. We consider a graphical model corresponding to a toroidal grid. The nodes are binary and all the pairwise potentials are of the form:

$$F = \begin{pmatrix} 3 & 1 \\ 1 & 2 \end{pmatrix}$$

These potentials correspond to an Ising model with a uniform external field – nodes prefer to be similar to their neighbors and there is a preference for one state over the other. In order to visualize the approximate free energies, we consider beliefs that are symmetric and identical for all pairs of nodes:

$$b_{ij} = \begin{pmatrix} x & y \\ y & 1 - (x + 2y) \end{pmatrix}$$

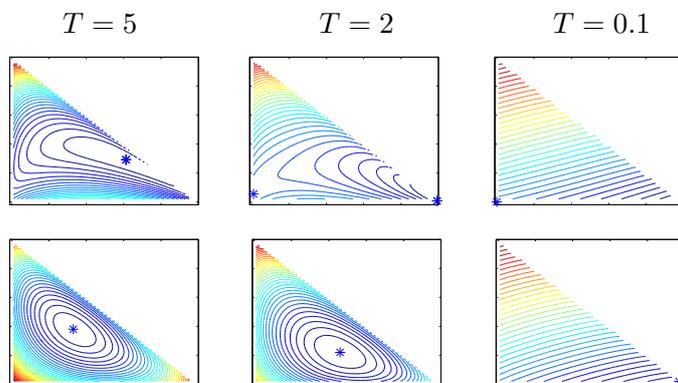


Figure 1.3 Contour plots of the Bethe free energy (top) and a convex free energy (bottom) for a 2D Ising model with uniform external field at different temperatures. The stars indicate local stationary points. Both free energies approach the LP as temperature is decreased, but for the Bethe free energy, a local minimum is present even for arbitrarily small temperatures.

Note that the MAP (and the optimum of the LP) occur at $x = 1, y = 0$ in which case all nodes are in their preferred state. Figure 1.3 shows the Bethe free energy (top) for this problem for different temperatures. At high temperature, the minimization problems are quite different, but as temperature is decreased the Bethe Free Energy is dominated by the linear term and becomes equivalent to the LP relaxation.

Note, however, that the convergence of the Bethe Free Energy problem to the LP relaxation, does not guarantee that any BP fixed-point will solve the LP relaxation as the temperature approaches zero. It only guarantees that there exists a good fixed-point, but there may be other fixed-points as well. The stars in figure 1.3 indicate the local stationary points of the Bethe Free Energy. A bad local minimum exists for small temperatures at $x = 0, y = 0$. This corresponds to a solution where all nodes have the same state but it is not the preferred state.

1.2.1 Convex BP and the LP Relaxation

In order to avoid local minima, we need a version of the Bethe Free Energy that has a unique stationary point. The question of when the Bethe Free Energy has a unique stationary point is surprisingly delicate (see e.g. [8, 12–14]) and can depend non-trivially on the graph and the energy function. Perhaps the simplest condition that guarantees a unique stationary point is *convexity*. As illustrated in figure 1.4a a 1D function is convex, if its second derivative is always positive and this

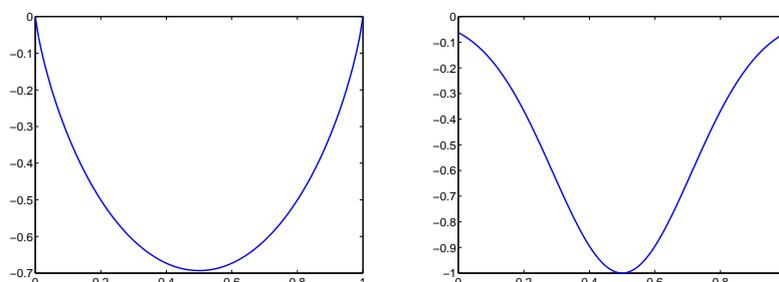


Figure 1.4 Two 1D functions defined on the range $[0, 1]$. The function on the left is the negative Shannon entropy and is convex. The function on the right (an inverted Gaussian) has a unique stationary point but is not convex. In order to guarantee uniqueness of BP fixed points, we seek free energies that are convex.

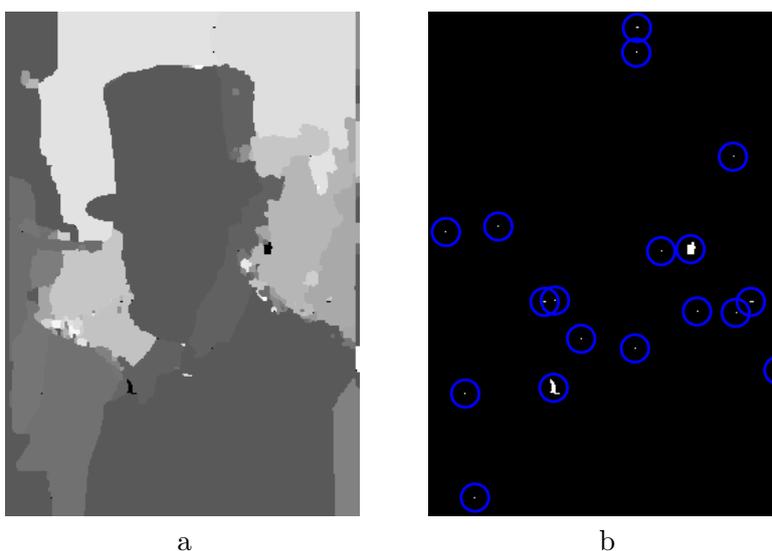


Figure 1.5 **a.** The solution to the LP obtained by running convex BP. Using convex BP we can solve the LP relaxation for full sized images. Pixels for which the LP solution is fractional are shown in black. **b.** A binary image indicating in white the pixels for which the LP solution is fractional.

guarantees that it has a unique stationary point. Convexity is a sufficient condition for uniqueness of stationary points but is not necessary. Figure 1.4b shows a 1D function that has a unique stationary point, but is not convex. Nevertheless, the easiest way to guarantee uniqueness of BP fixed-points is to require convexity of the free energy.

In dimensions larger than one, the definition of convexity simply requires positivity of the Hessian. This means that the convexity of the free energy does *not* depend on the terms in the energy function $\Psi_{ij}(x_i, x_j), \Phi_i(x_i)$. These terms only changes the linear term in the free energy and do not influence the Hessian. Thus the free energy will be convex if the sum of entropy terms are convex. This sum of entropies $H_\beta = \sum_{ij} H(b_{ij}) + \sum_i (1 - \deg_i) H(b_i)$ is called the Bethe entropy approximation. The negative Bethe entropy can be shown to be convex when the graph is a tree or has a single cycle. However, when the graph has multiple cycles, as in the toroidal grid discussed earlier, the Bethe negative entropy is not convex and hence BP can have many fixed-points.

We can avoid this problem by “convexifying” the Bethe entropy. We consider a family of entropy approximations of the form:

$$\tilde{H} = \sum_{ij} c_{ij} H(b_{ij}) + \sum_i c_i H(b_i) \quad (1.17)$$

Heskes [8] has shown that a sufficient condition for such an approximate entropy to be convex is if it can be rewritten as a *positive* combination of three types of terms: (1) pairwise entropies (e.g. $H(b_{ij})$) (2) singleton entropies (e.g. $H(b_i)$) and (3) conditional entropies (e.g. $H(b_{ij}) - H(b_i)$). Thus the Bethe entropy for a chain of three nodes will be convex since it can be written $H_\beta = H_{12} + H_{23} - H_2$ which is a positive combination of a pairwise entropy H_{12} and a conditional entropy $H_{23} - H_2$. However, for the toroidal grid discussed above, the Bethe entropy cannot be written in such a fashion. To see this, note that a 3x3 toroidal grid has nine nodes with degree 4 and 18 edges. This means that the Bethe entropy will have 27 negative entropy terms (i.e. nine times we will subtract $3H_i$). However, the maximum number of negative terms we can create with conditional entropies is 18 (the number of edges) so we have more negative terms than we can create with conditional entropies. In contrast, the entropy approximation $\sum_{\langle ij \rangle} (H_{ij} - H_i)$ is convex, since it is the sum of 18 conditional entropies.

Given an approximate entropy that satisfies the convexity conditions, we can replace the Bethe entropy with this new convex entropy and obtain a convex free energy. But how can we minimize it? It turns out that a slight modification to the BP update rules gives a new algorithm whose fixed-points are stationary points of any approximate free energy. The algorithm defines an extra scalar variable, for each node i : $\rho_i =$

$\frac{1}{c_i + \sum_{j \in N_i} c_{ij}}$ and for each edge ij $\rho_{ij} = \rho_j c_{ij}$ (note that ρ_{ij} may be different than ρ_{ji}). Using these extra scalars the update equations are:

$$m_{ij}(x_j) \leftarrow \sum_{x_i} F_{ij}^{\frac{1}{c_{ij}}}(x_i, x_j) F_i^{\rho_i}(x_i) \cdot \prod_{k \in N_i \setminus j} m_{ki}^{\rho_{ki}}(x_i) m_{ji}(x_i)^{\rho_{ji} - 1} \quad (1.18)$$

$$b_i(x_i) = F_i^{\rho_i}(x_i) \prod_{j \in N(i)} m_{ji}^{\rho_{ji}}(x_i) \quad (1.19)$$

$$b_{ij}(x_i, x_j) = F_{ij}(x_i, x_j)^{\frac{1}{c_{ij}}} F_i^{\rho_i}(x_i) F_j^{\rho_j}(x_j) \prod_{k \in N_i \setminus j} m_{ki}^{\rho_{ki}}(x_i) m_{ji}(x_i)^{\rho_{ji} - 1} \prod_{k \in N_j \setminus i} m_{kj}^{\rho_{kj}}(x_j) m_{ij}(x_j)^{\rho_{ij} - 1} \quad (1.20)$$

Note that this algorithm is very similar to ordinary BP, so it requires very little modification to an existing implementation of BP. In particular, we can use algorithms for efficient calculations of BP messages for certain energy function (e.g. [6]). Note that for the Bethe Free Energy, $c_{ij} = 1$ and $c_i = 1 - \deg_i$ (and thus $\rho_i, \rho_{ij} = 1$) the above update equation reduces to ordinary BP. However, by choosing c_{ij}, c_i so that the approximate free energy is convex, we can guarantee that this modified algorithm will have a single, unique fixed-point at any temperature.

Returning to the toroidal grid we discussed earlier, figure 1.3 (bottom) shows the convexified free energy (with an entropy approximation of the form $18H_{12} - 18H_1$) for this problem for different temperatures. As was the case for the Bethe Free Energy, at high temperature, the minimization problems are quite different, but as temperature is decreased the free energy is dominated by the linear term and becomes equivalent to the LP relaxation. However, unlike the Bethe free energy, the convex free energy always has a unique minimum (indicated by the star) so that the fixed point of the generalized BP algorithm is guaranteed to give the LP solution.

An important special case of a convex free energy are the class of “tree reweighted” (TRW) free energies. In these free energies the entropy is approximated as a linear combination of entropies over trees $\tilde{H} = \sum_{\tau} \rho_{\tau} H_{\tau}$. For this free energy, the generalized BP algorithm reduces to the TRW algorithm (since $\rho_i = 1$ and $\rho_{ij} = \rho_{ji}$ in this case).

To summarize, by choosing a convex free energy and running the generalized BP algorithm at small temperature we can approximate the LP

solution. Returning to the stereo problem depicted in figure 1.1, even though standard LP solvers fail on such a large problem, convex BP solved it in less than two hours. The results are shown in figure 1.5a,b. In figure 1.5a we display the disparity encoded by the LP solution. If the LP solution was indeed integer, we display the disparity for which the LP solution was nonzero. If the LP solution was fractional, that pixel is shown in black. In figure 1.5b we show a binary mask indicating which pixels had non-integer values in the LP solution. The existence of such pixels means that the LP relaxation is not tight.

1.3 Convex Max-Product BP

Although we have shown that one can use sum-product convex BP to solve the linear program, one needs to be able to run the sum-product algorithm at sufficiently small temperatures. There are two problems with this approach. First, running the algorithm at small temperatures requires defining $F_{ij} = \exp(-\Psi_{ij}(x_i, x_j)/T)$, $F_i(x_i) = \exp(-\Phi_i(x_i)/T)$ for small T . Note that as $T \rightarrow 0$ we are dividing by a number that approaches zero and this can cause numerical problems. In the appendix, we discuss how to implement the algorithm in “log space”, i.e. by working with the logarithms of the potentials and messages. This can greatly increase the numerical precision at low temperatures.

A second problem, however, is that it is not obvious how to choose the temperature so that it is “sufficiently small”. As evident by our discussion in the previous section, we need the temperature to be small enough so that the entropy contribution is negligible relative to the the average energy. So the requirement of the temperature being “sufficiently small” is problem dependent – as we change terms in the energy function the scale of the average energy may change as well requiring a different temperature.

In order to avoid choosing a “sufficiently small” temperature, we can work with the “zero temperature limit” of the convex BP algorithm. This algorithm, called the max-product convex BP algorithm is exactly the same as equations 1.18 but with the “sum” operator replaced with a “max”. It is easy to show that as the temperature T approaches zero, the update equations of the sum-product algorithm at temperature T approach those of the max-product algorithm at $T = 1$. Formally, if a set of messages form a fixed-point of sum-product at temperature T . As $T \rightarrow 0$, then these same messages raised to the $1/T$ power, will form a fixed-point of the max-product algorithm. This proof follows from the fact that the ℓ_p norm approaches the max norm as $p \rightarrow \infty$ [7, 15].

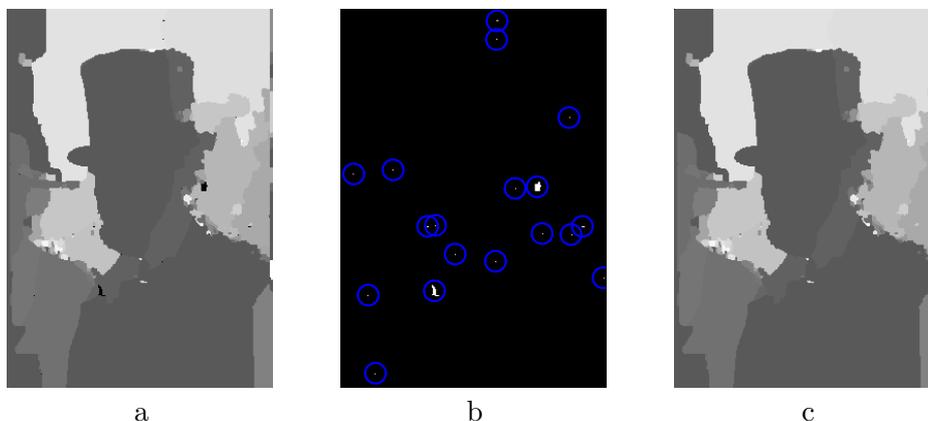


Figure 1.6 **a** Results of using convex max product BP on the stereo problem shown in figure 1.1a. Pixels for which there are ties in the belief are shown in black. **b.** A binary image indicating which pixels had ties. Note that these are exactly the same pixels for which the LP solution had non-integer values (see figure 1.5). **c** The *global* optimum found by resolving the tied pixels and verifying that the conditions for optimality hold. Note that this solution is not much better than the local optima found before (figure 1.1). Both the hat and the shirt of the foreground person are calculated to have “holes”.

Despite this direct connection to the sum-product algorithm, the max-product algorithm is more difficult to analyze. In particular, even for a convex free energy approximation, the max-product algorithm may have multiple fixed points even though the sum-product algorithm has a unique fixed-point. Thus one cannot guarantee that any fixed-point of convex max-product BP will solve the linear programming relaxation.

An important distinction in analyzing the fixed-points of max-product BP is the notion of “ties”. We say that a belief at a node has a tie if it does not have a unique maximizing value. Thus a belief of the form $(0.7, 0.2, 0.1)$ has no ties while the belief $(0.4, 0.4, 0.2)$ has a tie. Max-product fixed points without ties can be easily shown to indeed correspond to a limit of the sum product algorithm at zero temperature. This leads to the following result.

Claim: If max-product convex BP converges to a fixed-point without ties, then the assignment $x_i^* = \arg \max_{x_i} b_i(x_i)$ is the global minimum of the energy function.

This result is analogous to the claim on the LP relaxation. Only if the LP relaxation ends up being integral can we say that it corresponds to the global minimum of the energy.

Unfortunately, in many vision problems neither is the LP all integer nor are the max-product beliefs all tied. A typical example is shown in figure 1.6a where we have indicated in black the pixels for which ties exist (these same pixels are, not coincidentally, the pixels where the LP solution is non-integral). In all the non-tied pixels we have shown the disparity that maximizes the beliefs at the fixed-point. It can be seen that a small number of pixels are black (see also the mask of black pixels shown in figure 1.6b), so that we cannot guarantee optimality of the solution. Yet the disparities at the non-tied pixels seem reasonable. Under what conditions can we “trust” the values in the non-tied pixels?

In recent years, a number of results have been obtained that allow us to still prove partial optimality of an assignment obtained by maximizing the belief at a non-tied node after running max-product convex BP. Partial optimality means that we can fix the values at the non-tied nodes and only optimize over the remaining, tied, nodes. Under certain conditions, this procedure can still be guaranteed to find the global optimum.

We list here some results on partial optimality and refer the reader to [15, 16] for more exact definitions and proofs.

- When each node has only two possible states, partial optimality holds.
- Consider the subgraph formed by looking only at the tied nodes. If this graph is a tree partial optimality holds.
- Consider the subgraph formed by looking only at the tied nodes. Define its boundary as those nodes in the subgraph that are also connected to other nodes. If the beliefs at the boundary nodes is uniform then partial optimality holds.
- Consider the subgraph formed by looking only at the tied nodes. Define a new energy function on this subgraph and find the assignment in the tied nodes that minimizes this energy function. If that assignment does not contradict the beliefs at the boundary of the subgraph, then partial optimality holds.

Note that verifying that partial optimality holds may require additional computation after running max-product convex BP. Yet in many vision problems, we have found that this verification can be done efficiently and this allows us to provably find the global optimum of the energy function. Code implementing these verification steps is available at <http://www.cs.huji.ac.il/~talyam/stereo.html>. Figure 1.6c shows

the *global* optimum of the energy function for the image shown in figure 1.1. Although this is the global optimum for the energy function, it still suffers from mistakes. In particular, the calculated depth for the hat and the shirt still has holes. This indicates that a crucial part of stereo research is choosing a good energy function to minimize.

1.4 Discussion

Despite the NP-hardness of energy minimization in many computer vision problems, it is actually possible to find the global optimum of the energy in many instances. Theoretically, this could be done by relaxing the problem into a linear program. However, the large number of variables and constraints makes this linear program unsuitable for standard LP solvers. In this chapter, we have reviewed how variants of belief propagation can be used to solve the LP relaxation. Furthermore, we have shown how the max-product convex BP algorithm can be used to find the global optimum even if the LP relaxation is not tight.

While we have focused on the connection between BP and the LP relaxation, it can also be shown that the alpha expansion graph cut algorithm is also intimately connected to the same LP relaxation. In particular, Komodakis and Tziritas [10] have shown that the alpha expansion algorithm can be seen as an iterative “primal integer-dual” algorithm for solving the LP relaxation. Thus the graph cuts algorithm and BP, which are often seen as competing algorithms, are actually closely related. One important conclusion from this relationship, is that both algorithms are not expected to work well when the LP relaxation is loose. Indeed, despite the success recounted here in finding global optima for some energy functions in stereo vision, for other energy functions the number of “tied” pixels is far too large for the methods described here to be successful. Understanding the conditions under which energy minimization problems in computer vision have a tight LP relaxation is a promising direction for future research.

Acknowledgements

Supported by the Israeli Science Foundation. We thank Danny Rosenberg for his help in generating the figures.

1.5 Appendix - implementation details

1.5.1 Implementation in log space

To be able to run the algorithm with a small temperature T , we use the log-space – that is, we work directly with the costs/energies Φ_i, Ψ_{ij} instead of the potentials F_i, F_{ij} , and a set of messages $n_{ji}(x_i) = -T \log m_{ji}(x_i)$.

Yet when running sum-product, we need to sum over terms which are the exponent of the log-terms calculated, and then take the log again. Thus rewriting the generalized BP updates in log space gives:

$$n_{ji}(x_i) = -T \log \sum_{x_j} \exp \left(- \left(\Phi_j(x_j) + \frac{\Psi_{ij}(x_i, x_j)}{\rho_{ij}} + \sum_{k \in N_j \setminus i} \rho_{jk} n_{kj}(x_j) - (1 - \rho_{ij}) n_{ij}(x_j) \right) / T \right) \quad (1.21)$$

For efficiency and numerical stability, we use the following equality:

$$\log(\exp(x) + \exp(y)) = x + \log(1 + \exp(y - x)) \quad (1.22)$$

for $x \geq y$. In particular, when x is much greater than y we can ignore the second term and avoid exponentiating or taking the logarithm during the message update.

1.5.2 Efficient message computation for Potts-model

Calculating the vector $m_{ji}(x_i)$ is actually performing a matrix-vector multiplication:

$$m_{ji} = A_{ij} \cdot y_j \quad (1.23)$$

Where A_{ij} is a matrix and y_j is a vector. In the case of the generalized BP update the matrix and vector are given by: $A_{ij}(x_i, x_j) = F_{ij}^{1/\rho_{ij}}(x_i, x_j)$ and $y_j(x_j) = F_j(x_j) \prod_{k \in N_j \setminus i} m_{kj}^{\rho_{jk}}(x_j) \cdot m_{ij}^{\rho_{ij}-1}(x_j)$.

We consider the case where the pairwise-potentials are of Potts model, and thus $A_{ij} = (a_{ij} - b_{ij}) \cdot I + b_{ij}$. Thus, we obtain:

$$A_{ij} y_j = (a_{ij} - b_{ij}) y_j + b_{ij} \sum_{x_j} y_j(x_j) \quad (1.24)$$

Note that this way, we could compute the outgoing messages vector m_{ji} in $O(|X_j| + |X_i|)$ complexity: one loop of $O(|X_j|)$ for computing the sum $S_j = \sum_{x_j} y_j(x_j)$, and another loop of $O(|X_i|)$ for computing the value:

$$m_{ji}(x_i) = (a_{ij} - b_{ij})y_j(x_i) + b_{ij}S_j \quad (1.25)$$

for each assignment x_i .

Bibliography

- [1] F. Barahona. On the computational complexity of ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10), 1982.
- [2] F. Barahona, M. Junger, and G. Reinelt. Experiments in quadratic 01 programming. *Mathematical Programming*, 44:127–137, 1989.
- [3] D. Bertismas and J. Tsitsikilis. *Introduction to linear optimization*. Athena Scientific, 1997.
- [4] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*, 1999.
- [5] J. Feldman, D. Karger, and M. J. Wainwright. LP decoding. In *Allerton Conference on Communication, Control, and Computing*, 2003.
- [6] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. In *Proceedings of IEEE CVPR*, pages 261–268, 2004.
- [7] Tamir Hazan and Amnon Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energies. In *UAI*, pages 264–273, 2008.
- [8] T. Heskes. Convexity arguments for efficient minimization of the Bethe and Kikuchi free energies. *Journal of AI Research*, 26:153–190, 2006.
- [9] C. L. Kingsford, B. Chazelle, and M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics*, 21(7):1028–1039, 2005.

- [10] Nikos Komodakis and Georgios Tziritas. Approximate labeling via graph cuts based on linear programming. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(8):1436–1453, 2007.
- [11] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.
- [12] Tanya Roosta, Martin J. Wainwright, and Shankar S. Sastry. Convergence analysis of reweighted sum-product algorithms. *IEEE Transactions on Signal Processing*, 56(9):4293–4305, 2008.
- [13] Sujay Sanghavi, Devavrat Shah, and Alan S. Willsky. Message passing for max-weight independent set. In *NIPS*, 2007.
- [14] Sekhar Tatikonda and Michael I. Jordan. Loopy belief propagation and gibbs measures. In *UAI*, pages 493–500, 2002.
- [15] Y. Weiss, C. Yanover, and T. Meltzer. Map estimation, linear programming and belief propagation with convex free energies. In *UAI*, 2007.
- [16] C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation – an empirical study. *Journal of Machine Learning Research*, 7:1887–1907, 2006.
- [17] C. Yanover and Y. Weiss. Approximate inference and protein folding. *Advances in Neural Information Processing Systems*, 2002.
- [18] J.S. Yedidia, W.T. Freeman, and Yair Weiss. Constructing free energy approximations and generalized belief propagation algorithms. *IEEE Transactions on Information Theory*, 51(7):2282–2312, 2005.