On the Convergence of the Convex Relaxation Method and Distributed Optimization of Bethe Free Energy

Ming Su
Department of Electrical Engineering and Department of Statistics
University of Washington, Seattle, WA
mingsu@u.washington.edu

Abstract

Exact probabilistic inference in graphical models can be computationally intractable. One often resorts to approximate inference methods. We give a convergence analysis of a convex relaxation method for approximate inference. This method is a natural generalization of Unified Propagation and Scaling (UPS) algorithm. We derive sufficient conditions for the method's convergence from a mathematical programming point of view. We also propose a distributed implementation of the method and show how the synchronization cost can be minimized at the algorithmic level.

1 Introduction

Exact probabilistic inference in graphical models can be computationally intractable, indeed, NP-hard (Cooper, 1990). The complexity of most exact inference algorithms is exponential in the tree-width of the graph. For graphs with large tree-width, one has to resort to approximate methods. As extensions of the exact Belief Propagation (BP) algorithm on trees, loopy and generalized Belief Propagation (Yedidia, Freeman, & Weiss 2000) work well in practice without guaranteeing convergence in general. Mooij and Kappen recently derived strong sufficient condition for the convergence of parallel loop BP by showing that the algorithmic mapping is a contraction (Mooij & Kappen 2005). There have been many efforts towards general convergent BP-like algorithms, such as Concave-Convex Procedure (CCCP) (Yuille & Rangarajan 2003), UPS algorithm (Teh & Welling 2001b), double-loop convex upper-bounding algorithms (Heskes, Albers, & Kappen 2003; Heskes 2006), variational message passing (Winn & Bishop 2005), oriented tree decomposition algorithm (Globerson & Jaakkola 2007), convex message passing algorithm (Hazan & Shashua 2008) and Tree Reweighted (TRW) algorithm (Wainwright, Jaakkola, & Willsky 2002). Meltzer et al. recently showed that TRW is convergent (Meltzer, Globerson, & Weiss 2009).

The convergence property of such iterative methods has significant impacts on both theory and application sides. UPS algorithm has gained popularity due to its reasonably good performance and easiness to implement (Carbonetto, de Freitas, & Barnard 2004; Xie, Gao, & Wu 2009). In this paper, we give a convergence analysis of a convex relaxation method, which is a natural generalization of UPS algorithm. Although Teh and Welling have indicated that UPS algorithm is convergent (Teh & Welling 2001b), there is no proof given and the convergence result does not follow from the preceding contents therein either. The proofs given in (Teh & Welling 2001a; Welling & Teh 2001) lack key components that guarantee global convergence. We give an alternative but complete proof for a weaker sufficient condition for the convergence, from a mathematical programming point of view.

While it has been demonstrated empirically that loopy and generalized BP work extremely well in many applications, such as Turbo decoding, observation of divergence has been documented as well (Botetz 2007). Yedidia et al. have shown that these methods are not guaranteed to converge for graphs with cycles(Yedidia, Freeman, & Weiss 2000; 2005). Approximate inference methods can be understood as nonconvex constrained minimization of the so-called Bethe free energy. Yedidia et al. has shown that fixed points of these iterative methods are stationary points of Bethe free energy and these methods are guaranteed to converge to some fixed point for singly-connected graphs (Yedidia, Freeman, & Weiss 2000; 2005). UPS algorithm, developed by Teh and Welling, is a convergent alternative to loopy BP. Teh and Welling have demonstrated that UPS algorithm works at least as well as loopy BP in practice (Teh & Welling 2001b). The algorithm iterates through a sequence of strictly convex subproblems. Each subproblem is solved by Iterative Scaling (IS) algorithm, which is exact and fast. UPS algorithm can be generalized by allowing any exact algorithm, including IS, to be used in solving the subproblems. We refer to this generalization as the convex relaxation method.

The convex relaxation method can be viewed as a block coordinate descent (BCD) method for constrained optimization (Teh & Welling 2001b; Heskes 2006). It is known that in general, BCD methods may not converge to any stationary point of the problem (Bertsekas 1999). In a well-known example given in (Powell 1973), the sequence of iterates cycles through several nonstationary points in the limit. The reason for such a behavior is that at each iteration the gradient of the objective function becomes progressively closer to being orthogonal to the coordinate search direction (Bertsekas 1999;

Nocedal & Wright 1999). In addition, coordinate blocks for the convex relaxation method overlap with each other. In contrast to BCD methods with orthogonal blocks, such methods have no general convergence results. Any particular method of such type needs to be treated specifically. An exmaple in statistics can be found in the work by Drton and Eichler (Drton & Eichler 2006), where an overlapping coordianate search is used in maximum likelihood estimation. Drton and Eichler have provided specific proof for the convergence of the algorithm.

Under some convexity assumption of the problem, one can ensure the convergence of the block-nonlinear Gauss-Seidel (GS) method, where coordinate blocks are orthogonal to each other, in both constrained and unconstrained case (Bertsekas 1999; Luo & Tseng 1992; L. Grippo 2000). Consider the following problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & x \in X = X_1 \times \cdots \times X_m \subseteq \mathbb{R}^n, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is continuously differentiable and the feasible set X is the Cartesian product of closed, nonempty and convex subsets $X_i \subseteq \mathbb{R}^{n_i}$, for $i = 1, \ldots, m$, and $\sum_{i=1}^{m} n_i = n$. Let $x \in \mathbb{R}^n$ be partitioned into m components with $x_i \in \mathbb{R}^{n_i}$. A naturally defined block GS method for (1) has the following update rule:

$$x_i^{k+1} = \arg\min_{y_i \in X_i} f(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, y_i, x_{i+1}^{k}, \ldots, x_m^{k}).$$

The result in (Bertsekas 1999; L. Grippo 2000) states that for (1), any cluster point of the sequence $\{x^k\}$ generated by the block GS method is a stationary point of (1), under the assumption that $\{x^k\}$ has cluster point and $f(x)$ is component-wise strictly convex. This result is of particular interests to us because in the convex relaxation method, the Bethe free energy function is component-wise strictly convex with respect to each coordinate block. However the applicability of this result is compromised by the fact that the coordinate blocks in the convex relaxation method overlap with each other.

We first prove some linear-algebraic properties of the constraint set. Together with a mild assumption that all clique potentials are positive, these properties help us obtain desired convergence results even in the presence of the overlapping of the coordinate blocks. Then we show how to distribute the optimization task to multiple processors by using the idea of graph partitioning and how the synchronization costs can be minimized at the algorithmic level by using the convexity. The rest of the paper is organized as follows. In Section 2, we review the background of Bethe optimization and prove the properties of the constraint set. In Section 3, we present the convex relaxation method and establish some basic understanding of the method. In Section 4, we prove the major convergence results. In Section 5, we demonstrate the distributed convex relaxation method. Some discussion is given in Section 6.

2 Bethe Approximation

In the rest of the paper, we focus on Markov random fields. No generality is lost because there always exists equivalent transformation between Markov random field and other types of graphical models, such as Bayesian network and factor graph (Kschischang, Frey, & Loeliger 2001). Formally speaking, a Markov random field (also known as Markov network) is a triple $(\mathcal{G}, X, \Psi)$, where $\mathcal{G} = (V, E)$ is a undirected graph and X is a discrete random vector indexed by V. Ψ is the set of potential functions $\Psi_c(X_c) : S_c \to \mathbb{R}$, where c is a clique in $\mathcal{G}$ and S_c is the sample space of the random vector X_c. The joint probability distribution of X factorizes over the cliques of $\mathcal{G}$:

$$p(X) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \Psi_c(X_c)$$

where Z is the normalization constant (also known as partition function) and $\mathcal{C}$ denotes the set of all cliques in the graph. Some inference problem computes the partition function and marginal distributions over subsets of X, which requires summation over an exponential number of states. In variational methods, the inference problem is cast to minimization of the so-called free energy function defined as follows:

$$\begin{aligned} F(p) &= -\sum_{\alpha \in \mathcal{C}} \sum_{X_\alpha} p(X_\alpha) \log \psi_\alpha(X_\alpha) + \sum_X p(X) \log p(X) \\ &\equiv E(p) - S(p), \end{aligned}$$

where $E(p)$ and $S(p)$ are referred as energy and entropy, respectively. The global optimal solution yields true marginal probability distributions and the partition function. However this will not work in practice because the entropy term still requires summing over an exponential number of states.

Bethe free energy is an approximation to the free energy function. Indeed, in the Bethe approximation a sum of marginal entropies is used to replace the original exponential entropy. The entropy approximation is defined as follows:

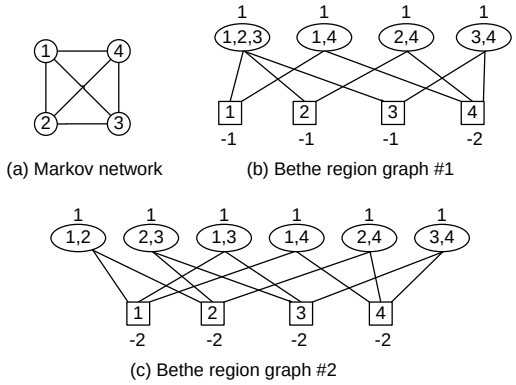
$$\begin{aligned} -S(p) &\equiv \sum_X p(X) \log p(X) \approx \sum_{\gamma \in \mathcal{R}} c_\gamma S_\gamma(p) \\ &= \sum_{\gamma \in \mathcal{R}} c_\gamma \sum_{X_\gamma} p(X_\gamma) \log p(X_\gamma) \end{aligned}$$

Here $\mathcal{R}$ denotes a collection of so-called *regions* and the parameters c_γ is called *Moebius number* or *overcounting number*. A region γ is a subset of nodes and the random variables associated with it is denoted by X_γ. There are two types of regions, namely, outer regions and inner regions. If the nodes associated with a region α are contained in the subset associated with region β, then we say α is included by β. An outer region is a region that is not included by any other region. Natural choices of outer regions are cliques in the original graph $\mathcal{G}$. An inner region corresponds to the intersection of some regions. Let $\mathcal{O}$ denote the set of all outer regions and $\mathcal{I}$ denote the set of all inner regions. We have $\mathcal{R} = \mathcal{O} \bigcup \mathcal{I}$. A particular selection of regions can be visualized by a Bethe region graph, where each node represents a region and the edges between nodes correspond to the inclusion relationship of the regions, i.e., if $\gamma \subset \gamma'$, then there is an edge between γ and γ'. Let us use $\mathcal{E}$ to denote the set of edges in a Bethe region graph and $B(\mathcal{O}, \mathcal{I}, \mathcal{E})$ to denote a Bethe region graph. In Bethe approximation, it does

require that inner regions are intersections of only outer regions, i.e., there is no inner regions that are intersections of inner regions. Under this condition, the Moebius number can be readily computed:

$$c_\gamma = \begin{cases} 1 & \text{if } \gamma \in \mathcal{O} \\ 1 - \deg(\gamma) & \text{if } \gamma \in \mathcal{I} \end{cases}$$

where $\deg(\cdot)$ denotes the degree of a node in the region graph. Figure 1 shows an example of Markov network and two possible region graphs associated with it.



(a) Markov network (b) Bethe region graph #1

(c) Bethe region graph #2

Figure 1: Markov network and region graphs. In (b) and (c), the ovals represent outer regions, and the boxes represent inner regions. The overcounting numbers are shown beside the regions.

It can be shown if the original graphical model admits a singly-connected Bethe region graph, the associated Bethe approximation is exact (Yedidia, Freeman, & Weiss 2000; Heskes 2002; Yedidia, Freeman, & Weiss 2005). With graphs of large tree-width, one minimizes the Bethe free energy function and uses its solution to obtain an estimation of the partition function and true marginal distributions. The Bethe free energy is a function of the so-called pseudo-marginals, whose definition is given below.

Definition For a Bethe region graph $B(\mathcal{O}, \mathcal{I}, \mathcal{E})$, the pseudo-marginal of a region $\gamma \in \mathcal{R}$ is

$$q_\gamma \in [0,1]^{|S_\gamma|}.$$

where S_γ is the sample space of the random vector X_γ. Let x_γ be a point in the sample space S_γ. It can be used to index q_γ such that $q_\gamma(x_\gamma)$ is a component of q_γ. The pseudo-marginals of the Bethe region graph is a vector of the form $q = (\ldots, q_\alpha, \ldots, q_\beta, \ldots)$, where α is an index running over $\mathcal{O}$ and β is an index running over $\mathcal{I}$. Our notations also allow us to use a subset of regions to index the pseudo-marginal vector. $q_\mathcal{S}$ is the collection of q's components that are associated with the regions in $\mathcal{S}$.

The Bethe free energy can be expressed as a function of the pseudo-marginals:

$$F_b(q) = \sum_{\alpha \in \mathcal{O}} \sum_{X_\alpha} q(X_\alpha) \log \frac{q(X_\alpha)}{\psi_\alpha(X_\alpha)} + \sum_{\beta \in \mathcal{I}} c_\beta \sum_{X_\beta} q(X_\beta) \log q(X_\beta)$$

Now we formally state the Bethe free energy minimization problem (Yedidia, Freeman, & Weiss 2000; 2005; Heskes 2006):

$$\begin{array}{ll} \text{minimize} & F_b(q) \\ \text{subject to} & q_\gamma(X_\gamma) \geq 0 \ \ \forall \gamma \in \mathcal{R} \, \forall X_\gamma \in S_\gamma \\ & \sum_{X_\gamma} q_\gamma(X_\gamma) = 1 \ \ \forall \gamma \in \mathcal{R} \\ & \sum_{X_{\alpha \setminus \beta}} q_\alpha(X_\alpha) = q_\beta(X_\beta) \\ & \quad \forall \alpha \in \mathcal{O}, \beta \in \mathcal{I}, \alpha \supset \beta, X_\beta \in S_\beta. \end{array} \tag{2}$$

The second and third set of constraints in (2) are known as normalization and consistency constraints, respectively. Note that with consistency constraints, we only need the normalization constraints for either outer regions or inner regions, but not both. Indeed, we can eliminate variables which are pseudo-marginals of all inner regions because they can be obtained by marginalizing the pseudo-marginals of their including outer regions. Consistency constraints need to be written in terms of pseudo-marginals of every pair of intersecting outer regions. This alternative formulation is given below:

$$\begin{array}{ll} \text{minimize} & F_b(q) \\ \text{subject to} & q_\alpha(X_\alpha) \geq 0 \ \ \forall \alpha \in \mathcal{O} \, \forall X_\alpha \in S_\alpha \\ & \sum_{X_\alpha} q_\alpha(X_\alpha) = 1 \ \ \forall \alpha \in \mathcal{O} \\ & \sum_{X_{\alpha \setminus \beta}} q_\alpha(X_\alpha) = \sum_{X_{\alpha' \setminus \beta}} q_{\alpha'}(X_{\alpha'}) \\ & \quad \forall \alpha, \alpha' \in \mathcal{O}, \alpha \cap \alpha' = \beta, X_\beta \in S_\beta. \end{array} \tag{3}$$

Note here F_b is a function of pseudo-marginals of only outer regions (the pseudo-marginals for inner regions are substituted by those of outer regions). Let us denote the feasible set of (3) by Q. It can be described by $Q = \{q | Aq = d, q \geq 0\}$, where A is the appropriate coefficient matrix and d is the appropriate right hand side vector. Let S_v be the sample space of random variable X_v with $v \in V$. Without loss of generality we assume that $|S_{v_1}| = |S_{v_2}| = s$, $\forall v_1, v_2 \in V$. Also assume that each inner region contains only one node. This is always possible because we can choose outer regions to be pairs of adjacent nodes, which is usually the approach taken in Ising models (Pelizzola 2005). These two assumptions will greatly simplify the presentation of next two propositions. The rank of matrix A is determined by next proposition.

Proposition 2.1. *Rank* $A = |\mathcal{O}| + (s-1) \times \sum_{\beta \in \mathcal{I}} |c_\beta|$.

Proof. Let A_n denote the submatrix associated with the normalization constraints. Since each normalization constraint contains a set of variables that are absent in other normalization constraints, the rank of A_n is just the number of normalization constraints, i.e., the number of outer regions $|\mathcal{O}|$.

For a particular $X_\beta \in S_\beta$, we have

$$\sum_{X_{\alpha_1 \setminus \beta}} q_{\alpha_1}(X_{\alpha_1}) = \cdots = \sum_{X_{\alpha_{deg(\beta)}} \setminus \beta} q_{\alpha_{deg(\beta)}}(X_{\alpha_{deg(\beta)}}) = q_\beta(X_\beta), \tag{4}$$

where $\alpha_1, \ldots, \alpha_{deg(\beta)}$ are the outer regions adjacent to β. Considering an equality between two outer regions as an edge connecting them, the set of equalities representing (4) corresponds to a set of edges connecting these outer regions.

So the minimum number of such edges is the number of outer regions minus 1, i.e., $\deg(\beta) - 1 = |c_\beta|$. Consider all $X_\beta \in S_\beta$, the number of constraints is $s \times |c_\beta|$. Note we can use normalization constraints to remove the equalities for one particular X_β. So overall we have $(s-1) \times |c_\beta|$ consistency constraints for β. By induction, we can show that consistency constraints for different inner regions are independent. Thus the total number of independent equality constraints is $|\mathcal{O}| + (s-1) \times \sum_{\beta \in \mathcal{I}} |c_\beta|$. □

Consider fixing the pseudo-marginals associated with a subset of inner regions, denoted by $\mathcal{I}'$, to decrease the dimension of the feasible set. This is equivalent to adding equality constraints to (3). The resulting feasible set can be described by

$$\{q | \begin{bmatrix} A \\ C \end{bmatrix} q = \begin{bmatrix} d \\ f \end{bmatrix}, q \geq 0\},$$

where C is the appropriate coefficient matrix and f is the appropriate right hand side vector, for the added equality constraints. The rank of the new constraint matrix is determined by the next proposition.

Proposition 2.2.

$$Rank \begin{bmatrix} A \\ C \end{bmatrix} = rank\ A + (s-1) \times |\mathcal{I}'|.$$

Proof. For a particular $X_\beta \in S_\beta$, we add one equality constraint: $\sum_{X_{\alpha \setminus \beta}} q_\alpha(X_\alpha) = q_\beta(X_\beta)$, where α is some adjacent outer region of β and $q_\beta(X_\beta)$ is a fixed pseudo-marginal. Again with normalization constraints, the number of added equalities for β is $s-1$. These equalities are linearly independent. The added constraints between different inner region are also linearly independent. Finally added constraints are independent of the original constraints. So the total number of independent constraints is $rank\ A + (s-1) \times |\mathcal{I}'|$. □

3 Convex Relaxation Method

As a well-known result (Pakzad & Anantharam 2002; Heskes, Albers, & Kappen 2003; Pakzad & Anantharam 2005), the following theorem is a key element in the development of the UPS algorithm. It also plays an important role in the convergence analysis of the convex relaxation method.

Theorem 3.1. *If each connected component of a Bethe region graph $B(\mathcal{O}, \mathcal{I}, \mathcal{E})$ is singly-connected, then the Bethe free energy function defined on this region graph is strictly convex over the constraint set.*

Let $\mathcal{I}'$ be a subset of inner regions. We can obtain a subproblem of (2) by fixing the pseudo-marginals associated with $\mathcal{I}'$ to a constant vector p. Denote this subproblem by $\mathcal{P}(\mathcal{I}', p)$. If the induced subgraph of B by $\mathcal{O} \cup \mathcal{I} \setminus \mathcal{I}'$ is singly-connected, then by Theorem 3.1, $\mathcal{P}(\mathcal{I}', p)$ is convex. In such a case, we call $\mathcal{I}'$ a *condition set*.

A formal description of the convex relaxation method is given in Procedure 1.

In Procedure 1, the objective function of each subproblem is strictly convex over a reduced feasible set (as a result of fixing some variables). Thus the unique global minimum can be easily attained by an exact algorithm, such as the one used in (Teh & Welling 2001b). The subproblems interact with each other through shared pseudo-marginal variables. Procedure 1 is considered as converged when the value of all pseudo-marginal variables remain unchanged after any consecutive execution of all subproblems, i.e., one round of the **while** loop.

Procedure 1 The convex relaxation method

```
input: A sequence of condition sets {I_1, ..., I_m}.
input: q – initial value for pseudo-marginals.
output: q – final value for pseudo-marginals.
while ¬ converge do
  for i = 1 to m do
    p ← q_{I_i}.
    Solve the subproblem P(I_i, p).
    q ← solution of P(I_i, p).
  end for
end while
```

Formulation (3) gives us insight to some linear-algebraic properties of the feasible sets. For next two propositions we will focus on formulation (3). At each iteration, we reduce the original feasible set to a lower dimensional convex polyhedral set by adding a set of equality constraints. Let $\mathcal{S}$ be the smallest affine subspace in which the original feasible set lies. Also let $\mathcal{S}_i$ be the smallest affine subspace in which the feasible set of the i-th subproblem lies. Let C_i denote the coefficient matrix for the equality constraints added in the i-th subproblem. The following proposition states a necessary and sufficient condition for $\{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ spanning $\mathcal{S}$.

Proposition 3.2. *The affine subspace $\mathcal{S}$ is spanned by $\{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ if and only if $\cap_{i=1}^m \mathcal{I}_i = \emptyset$.*

Proof. By the Inclusion-Exclusion principle, the necessary and sufficient condition for $\{\mathcal{S}_1, \ldots, \mathcal{S}_m\}$ spanning $\mathcal{S}$ is:

$$\begin{aligned} \text{rank } A = &\sum_{i=1}^{m} \text{rank} \begin{bmatrix} A \\ C_i \end{bmatrix} - \sum_{i,j:1 \leq i < j \leq m} \text{rank} \begin{bmatrix} A \\ C_i \\ C_j \end{bmatrix} \\ &+ \sum_{i,j,k:1 \leq i < j < k \leq m} \text{rank} \begin{bmatrix} A \\ C_i \\ C_j \\ C_k \end{bmatrix} \\ &- \cdots + (-1)^{m-1} \text{rank} \begin{bmatrix} A \\ C_1 \\ \vdots \\ C_m \end{bmatrix}. \end{aligned} \tag{5}$$

Invoking Proposition 2.2, we can show (5) is equivalent to:

$$\begin{aligned} &\sum_{i=1}^{m} |\mathcal{I}_i| - \sum_{\substack{i,j: \\ 1 \leq i < j \leq m}} |\mathcal{I}_i \cup \mathcal{I}_j| + \sum_{\substack{i,j,k: \\ 1 \leq i < j < k \leq m}} |\mathcal{I}_i \cup \mathcal{I}_j \cup I_k| \\ &- \cdots + (-1)^{m-1} |\mathcal{I}_1 \cup \cdots \cup \mathcal{I}_m| = 0, \end{aligned} \tag{6}$$

where we use the binomial identity

$$\sum_{k=1}^{m}(-1)^{k-1}\binom{m}{k}=1$$

to eliminate rank A from both sides of (5).

By a corollary of the Inclusion-Exclusion principle, (6) is true if and only if $\cap_{i=1}^{m}\mathcal{I}_i=\emptyset$. □

Now we make an assumption about the positivity of the clique potentials. This assumption is essential to the proof of our convergence results. This assumption satisfies the condition of Proposition 5 in (Yedidia, Freeman, & Weiss 2005), which will be used to prove next proposition.

Assumption 3.3. *All clique potentials are strictly positive, i.e.,* $\psi_\alpha(x_\alpha)>0\ \ \forall\alpha\in\mathcal{C}, x_\alpha\in S_\alpha$.

Let $\mathcal{I}_i$ denote the subset of inner regions that are conditioned on at the i-th iteration of the inner **for** loop of Procedure 1. Next proposition gives a sufficient condition for the equivalence of fixed points of Procedure 1 and stationary points of the free energy minimization problem.

Proposition 3.4. *Under Assumption 3.3, any fixed point of Procedure 1 is a stationary point of the constrained minimization problem and vice versa if* $\cap_{i=1}^{m}\mathcal{I}_i=\emptyset$.

Proof. Suppose the algorithm converges to a fixed point $\bar{q}$, then we have for $i=1,\ldots,m$, $\bar{q}_i$ is the unique global minimum of the subproblem at the $i-th$ inner iteration. Adapting the proof for Proposition 5 in (Yedidia, Freeman, & Weiss 2005), we can show that under Assumption 3.3 the unique global minimum of each subproblem of Procedure 1 lies in the relative interior of that subproblem's constraint set. Indeed, if we initialize the pseudo-marginals and messages to be positive, then they will remain positive during the execution of algorithm. This means for $i=1,\ldots,m$, $\nabla F_b(\bar{q})$ is orthogonal to the smallest affine subspace in which the i-th subproblem's feasible set lies. If these affine subspaces span the smallest affine subspace where the original feasible set lies in, then $\nabla F_b(\bar{q})$ is orthogonal to the original feasible set, ensuring that $\bar{q}$ is a stationary point. Then the claim follows from the result in Proposition 3.2. □

The feasible set of (3) is a nonempty, compact polyhedral set. In addition, Assumption 3.3 ensures that the energy function stays finite because logarithm functions will not be evaluated at 0. Thus there exists at least two stationary points because at least the global minimum and maximum are attained. According to Bolzano-Weierstrass theorem, the sequence of iterates generated by any algorithm will have at least one cluster point. If it can be proved that every cluster point is stationary, then the algorithm is guaranteed to converge.

4 Convergence Analysis

In this section, we prove the convergence result for the convex relaxation method. The result presented here is similar to those in (Bertsekas 1999; L. Grippo 2000) but the overlapping nature of the coordinate blocks prevents an simple extension from these results. The result in (Bertsekas 1999; L. Grippo 2000) states that for a problem with C^1 objective function and feasible set being a Cartesian product of m orthogonal nonempty, closed convex sets, any cluster point of the sequence generated by the block GS method is a stationary point, under the assumption that the sequence admits cluster points and the objective function is componentwise strictly quasiconvex with respect to all components.

Let us use $k\in\mathbb{Z}^+$ to index the iterations of the outer **while** loop of Procedure 1. Also we use $i\in\{0,\ldots,m\}$ to index the iterations of the inner **for** loop. Let $w(k,i)$ be the vector of pseudo-marginals obtained at the end of i-th iteration of the inner loop in the k-th iteration of the outer loop. Let $Q_i\equiv Q_i(w(k,i-1))$ be the feasible set of the i-th subproblem given the current iterate $w(k,i-1)$. $w(k,i)$ is defined recursively as follows:

$$\begin{aligned}&w(k,0)=q^k,\\&w(k,i)=(w_{\mathcal{I}_i}(k,i-1),\arg\min_{p_{\mathcal{R}_i}\in Q_i}F_b(p_{\mathcal{R}_i}))\\&\quad\forall i=1,\ldots,m,\\&w(k+1,0)=w(k,m)=q^{k+1},\end{aligned}$$

where we use the subset of regions to index the pseudo-marginal vector p and w. In our notation, we assume that the operator $(\cdot)$ rearrange the components of its argument according to the original order in q.

We use the following proposition to show the convergence of the objective function along a converging subsequence of $\{w(k,i)\}$.

Proposition 4.1. *If some sequence* $\{w(k,i)\}$, $i\in\{0,\ldots,m\}$, *admits a cluster point* $\bar{w}$, *then for every* $j\in\{0,\ldots,m\}$, *we have*

$$\lim_{k\to\infty}F_b(w(k,j))=F_b(\bar{w}).$$

Proof. Let $\{w(k,i)\}_{k\in\mathcal{K}}$ be the subsequence of $\{w(k,i)\}$ converging to $\bar{w}$. Since F_b is continuous, $\{F_b(w(k,i))\}_{k\in\mathcal{K}}$ converges to $F_b(\bar{w})$. Also because F_b is nonincreasing, so the whole sequence $\{F_b(w(k,i))\}$ converges to $F_b(\bar{w})$. We also have

$$F_b(w(k+1,i))\le F_b(w(k,j))\le F_b(w(k,i))\quad\forall j\in\{i+1,\ldots,m\}\qquad(7)$$

and

$$F_b(w(k,i))\le F_b(w(k,j))\le F_b(w(k-1,i))\quad\forall j\in\{0,\ldots,i-1\}.\qquad(8)$$

Taking limit on the inequalities (7) and (8) yields $\lim_{k\to\infty}F_b(w(k,j))=F_b(\bar{w}),\forall j\in\{0,\ldots,m\}$. □

Now we show the last proposition before our final convergence result.

Proposition 4.2. *Let* $\{q^k\}$ *be a sequence of points in* Q *converging to* $\bar{q}$. *For a particular* $i\in\{1,\cdots,m\}$, *let* $\{p^k\}$ *be a sequence of vectors whose components are defined as follows:*

$$p_j^k=\begin{cases}q_j^k & \text{if } j\neq i,\\ \arg\min_{u\in Q_i}F_b(u,q^k_{\mathcal{I}_i}) & \text{if } j=i.\end{cases}$$

Then, if $\lim_{k\to\infty}F_b(p^k)-F_b(q^k)=0$, *we have* $\lim_{k\to\infty}\|p^k-q^k\|=0$.

Now we present the convergence result.

Proposition 4.3. *Suppose Assumption 3.3 holds and the convex relaxation method (Procedure 1) satisfies* $\cap_{i=1}^{m} \mathcal{I}_i = \emptyset$*, the sequence of iterates generated by the method converges in the sense that*

1. *It has cluster points.*
2. *Any cluster point is a stationary point of the Bethe free energy minimization problem.*

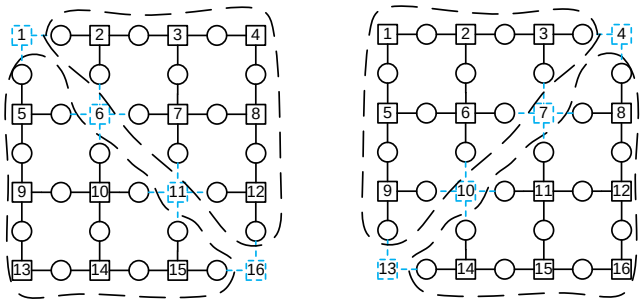
Proof. Let $\bar{q}$ be any cluster point of $\{q^k\}$ and $\{q^k\}_{k\in K}$ be the subsequence converging to $\bar{q}$. Note that $q^k = w(k,0) = w(k-1,m)$. By Proposition 4.1, we have $\lim_{k\to\infty} F_b(w(k,1)) = F_b(\bar{q})$. Using Proposition 4.2 for $i = 1$, identifying $\{q^k\}$ with $\{w(k,1)\}$, we can show that $\{w(k,1)\}_K$ converges to $\bar{q}$. Sequentially applying Proposition 4.2 for $i = 2, \cdots, m-1$, we have $\{w(k,i)\}_K$ converges to $\bar{q}$ for $i = 2, \cdots, m$. Thus $\bar{q}$ is a fixed point and in addition a stationary point (by Proposition 3.4). □

5 Distributed Optimization

From an algorithmic point of view, a natural extension of Procedure 1 to a distributed optimization can be obtained by further constraining the condition set to be graph cuts. That means, at each iteration of the convex relaxation method, the removal of the condition set separates the region graph into disconnected components. This way we obtain a decomposition of the original problem into smaller independent ones, each of which defined on one component. These smaller problems can be solved in parallel on a multiprocessor system. We assume at each iteration the region graph is partitioned into t components thus the original problem can be solved in parallel on t processors. By modeling the region graphs as hypergraphs, we are able to apply a hypergraph partition algorithm to obtain balanced decompositions.

At one particular iteration, each region (not in the condition set), is assigned to only one processor. However, at the following iteration, it can be assigned to a different processor. A region in the condition set may be assigned to more than one processors. When a region turns to be in the condition set at some iteration, the processor that owns this region in previous iteration, needs to send the data for this region to all processors that need this region in the current iteration. Thus synchronization among the processors is needed after every iteration, assuming the parallel system uses distributed memories.

We demonstrate the synchronization costs through the example in Figure 2. At the first iteration, we have two disconnected graphs as shown in Figure 2(a). Processor P_1 holds the graph at the lower left corner and processor P_2 holds the one at the upper right corner. P_1 contains data for inner regions $\{5, 9, 10, 13, 14, 15\}$ while P_2 contains data for $\{2, 3, 4, 7, 8, 12\}$. Both processors have the separator $\{1, 6, 11, 16\}$ since the two subproblems depend on them. At next iteration, we have two different graphs as shown in Figure 2(b). P_1 holds the graph at the upper left corner and P_2 holds the one at the lower right corner. P_1 has inner regions $\{1, 2, 3, 5, 6, 9\}$ while P_2 has $\{8, 11, 12, 14, 15, 16\}$. Both processors need to contain the separator $\{4, 7, 10, 13\}$.



(a) separator: 1,6,11,16 (b) separator: 4,7,10,13

Figure 2: 4×4 grid graph. Boxes: outer regions. Cycles: inner regions. The separators are drawn in dashed lines. The components are surrounded by dashed lines.

At every iteration a processor needs data that are updated in other processors in previous iterations. For example, at the second iteration, P_1 needs data for inner regions 2,3 and their adjacent outer regions. These data needs to be sent from P_2 to P_1 because they are updated in P_2 at the first iteration.

If at the end of each iteration, each processor sends all data that are needed by other processors and receives from other processors the data it needs, then we call this method the *fully synchronized method*. The synchronization can incur significant communication overhead in the worst case. If for each processor, the pseudo-marginals of the regions that are in the separator for this iteration are updated, i.e., they have been synchronized with some processors that last modified the pseudo-marginals of these regions in some iteration prior to the current one, we call this method the *partially synchronized parallel method*. Compared to the fully parallel method, the synchronization cost is minimal for the partially parallel method. Through next proposition we show that the partially parallel method has the same solution as the fully synchronized parallel method.

Proposition 5.1. *Given a Bethe region graph* $B(\mathcal{O}, \mathcal{I}, \mathcal{E})$ *and a sequence of condition set* $\{\mathcal{I}_1, \ldots, \mathcal{I}_m\}$ *such that* $\mathcal{I}_i$*,* $i \in \{1, \ldots, m\}$*, separates* B *into* t *components. Then the iterates* $\{q^k\}$ *generated by the fully and partially synchronized parallel method are same provided both methods have same initial solution.*

Proof. Let $\mathcal{P}_{ij}$ denote the regions assigned to the i-th processor at k-th iteration such that $j = ((k-1) \bmod m) + 1$. Let $\mathcal{P}_i = \cup_{j=1}^{m} \mathcal{P}_{ij}$ $i = 1, \ldots, t$. Let $\mathcal{S}_{jl} = \cup_{i=1}^{t} (\mathcal{P}_{il} \setminus \mathcal{P}_{ij})$, where $l = ((j+1) \bmod m) + 1$. We have $\mathcal{R} = \mathcal{O} \cup \mathcal{I} = \cup_{i=1}^{t} P_i$.

We prove that the iterates generated by two methods are always same, by induction on the iteration number. By assumption, at first iteration, the pseudo-marginals q of both methods are same. Now assume q_k are same at the end of iteration k, where $j = ((k-1) \bmod m) + 1$, then we have for both methods,

$$F_b^k = F_b(q_{\mathcal{S}_{jl}}^k, q_{\mathcal{R}\setminus\mathcal{S}_{jl}}^k),$$

where we use the subsets of regions to index the whole pseudo-marginal vector q. Let $\Phi^{k+1}(q_{\mathcal{S}_{jl}}, q_{\mathcal{R}_l \setminus \mathcal{S}_{jl}}) = F_b(q_{\mathcal{S}_{jl}}, q_{\mathcal{R}_l \setminus \mathcal{S}_{jl}}, q^k_{\mathcal{R} \setminus \mathcal{R}_l})$. At the end of iteration $k+1$, for both methods we have

$$q^{k+1}_{\mathcal{R}_l} = \arg \min_{q_{\mathcal{R}_i} : (q_{\mathcal{R}_i}, q_{\mathcal{R} \setminus \mathcal{R}_i}) \in Q} \Phi^{k+1}(q_{\mathcal{S}_{jl}}, q_{\mathcal{R}_l \setminus \mathcal{S}_{jl}}), \quad (9)$$

$$q^{k+1} = (q^{k+1}_{\mathcal{R}_l}, q^k_{\mathcal{R} \setminus \mathcal{R}_l}), \quad (10)$$

and

$$F^{k+1}_b = \Phi^{k+1}(q^{k+1}_{\mathcal{S}_{jl}}, q^{k+1}_{\mathcal{R}_l \setminus \mathcal{S}_{jl}}) = F_b(q^{k+1}),$$

where Q denotes the constraint set. For iteration $k+1$, we have $q^k_{\mathcal{R} \setminus \mathcal{R}_l}$ because $\mathcal{R} \setminus \mathcal{R}_l$ are the separators in the current iteration and by assumption they are synchronized in both parallel methods.

We know that $F^{k+1}_b \leq F^k_b$ because $(q^k_{\mathcal{S}_{jl}}, q^k_{\mathcal{R} \setminus \mathcal{S}_{jl}})$ is a feasible solution of Φ^{k+1} and Φ^{k+1} is convex. The initial solutions of the optimization problem in (9) for two methods are not necessarily the same. In addition, the initial solution for the partially parallel method is likely to be infeasible, i.e., it does not satisfy the consistency constraints. However it can be shown that the algorithm for solving the subproblems works with infeasible initial solutions. $B[\mathcal{R}_l]$ satisfies Theorem 3.1, thus Φ^{k+1} is strictly convex and the solution to (9) in both methods is the unique global solution of Φ^{k+1}. This means that q^{k+1} is the same for two methods. This completes our proof. □

Figure 3 illustrates the proof idea of Proposition 5.1 in 2 dimension. Two different initial solution q^k, lead to the same unique global minimum q^{k+1} of the strictly convex function Φ^{k+1}.

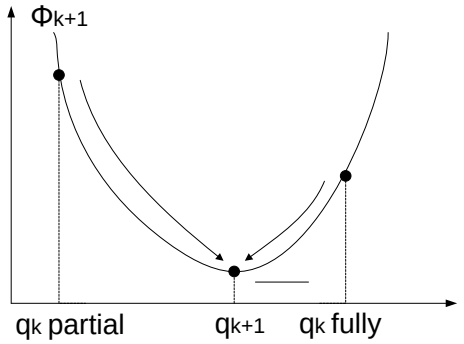


Figure 3: Two-dimensional illustration of the proof idea of Proposition 5.1.

Indeed, the algorithm converges with any permutation of the sequence $\{\mathcal{I}_1, \ldots, \mathcal{I}_m\}$. The communication cost between i-th and $i+1$-th iteration depends on $\mathcal{I}_i$ and $\mathcal{I}_{i+1}$. We find the optimal ordering of condition sets by solving a min-cost flow problem.

6 Discussion

We proved the sufficient condition for the convergence of the convex relaxation method. Whether there exists stronger sufficient condition remains unknown though. Our work provides insights to the analyses of those methods, which directly optimize Bethe free energy or similar function. The structure of the optimization problem, which is represented as the linear-algebraic properties of the constraint set, directly relate to the topology of the region graph. This suggests that desirable structures may exist in methods that build subproblems on a collection of subgraphs. These structures could be exploited to improve the efficiency of the algorithm.

The parallel implementation is at the algorithmic level, which indicates that it can be used in conjunction with lower level parallelization techniques. Preliminary results have shown that general region graphs can be partitioned with well-balanced components. The computation time for balanced components is approximately the same while the communication cost is always dominated by the computation.

Acknowledgements

This work is supported by NIH grant HG004175. The author thanks Elizabeth Thompson and Paul Tseng for helpful discussion. The author also thanks anonymous reviewers for their helpful comments.

References

Bertsekas, D. P. 1999. *Nonlinear Programming*. Athena Scientific.

Botetz, B. 2007. Efficient belief propagation for vision using linear constraint nodes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Carbonetto, P.; de Freitas, N.; and Barnard, K. 2004. A statistical model for general contextual object recognition. In *In ECCV*, 350–362.

Drton, M., and Eichler, M. 2006. Maximum likelihood estimation in gaussian chain graph models under the alternative markov property. *Scandinavian Journal of Statistics* 33(2):247–257.

Globerson, A., and Jaakkola, T. 2007. Convergent propagation algorithms via oriented trees. In *In Uncertainty in Artificial Intelligence*.

Hazan, T., and Shashua, A. 2008. Convergent message-passing algorithms for inference over general graphs with convex free energies. *The 24th Conference on Uncertainty in Artificial Intelligence (UAI)*.

Heskes, T.; Albers, K.; and Kappen, B. 2003. Approximate inference and constrained optimization. In *Uncertainty in Artificial Intelligence*, 313–320. Morgan Kaufmann Publishers.

Heskes, T. 2002. Stable fixed points of loopy belief propagation are local minima of the bethe free energy. In *NIPS*, 343–350.

Heskes, T. 2006. Convexity arguments for efficient minimization of the bethe and kikuchi free energies. *Journal of Artificial Intelligence Research* 26:153–190.

Kschischang, F. R.; Frey, B. J.; and Loeliger, H. A. 2001. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on* 47(2):498–519.

L. Grippo, M. S. 2000. On the convergence of the block nonlinear gauss-seidel method under convex constraints. *Operations Research Letter* 26(3):127–136.

Luo, Z. Q., and Tseng, P. 1992. On the convergence of the coordinate descent method for convex differentiable minimization. *Journal Optimization Theory and Applications* 72(1):7–35.

Meltzer, T.; Globerson, A.; and Weiss, Y. 2009. Convergent message passing algorithms - a unifying view. In *In Uncertainty in Artificial Intelligence*.

Mooij, J., and Kappen, H. 2005. Sufficient conditions for convergence of loopy belief propagation. In *Proceedings of the 25th Conference Annual Conference on Uncertainty in Artificial Intelligence*, 396–40.

Nocedal, J., and Wright, S. 1999. *Numerical Optimization*. Springer.

Pakzad, P., and Anantharam, V. 2002. Belief propagation and statistical physics. In *Princeton University*.

Pakzad, P., and Anantharam, V. 2005. Estimation and marginalization using kikuchi approximation methods. *Neural Computation* 17:1836–1873.

Pelizzola, A. 2005. Cluster variation method in statistical physics and probabilistic graphical models. *Physics A: Mathematical and General* 38:309–339.

Powell, M. 1973. On search directions for minimization algorithms. *Mathematical Programming* 4(1):193–201.

Teh, Y. W., and Welling, M. 2001a. Passing and bouncing messages for generalized inference. Technical Report GCNU TR 2001-01, Gatsby Computational Neuroscience Unit, University College London.

Teh, Y. W., and Welling, M. 2001b. The unified propagation and scaling algorithm. In *NIPS*, 953–960.

Wainwright, M. J.; Jaakkola, T. S.; and Willsky, A. S. 2002. A new class of upper bounds on the log partition function. In *In Uncertainty in Artificial Intelligence*, 536–543.

Welling, M., and Teh, Y. W. 2001. Belief optimization for binary networks: a stable alternative to loopy belief propagation. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, volume 17.

Winn, J., and Bishop, C. M. 2005. Variational message passing. *Joural of Machine Learning Reseach* 6:661–694.

Xie, Z.; Gao, J.; and Wu, X. 2009. Regional category parsing in undirected graphical models. *Pattern Recognition Letters* 30(14):1264–1272.

Yedidia, J. S.; Freeman, W. T.; and Weiss, Y. 2000. Generalized belief propagation. In *NIPS*, 689–695. MIT Press.

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

Yuille, A. L., and Rangarajan, A. 2003. The concave-convex procedure. *Neural Computation* 15(4):915–936.