Message-Passing Algorithms for MAP Estimation Using DC Programming

Akshat Kumar
Univ. of Massachusetts Amherst

Shlomo Zilberstein
Univ. of Massachusetts Amherst

Marc Toussaint
FU Berlin

Abstract

We address the problem of finding the most likely assignment or MAP estimation in a Markov random field. We analyze the linear programming formulation of MAP through the lens of difference of convex functions (DC) programming, and use the concave-convex procedure (CCCP) to develop efficient message-passing solvers. The resulting algorithms are guaranteed to converge to a global optimum of the well-studied local polytope, an outer bound on the MAP marginal polytope. To tighten the outer bound, we show how to combine it with the mean-field based inner bound and, again, solve it using CCCP. We also identify a useful relationship between the DC formulations and some recently proposed algorithms based on Bregman divergence. Experimentally, this hybrid approach produces optimal solutions for a range of hard OR problems and near-optimal solutions for standard benchmarks.

1 Introduction

Probabilistic graphical models provide an effective framework for compactly representing probability distributions over high-dimensional spaces and performing complex inference using simple local update procedures. In this work, we focus on the class of undirected models called *Markov random fields* (MRFs) [3, 22]. A common inference problem in this model is to compute the most probable assignment to variables, also called the *maximum a posteriori* (MAP) assignment. MAP estimation is crucial for many practical applications in computer vision and bioinformatics such as protein design [25, 19] among others. Computing MAP exactly is NP-hard for general graphs. Thus, approximate inference techniques are often used [22, 16].

Appearing in Proceedings of the 15th International Conference on Artificial Intelligence and Statistics (AISTATS) 2012, La Palma, Canary Islands. Volume 22 of JMLR: W&CP 22. Copyright 2012 by the authors.

Several popular methods for MAP estimation such as tree-reweighted max-product (TRMP) [21, 8] and max-product LP (MPLP) [6, 19] are based on a linear programming (LP) relaxation of the MAP problem. The LP approaches first reformulate MAP as a linear objective subject to constraints that enforce the variables to be marginals of a joint probability distribution—the respective constraint set is called the *marginal polytope.* The LP relaxations relax this constrain set to enforce only local consistency—the respective constraint set, the *local polytope*, is an outer bound on the marginal polytope [22]. Complementary to this approach, Ravikumar and Lafferty [12] present an inner bound on the marginal polytope based on enforcing the mean-field structure over the joint distribution of all the variables. This inner bound is quite compact with much fewer variables and simple constraints. And it is accurate in that optimizing over the inner bound will yield the MAP assignment. However, the inner bound is not convex, which makes global optimization challenging. We present our contributions in light of these two formulations of the MAP problem.

We start by analyzing the linear programming relaxation of MAP through the lens of *difference of convex functions* (DC) programming. We then use the concave-convex procedure (CCCP) [26] to solve it efficiently using message passing. The resulting algorithm is guaranteed to converge to the global optimum of the LP relaxation. In contrast, existing algorithms that work on the dual of the LP relaxation such as TRMP [21], MPLP [6], and MSD [23] may get stuck in local optima. A number of globally convergent alternatives also exist such as the dual decomposition based approaches [9, 7]. Some of these approaches smooth the discrete objective using a smoothing parameter, but achieving high accuracy may lead to numerical instability [7]. Furthermore, it is unclear how some of the approaches to tighten the LP relaxation that work in the primal can be applied to the dual formulation [17]. In contrast, our approach works in the primal formulation, does not require any smoothing parameter and can easily accommodate additional constraints to tighten the LP relaxation. Additionally, black-box LP solvers are often not as scalable as graph-based message-passing techniques, as shown in [25]. Therefore developing globally convergent message-passing algorithms for the LP relaxation is important.

On many problems, the outer bound based LP relaxation may not provide a tight approximation, as noted in [19, 18, 24, 1]. A popular approach to tighten the outer bound is to enforce the local consistency over higher order cliques of variables or clusters [19, 24, 1]. However, the complexity of this approach increases exponentially with the cluster size, limiting its scalability. We take an alternative approach. As noted earlier, the inner bound over the marginal polytope is *exact*. We therefore combine the inner bound with the outer bound resulting in a tighter approximation to the marginal polytope. The resulting optimization problem may not be convex. However, our approach is parameterized by a set Q, which precisely controls how non-convexity is introduced. When $|Q| = 0$, it corresponds to the LP relaxation and can be solved optimally. Using such flexible control over non-convexity, we produce optimal solutions for a range of operations research (OR) problems for which other approaches such as MPLP fail to do so. We also discuss the close relationship between the DC programming formulations and recently proposed globally convergent algorithms for solving the LP relaxation using Bregman divergence [11].

2 The MAP Problem

A pairwise Markov random field (MRF) is described using an undirected graph $G = (V, E)$. A discrete random variable x_i with a finite domain is associated with each node $i \in V$ of the graph. We assume that there are n variables with maximal domain size k. Associated with each edge $(i, j) \in E$ is a potential function $\theta_{ij}(x_i, x_j)$. The complete assignment $\boldsymbol{x}$ has the probability: $p(\boldsymbol{x}; \theta) \propto \exp\big(\sum_{ij\in E} \theta_{ij}(x_i, x_j)\big)$.

The MAP problem consists of finding the most probable assignment to all the variables under $p(\boldsymbol{x}; \theta)$. This is equivalent to finding the assignment $\boldsymbol{x}$ that maximizes the function $f(\boldsymbol{x}; \theta) = \sum_{ij\in E} \theta_{ij}(x_i, x_j)$. We assume wlog that each θ_{ij} is nonnegative, otherwise a constant can be added to each θ_{ij} without changing the optimal solution. We now describe the marginal polytope associated with a MRF.

Let $\boldsymbol{\mu}$ denote a vector of marginal probabilities for each node and edge of the MRF. That is, it includes $\mu_i(x_i)\ \forall i \in V$ and $\mu_{ij}(x_i, x_j)\ \forall (i, j) \in E$. The set of $\boldsymbol{\mu}$ that arises from some joint distribution p is referred to as the *marginal polytope*, $\mathcal{M}(G) =$

$$\{\boldsymbol{\mu} \mid \exists p(\boldsymbol{x}) : p(x_i, x_j) = \mu_{ij}(x_i, x_j),\ p(x_i) = \mu_i(x_i)\} \quad (1)$$

The MAP problem now becomes equivalent to the LP:

$$\max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} = \max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \sum_{ij\in E}\sum_{x_i x_j} \theta_{ij}(x_i, x_j)\mu_{ij}(x_i, x_j) \quad (2)$$

The constraint that the edge and node marginals must arise from a valid joint distribution is generally prohibitive to represent. Therefore it is relaxed, yielding an *outer bound* on the marginal polytope, also called the *local polytope* $\mathcal{L}(G) \supseteq \mathcal{M}(G)$ as it enforces only the local consistency on the marginals:

$$\begin{aligned}&\sum_{x_i} \mu_i(x_i) = 1\ \forall i \in V;\\ &\sum_{\hat{x}_j} \mu_{ij}(x_i, \hat{x}_j) = \mu_i(x_i)\ \forall i \in V, \forall x_i, \forall j \in Nb(i)\end{aligned} \quad (3)$$

The polytope $\mathcal{L}(G)$ comprises of all the $\boldsymbol{\mu}$ which satisfy the above constraints. The inner bound $\mathcal{I}(G) \subseteq \mathcal{M}(G)$ is constructed by assuming the mean-field structure on the joint distribution [12, 22]. That is, $p(\boldsymbol{x}) = \prod_{i=1}^n p_i(x_i)$. It is described as follows:

$$\begin{aligned}&\mu_{ij}(x_i, x_j) = \mu_i(x_i)\mu_j(x_j)\ \forall (i, j) \in E\\ &\sum_{x_i} \mu_i(x_i) = 1\ \forall i \in V\end{aligned} \quad (4)$$

The set $\mathcal{I}(G)$ includes all the $\boldsymbol{\mu}$ that satisfy the above constraints. The first constraint is non-linear in $\boldsymbol{\mu}$ and, in general, $\mathcal{I}(G)$ is non-convex. The relationship among these different formulations and the MAP is given by:

Proposition 1. *The MAP solution quality $f^\star$ satisfies*

$$f^\star = \max_{\boldsymbol{\mu}\in\mathcal{M}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} = \max_{\boldsymbol{\mu}\in\mathcal{I}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} \le \max_{\boldsymbol{\mu}\in\mathcal{L}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} \quad (5)$$

This known result [21, 12, 22] shows that optimizing over $\mathcal{I}(G)$ is exact. Furthermore, $\mathcal{I}(G)$ can be compactly represented using only $O(n)$ normalization constraints for node marginals and $O(nk)$ variables for each $\mu_i(x_i)$.

The Hybrid Set We now present our approach, which combines the inner bound with the outer bound resulting in the hybrid set of parameters $\mathcal{H}(G; Q)$. It is parameterized by a set of MRF edges Q. Intuitively, the non-convexity in the set $\mathcal{I}(G)$ arises from the mean-field constraint that $\mu_{ij}(x_i, x_j) = \mu_i(x_i)\mu_j(x_j)$. The set Q contains all the edges $e \in E$ for which the mean-field constraint is enforced. Let $L = E \setminus Q$ denote the rest of the edges, which intuitively correspond to the outer bound constraints. We denote the edges in

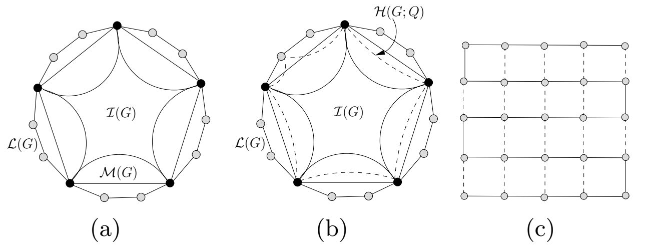


Figure 1: a) Relationship among marginal polytope, inner and outer bound; b) Hybrid inner-outer bound; c) A grid graph with dotted links denoting QP edges.

the set Q as QP edges (short for quadratic) and the edges in L as LP edges. The set $\mathcal{H}(G;Q)$ contains all the $\boldsymbol{\mu}$ that satisfy the following constraints:

$$\begin{aligned}
&\mu_{ij}(x_i,x_j) = \mu_i(x_i)\mu_j(x_j)\ \forall(i,j) \in Q\ ;\\
&\sum_{x_i} \mu_i(x_i) = 1\ \forall i \in V;\\
&\sum_{\hat{x}_j} \mu_{ij}(x_i,\hat{x}_j) = \mu_i(x_i)\ \forall i, \forall x_i, \forall j \in Nb_l(i) \qquad (6)
\end{aligned}$$

where $Nb_l(i)$ denotes the *LP neighbors* of a node i: $Nb_l(i) = \{j : j \in Nb(i) \wedge (i,j) \in L\}$.

Proposition 2. *It holds that:* $\max_{\boldsymbol{\mu}\in\mathcal{I}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} \leq \max_{\boldsymbol{\mu}\in\mathcal{H}(G;Q)} \boldsymbol{\mu}\cdot\boldsymbol{\theta} \leq \max_{\boldsymbol{\mu}\in\mathcal{L}(G)} \boldsymbol{\mu}\cdot\boldsymbol{\theta}$

Proof. First notice that the mean-field constraint $\mu_{ij}(x_i,x_j) = \mu_i(x_i)\mu_j(x_j)$ for an edge (i,j) automatically enforces the edge consistency constraint $\sum_{\hat{x}_i}\mu_{ij}(\hat{x}_i,x_j) = \mu_j(x_j)$ in both directions. Therefore, intuitively, the set $\mathcal{I}(G)$ is much smaller than the local polytope, which only enforces the edge consistency constraints. The first inequality holds because the mean-field constraint is not enforced for the LP edges in $\mathcal{H}(G;Q)$. Therefore the set $\mathcal{I}(G) \subseteq \mathcal{H}(G;Q)$. The second inequality holds because the mean-field constraint is enforced in addition to edge consistency for the QP edges in Q. Therefore $\mathcal{H}(G;Q) \subseteq \mathcal{L}(G)$. □

The above proposition highlights that the set $\mathcal{H}(G;Q)$ provides a tighter approximation to the MAP than the local polytope. In fact, by controlling the set of QP edges Q, the set $\mathcal{H}(G;Q)$ provides a hierarchy of relaxations ranging from $\mathcal{I}(G)$ (when $Q = E$) to $\mathcal{L}(G)$ (when $|Q| = 0$). As the size of Q increases, the relaxation becomes tighter, but the non-convexity increases. However, since we have control over the degree of non-convexity, a judicious choice of the set Q will not only make the relaxation tighter, but also provide a good solution quality with respect to the optimization over the local polytope. Experimentally, we found this to be the case for almost all the instances.

Fig. 1(a) shows the high-level relationship among the marginal polytope and the inner and outer bounds and is based on [22]. Fig. 1(b) shows the hybrid set $\mathcal{H}(G;Q)$, which is qualitatively less non-convex than the inner bound, thereby decreasing the chances of getting trapped in a poor local optimum and is more accurate than the outer bound $\mathcal{L}(G)$. Fig. 1(c) shows a strategy to select the QP edges set Q. As the outer bound is tight for tree-structured graphs, we randomly choose a spanning tree of the graph with all its edges constituting the set L; the rest are the QP edges. We can also construct multiple such spanning trees independently with L being the union of their edges. Empirically, this method worked well in our experiments.

Proposition 3. *As the size of the QP edge set Q increases by 1, the number of parameters decreases by $O(k^2)$ and the number of constraints decreases by $O(2k)$.*

The above result shows that unlike the previous cluster-based approaches [19, 1], where the number of parameters increases exponentially w.r.t. the cluster size to make the relaxation tighter, the number of parameters decreases in our hybrid inner-outer bound based approach as it becomes tighter. This can potentially make the optimization easier at the expense of losing some convexity.

3 DC Formulation of MAP

First we explain the general DC optimization problem and briefly describe the concave-convex procedure, introduced in [26]. Consider the optimization problem: $\min\{g(x) : x \in \Omega\}$ where $g(x) = u(x) - v(x)$ is an arbitrary function with u, v being real-valued, differentiable *convex* functions and Ω being a constraint set. CCCP was originally proposed for Ω that is described by linear equality constraints, but [20] showed the same idea extends to any constraint set including non-convex constraints. The CCCP method provides an iterative procedure that generates a sequence of points x^l by solving the following convex program:

$$x^{l+1} = \arg\min\{u(x) - x^T\nabla v(x^l) : x \in \Omega\} \qquad (7)$$

Each iteration of CCCP decreases the objective function $g(x)$ for any Ω [20]. Furthermore, it is guaranteed to converge to a stationary point where the Karush-Kuhn-Tucker (KKT) conditions are satisfied when the constraint set Ω is convex [20]. Note that the objective $g(x)$ may be non-convex, which makes CCCP a general approach for non-linear optimization. The optimization problem over the set $\mathcal{H}(G;Q)$ is described as minimizing the following function subject to constraints Ω of Eq. (6):

$$\begin{aligned}
g(\boldsymbol{\mu};\theta,Q) = &-\sum_{(i,j)\in Q}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i)\mu(x_j) -\\
&\sum_{(i,j)\in L}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i,x_j) \qquad (8)
\end{aligned}$$

For the sake of readability, we drop the subscripts, using $\theta(x_i, x_j)$ for $\theta_{ij}(x_i, x_j)$, and $\mu(x_i, x_j)$ for $\mu_{ij}(x_i, x_j)$, as long as it is unambiguous. We also negate the MAP objective to get a minimization problem. The critical question in this reformulation into a DC program is how easy it is to perform the CCCP iteration in Eq. (7). Notice that the first term in $g(\boldsymbol{\mu}; \theta, Q)$ is quadratic and neither concave nor convex. However, noting that all the marginals must be positive, a simple substitution $\mu(x_i) = e^{y(x_i)}$, where $y(x_i)$ is unconstrained, makes it convex because the function $e^{h(x)}$ is convex when $h(x)$ is affine. The CCCP procedure has also been applied to optimize the Bethe free energy of a MRF [27]. Motivated by the fact that negative entropy of an edge is convex and the constraints in the outer bound are the same as in the Bethe free energy, we further perform the following optimality preserving modifications:

- For each LP edge in L, add and subtract the negative entropy: $-H_{ij} = \sum_{x_i,x_j} \mu(x_i, x_j) \log \mu(x_i, x_j)$ to $g(\boldsymbol{\mu}; \theta, Q)$.
- For each node i, add and subtract the term: $\sum_{x_i} e^{y(x_i)}$ to $g(\boldsymbol{\mu}; \theta, Q)$.

The entropy term ensures that all the edge marginals μ_{ij} are positive and the exponential term $e^{y(x_i)}$ ensures that all the node marginals are positive as well as simplifies the CCCP iteration. The objective $g(\boldsymbol{\mu}; \theta, Q)$ can now be written as the difference two functions u and v. The function $u(\boldsymbol{\mu}, \boldsymbol{y})$ is defined as:

$$-\sum_{(i,j)\in L} \sum_{x_i,x_j} \theta(x_i, x_j)\mu(x_i, x_j) - \sum_{(i,j)\in L} H_{ij} + \sum_{i,x_i} e^{y(x_i)} \quad (9)$$

The function $v(\boldsymbol{\mu}, \boldsymbol{y})$ is defined as:

$$\sum_{(i,j)\in Q} \sum_{x_i,x_j} \theta(x_i, x_j) e^{y(x_i)+y(x_j)} - \sum_{(i,j)\in L} H_{ij} + \sum_{i,x_i} e^{y(x_i)} \quad (10)$$

The modified constraint set Ω' is described by the following non-convex constraints:

$$\sum_{x_i,x_j} \mu(x_i, x_j) = 1 \; \forall (i,j) \in L;$$
$$\sum_{\hat{x}_j} \mu_{ij}(x_i, \hat{x}_j) = e^{y(x_i)} \; \forall i \in V, \forall x_i, \forall Nb_l(i) \quad (11)$$

Notice that the constraints in Eq. (11) are different from those in Eq. (6) that describe $\mathcal{H}(G; Q)$. This is deliberate to simplify the CCCP iteration without changing the feasible parameter space. First, the factorization constraint $\mu(x_i, x_j) = \mu(x_i)\mu(x_j)$ for all $(i,j) \in Q$ is now explicit in the objective (8). Second, we impose a restriction that for each node i, there should be at least a single neighbor j such that edge (i,j) is an LP edge or $(i,j) \in L$. This restriction will not allow all the edges to become QP, i.e., $Q \neq E$. The reason for this restriction is that we no longer enforce the normalization constraint for any node explicitly. But the constraints $\sum_{x_j} \mu(x_i, x_j) = e^{y(x_i)}$ and $\sum_{x_i,x_j} \mu(x_i, x_j) = 1$ directly imply $\sum_{x_i} e^{y(x_i)} = 1$. We note that restricting the set Q in this way does not compromise generality—if we want to make the set $Q = E$, then for each MRF node i, we introduce a dummy node d_i with a single state and make an edge (i, d_i) with every potential being zero. Then we can add this edge to the LP edges set L. This will satisfy the previous requirement. Also notice that by using the substitution $\mu(x_i) = e^{y(x_i)}$, we have made the constraint set non-convex. Let us denote the objective $g(\boldsymbol{\mu}; \theta, Q)$ (see Eq. (8)) with the exp-transformation as $g(\boldsymbol{\mu}, \boldsymbol{y}; \theta, Q)$. We also refer as Ω to the counterpart of Ω' which uses the variables $\mu(x_i)$ rather than $e^{y(x_i)}$. We now study the properties of the stationary points of $g(\boldsymbol{\mu}; \theta, Q)$ and $g(\boldsymbol{\mu}, \boldsymbol{y}; \theta, Q)$ that will help answer questions about the convergence properties of CCCP.

Proposition 4. *Every stationary point of the problem* $\min g(\boldsymbol{\mu}, \boldsymbol{y}; \theta, Q)$ *subject to* Ω' *is also a stationary point of the problem* $\min g(\boldsymbol{\mu}; \theta, Q)$ *subject to* Ω.

The above proposition has interesting consequences. For the LP relaxation case when $|Q| = 0$, the optimization of $g(\boldsymbol{\mu}; \theta, Q)$ subject to Ω is equivalent to solving the LP relaxation of MAP and there are no local optima. However, even for the case when $|Q| = 0$, the second optimization problem of $g(\boldsymbol{\mu}, \boldsymbol{y}; \theta, Q)$ over Ω' is a non-convex optimization problem as the equality constraints are not linear. However, we have shown that every stationary point of the second optimization problem is also a stationary point of the LP, which shows that there are no local optima in the second problem even though it is non-convex (only for the case $|Q| = 0$). Thus, achieving the stationary point of the second optimization problem using CCCP will yield the global optima for the LP relaxation.

Each iteration of CCCP is given by:

$$\min_{\boldsymbol{\mu}, \boldsymbol{y}} -\sum_{(i,j)\in L} \sum_{x_i,x_j} \theta(x_i, x_j)\mu(x_i, x_j) - \sum_{(i,j)\in L} H_{ij} + \sum_{i,x_i} e^{y(x_i)}$$
$$- \sum_{(i,j)\in L} \sum_{x_i,x_j} \mu(x_i, x_j) \nabla_{\mu(x_i,x_j)} v - \sum_{i\in V, x_i} y(x_i) \nabla_{y(x_i)} v$$

subject to constraints Ω' of Eq. (11), which are non-convex. Thus it might appear that we may not be able to optimally solve the CCCP iteration. However this is remedied by again substituting $\mu(x_i) = e^{y(x_i)}$ back to the above problem resulting in:

$$\min_{\boldsymbol{\mu},\boldsymbol{y}} - \sum_{(i,j)\in L}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i,x_j) - \sum_{(i,j)\in L} H_{ij} + \sum_{i,x_i}\mu(x_i)$$
$$- \sum_{(i,j)\in L}\sum_{x_i,x_j}\mu(x_i,x_j)\nabla_{\mu(x_i,x_j)}v - \sum_{i\in V,x_i}\log\mu(x_i)\nabla_{y(x_i)}v \tag{12}$$

$$subject\ to: \sum_{x_i,x_j}\mu(x_i,x_j) = 1\ \forall(i,j)\in L;$$
$$\sum_{\hat{x}_j}\mu_{ij}(x_i,\hat{x}_j) = \mu(x_i)\ \forall i \in V, \forall x_i, \forall Nb_l(i) \tag{13}$$

Note that this re-substitution $\mu(x_i) = e^{y(x_i)}$ is in the context of CCCP iteration (7); it does not undo the substitution made in the context of the objective (8)—in (12) we still have the gradient $\nabla_{y(x_i)}v$ w.r.t. $y(x_i)$. Furthermore, the above optimization problem is a standard convex optimization problem with convex objective and linear equality constraints. Thus there are no local optima, and strong duality holds as we can show that Slater's conditions hold [4].

Proposition 5. *CCCP converges to a stationary point of the problem* $\min g(\boldsymbol{\mu},\boldsymbol{y};\theta,Q)$ *subject to* Ω'.

4 Message-Passing Implementation of CCCP

We now describe how the CCCP iterations can be implemented using a message-passing paradigm. At a high level, CCCP updates typically require a double loop. The outer loop computes the gradient $\nabla v(\mu^l, y^l)$ and the inner loop solves the optimization problem of Eq. (12). Both steps can be implemented using message passing on the MRF as shown below.

4.1 The Gradient of $v(\mu, y)$ and Outer Loop Message Passing

The gradient w.r.t. $y(x_i)$ is given by:

$$\nabla_{y(x_i)}v(\mu,y) = e^{y(x_i)} + e^{y(x_i)}\sum_{j\in Nb_q(i)} e^{y(x_j)}\theta(x_i,x_j) \tag{14}$$

where $Nb(i)$ denotes the neighbors of a node i and $Nb_q(i)$ denotes the QP neighbors of a node i: $Nb_q(i) = \{j : j \in Nb(i) \wedge (i,j) \in Q\}$. The second term in the above equation can be computed using message passing on the graph. The other gradient, $\nabla_{\mu(x_i,x_j)}v(\mu,y) = 1 + \log\mu(x_i,x_j)$, does not require any message passing.

4.2 CCCP Inner Loop and Message Passing

We now describe how to perform the CCCP iteration of Eq. (12). Because the dual of this optimization problem is unconstrained and has simpler structure, we will perform block coordinate ascent in the dual. First we introduce the following Lagrange multipliers:

$$\lambda_{ij}: \sum_{x_i,x_j}\mu(x_i,x_j) = 1;\quad \lambda_{ji}(x_i): \sum_{\hat{x}_j}\mu_{ij}(x_i,\hat{x}_j) = \mu(x_i)$$

The dual function is $q(\boldsymbol{\lambda}) = \inf_{\mu,y} L(\boldsymbol{\mu},\boldsymbol{y},\boldsymbol{\lambda})$, where $L(\cdot)$ denotes the Lagrangian. We can solve this optimization problem by solving for the KKT conditions, resulting in:

$$\mu(x_i) = \frac{\nabla_{y(x_i)}v}{1+\sum_{k\in Nb_l(i)}\lambda_{ki}(x_i)}$$
$$\mu(x_i,x_j) = e^{\theta(x_i,x_j)+\nabla_{\mu(x_i,x_j)}v+\lambda_{ij}(x_j)+\lambda_{ji}(x_i)-\lambda_{ij}-1} \tag{15}$$

Substituting these back into the Lagrangian, we get the dual. The dual can be maximized iteratively by using the block coordinate ascent: hold all the λ's fixed except one and optimize the dual w.r.t. a single λ [2]. This iterative procedure is also guaranteed to converge to the optimum of the convex program of Eq. (12).

Proposition 6. *The Lagrange multipliers λ_{ij}, $\lambda_{ij}(x_j)$ can be derived by using an inner loop in the CCCP indexed by τ such that the multipliers for each edge $(i,j)\in L$ are updated once per inner loop. The update equations are as follows:*

$$\lambda_{ij}^{\tau+1}(x_j) = W\left(\frac{\nabla_{y(x_j)}v\ e^{\sum_{k\in Nb_l(j)\setminus i}\lambda_{kj}^{\tau}(x_j)+1}}{\sum_{x_i} e^{\theta(x_i,x_j)+\nabla_{\mu(x_i,x_j)}v+\lambda_{ji}^{\tau}(x_i)-\lambda_{ij}^{\tau}-1}}\right)$$
$$-1-\sum_{k\in Nb_l(j)\setminus i}\lambda_{kj}^{\tau}(x_j) \tag{16}$$

$$e^{\lambda_{ij}^{\tau+1}} = \sum_{x_i,x_j} e^{\theta(x_i,x_j)+\nabla_{\mu(x_i,x_j)}v+\lambda_{ij}^{\tau}(x_j)+\lambda_{ji}^{\tau}(x_i)-1} \tag{17}$$

where $W(\cdot)$ is the Lambert W-function.

The Lambert W-function is the multi-valued inverse of the function $w \mapsto we^w$ and is useful in many engineering applications such as jet fuel planning and enzyme kinematics [5] . In our case, the argument of $W(\cdot)$ in Eq. (16) is always positive and the W-function is properly defined over this range. Furthermore, these dual updates can be reinterpreted in terms of the primal parameters, which alleviates the need to store the Lagrange multipliers explicitly and greatly simplifies the implementation. The proof is given in the supplementary material. The message-passing procedure is detailed in Alg. 1. The superscripts on parameters (e.g. Z^τ) differentiate between the old parameters and the new updated ones. The function *Normalize*$(\cdot)$ normalizes a distribution by dividing each element by their sum. The inner loop convergence is detected by checking if the node and edge marginals sum up to 1, with some tolerance allowed (10^{-3}).

Algorithm 1: Graph-based message passing for MAP

```
input: Graph $G = (V, E)$ and potentials $\theta_{ij}$ per edge
1  repeat
2      Send message $\delta$ to $j \in Nb_q(i)$:
       $\delta_{i\to j}(x_j) = \sum_{x_i} \mu(x_i)\theta(x_i, x_j)$ for each node $i \in V$
3      $\nabla_{y(x_i)} v = \mu(x_i)\big(1 + \sum_{j\in Nb_q(i)} \delta_{j\to i}(x_i)\big)\ \forall i \in V$
4      Initialize $\mu^0(x_i) = \nabla_{y(x_i)} v\ \forall i \in V$,
       $\mu^0(x_i, x_j) = \mu(x_i, x_j)e^{\theta(x_i,x_j)}\ \forall (i, j) \in L$
5      repeat
6          foreach edge $(i, j) \in L$ do
7              $Normalize\big(\mu^\tau(x_i, x_j)\big)$
8              $Z^\tau(x_j) = \nabla_{y(x_j)} \exp\big(\frac{\nabla_{y(x_j)} v}{\mu^\tau(x_j)}\big)/\sum_{x_i} \mu^\tau(x_i, x_j)$
9              $\mu^{\tau+1}(x_i, x_j) =$
               $\mu^\tau(x_i, x_j) \exp\left(W(Z^\tau(x_j)) - \frac{\nabla_{y(x_j)} v}{\mu^\tau(x_j)}\right)$
10             $\mu^{\tau+1}(x_j) = \nabla_{y(x_j)} v/W(Z^\tau(x_j))$
11             Repeat steps 8 to 10 analogously for node $x_i$
12     until inner loop converges
13 until outer loop converges
return: The decoded complete integral assignment
```

Complexity From Proposition 3, the total number of parameters is $O(k^2|L| + nk)$, where k is the domain size. The total space complexity of Alg. 1, including the space for the δ messages, is $O(k^2|L| + nk + |Q|k)$. The time complexity is $O\big(Tk^2\big(|Q| + I|L|\big)\big)$ where T denotes the total number of outer loop iterations and I denotes the total number of inner loop iterations.

4.3 DC programs and proximal minimization

We now highlight a close relationship between DC programs, CCCP and proximal minimization algorithms based on Bregman divergence. Proximal minimization schemes solve a convex program $\min_{\mu\in\Omega} g(\mu)$ indirectly via a sequence of problems of the form [11]:

$$\mu^{n+1} = \arg\min_{\mu\in\Omega} \big\{g(\mu) + \frac{1}{\omega^n} D_f(\mu||\mu^n)\big\} \qquad (18)$$

where D_f is a generalized distance function and ω^n a positive weight. We focus on Bregman divergence based distance function defined as $D_f(\mu||\nu) = f(\mu) - f(\nu) - \langle\nabla f(\nu), \mu - \nu\rangle$, where f is a strictly convex, differentiable function (also called a Bregman function).

Proposition 7. *Each proximal iteration with Bregman divergence based distance function and a fixed weight ω is equivalent to the CCCP iteration for the DC program $\min_{\mu\in\Omega}\{u(\mu) - v(\mu)\}$ where $u(\mu) = g(\mu) + \frac{1}{\omega} f(\mu)$ and $v(\mu) = \frac{1}{\omega} f(\mu)$.*

Note that the DC program in the above proposition is equivalent to the original convex program $\min_{\mu\in\Omega} g(\mu)$. We also note that the weights ω^n can be adjusted for faster convergence in proximal schemes, but they can also be set to constant as the Bregman distance function D_f itself converges to zero as the algorithm approaches the optimum [11]. Furthermore, the DC program view can also simulate changing weight scenario as the DC decomposition $u(\mu) = g(\mu) + \frac{1}{\omega} f(\mu)$ and $v(\mu) = \frac{1}{\omega} f(\mu)$ is valid for any weight ω. Ravikumar et al. [11] propose several globally convergent algorithms to solve the LP relaxation based on different choices of the Bregman function f such as entropic, quadratic, etc. Such algorithms can be interpreted as DC programs. Moreover, the proximal minimization schemes are only applicable to convex functions, while DC programs need not be convex and are therefore more general. Nonetheless, the connection we established is important. For example, CCCP is generally considered to have a first-order convergence rate [15]. However Bregman projections for certain choices of Bregman functions have fast superlinear convergence [11], implying a similar convergence rate for CCCP too in these cases. To recast the proximal iteration of Eq. (18) as Bregman projections, the Bregman functions are restricted to be of Legendre type in [11]. This prohibits them from using a tree-reweighted entropy-based Bregman function. However the DC view of the proximal approach does not pose this restriction and any Bregman function can be used.

5 Experiments

We experimented on two datasets: Binary integer quadratic (Biq) problems publicly available from the *Biq Mac* library [14, 13] and 50×50 2D grid graphs with potentials set using the Potts and Ising models similar to [11, 18]. We compared several algorithms including CCCP, entropic proximal solver (EP) from [11], MPLP with cluster based tightening [19] and max-product (MP) [10]. All the algorithms except MPLP were implemented in JAVA; MPLP's code was provided by the authors and the standard settings were used. Our main goal in these experiments was twofold: a) show that the optimization over the hybrid bound $\mathcal{H}(G;Q)$ provides a tighter approximation of the MAP problem and a significantly better solution quality than LP relaxation over the outer bound; and b) show that CCCP converges to a global optimum of the LP relaxation over the outer bound by comparing its quality against EP, which optimizes the same objective function.

Table 1 shows the results for the Biq benchmarks. These benchmarks are particularly interesting in that the LP relaxation on these problems is loose. Therefore tightening the outer bound is particularly beneficial. The last two columns show time (in sec.), while the other columns show the best quality achieved for a given instance. For details on generating these Biq problems and formulating them as MRFs we refer to [14, 13]. The CCCP algorithm over the hybrid set

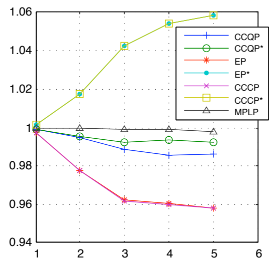
Table 1: Quality and runtime comparisons for the Biq benchmarks. ‘Inst.’ is the instance name and ‘Opt.’ is the known optimal quality from [14]. ‘–’ denotes failure of the algorithm to produce a solution.

Inst.	Opt.	CCQP	$\|L\|$	CCQP*	MPLP	MPLP*	CCCP	CCCP*	EP	EP*	T_{ccqp}/r	T_{mplp}
100-1	7970	**7970**	0.553	7963.9	7594	8743.8	7634	10147.6	7634	10183.5	4.26/0	3.6
100-2	11036	**11036**	0.512	11035.9	10866	11341	10976	12231.1	11014	12228.41	33.4/1	10.1
100-3	12723	**12723**	0.514	12722.9	12723	12988.1	12723	13652.5	12723	13598.1	0.07/0	62.5
100-4	10368	**10368**	0.532	10367.7	10076	10728.1	10172	12044.9	10172	12059.2	2.40/0	4.4
100-5	9083	**9083**	0.562	9082.9	8685	9506.8	8914	10629.9	8914	10613.9	14.4/1	5.8
100-6	10210	**10210**	0.533	10099.9	9704	11553.7	10210	12937.6	10210	13045.3	6.9/1	3.5
100-7	10125	**10125**	0.554	10054.6	9952	10904.4	9977	11978.3	9977	11973.7	3.9/1	2.8
100-8	11435	**11435**	0.528	11434.6	11310	11793.8	11397	12615.4	11397	12436.1	4.6/0	3.3
100-9	11455	**11455**	0.533	11454.9	11361	11628.4	11368	12259.6	11348	12219.2	5.8/0	3.6
100-10	12565	**12565**	0.536	12564.6	12459	12831.1	12511	13676.8	12511	13617.9	40.3/0	6.1
250-1	45607	**45607**	0.477	45606.9	41862	54477.5	44872	77827.5	44956	77382.8	56.1/0	4498.1
250-2	44810	**44810**	0.484	44506.4	38659	55664.5	44234	77469.2	44234	77314.6	209.5/2	775.9
250-3	49037	**49037**	0.477	49011.6	44954	57118.4	48980	80278.5	48980	80015.4	3.98/1	822.1
250-4	41274	**41274**	0.474	41273.4	34670	51687.8	41124	74784.7	40637	74608.7	146.2/5	6164.9
250-5	47961	47939	0.520	47938.9	44461	56607.6	47843	79256.0	47843	78912.9	196.1/7	2708.8
250-6	41014	**41014**	0.475	39748.4	34092	53157.7	40225	77758.0	40225	77617.2	289.1/11	1271.5
250-7	46757	**46757**	0.477	46521.6	41974	56397.7	46449	79393.9	46431	79075.1	61.1/0	1992.5
250-8	35726	35336	0.489	35504.1	29312	49492.6	34656	72101.1	34656	71896.3	380.4/6	477.4
250-9	48916	**48916**	0.473	48907.6	44057	58305.3	48677	81115.7	48615	80684.5	353.5/0	3236.1
250-10	40442	40330	0.497	40032.6	32534	52525.6	40194	75142.7	40198	74783.1	116.8/7	908.1
1b.20	133	**133**	0.197	96.9	98	260.7	98	394.5	98	387.2	1.4/0	0.82
2b.30	121	**121**	0.121	86.9	67	345.7	65	505.8	65	504.2	2.2/0	0.08
3b.40	118	**118**	0.097	46.1	102	416.3	102	612.1	102	609.1	66.6/0	240.4
4b.50	129	**129**	0.079	23.1	85	548.3	54	799.8	54	797.5	3.7/0	702.7
5b.60	150	**150**	0.065	46.7	88	632.6	150	922.1	150	918.0	0.17/0	71.9
6b.70	146	133	0.057	14.8	63	758.1	62	1101.6	62	1099.6	49.2/0	714.16
7b.80	160	**160**	0.050	14.8	122	840.1	54	1220.1	54	1218.4	28.3/0	66.5
8b.90	145	**145**	0.040	13.8	58	1435.5	72	1389.4	72	1386.7	47.7/0	0.27
9b.100	137	135	0.032	12.9	58	1656.5	120	1599.6	120	1597.1	36.7/0	0.37
10b.125	154	**154**	0.044	10.8	–	–	104	1898.3	104	1895.6	84.9/0	–

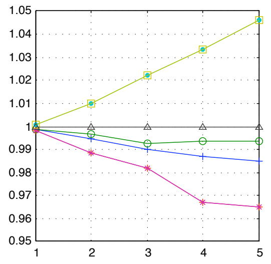
$\mathcal{H}(G;Q)$ with $|Q|>0$ is called ‘CCQP’. When optimizing the LP relaxation ($|Q|=0$), we simply refer to it as ‘CCCP’. An asterisk (*) for a given algorithm denotes the best objective the algorithm achieved upon convergence (can be fractional); the name without it shows the decoded integral assignment. For example, MPLP* denotes the upper bound which MPLP minimizes, whereas MPLP denotes the integral quality. We chose Biq problems with a varying number of nodes and edge densities. The instances ‘100-1’ to ‘100-10’ have 100-node graphs with edge density 0.1, and edge potentials in the range $[-200, 200]$ (referred to as ‘bqp100-i’ in [13]). The instances ‘250-1’ to ‘250-10’ have 250-node graphs with edge density 0.1 (‘bqp250-i’ in [13]). The instances $1b.n$ to $10b.n$ are complete graphs (density 1) with ‘n’ denoting the number of nodes (‘gka$i$b.n’ in [13]). We used the decoding scheme of [12] for all the CCCP and EP algorithms.

Our hybrid bound based approach ‘CCQP’ performs quite well in this dataset and achieves the optimal solution for 26 out of 30 instances. The column ‘$|L|$’ shows the fraction of the LP edges ($|L|/|E|$) in the hybrid bound. We selected the LP edges using the strategy shown in Fig. 1(c): construct randomly a number of independent spanning tress and make L the union of their edges. For instances ‘100-1’ to ‘250-10’, we used 8 spanning trees and for instances ‘1b.n’ to ‘10b.n’, only 2 trees were required. Despite such low fraction of LP edges, our approach returns optimal solutions indicating that a judicious choice of the LP edges can mitigate the adverse effect of non-convexity. This observation is further reinforced by noting the fractional objective under ‘CCQP*’, which is quite close to the true optimum. On the other hand, the relaxed LP objective (under ‘CCCP*’ and ‘EP*’) seems to be loose, specially for ‘250-i’ and ‘gka’ instances. The minor differences between ‘CCCP*’ and ‘EP*’ are mainly because the normalization constraints were enforced only to an accuracy of 10^{-3} for faster convergence. The decoded solution quality for CCCP and EP is not as good as CCQP. MPLP provides tighter upper bounds than CCCP (under ‘MPLP*’), but it fails to translate it into good solution quality as the upper bounds are still loose when compared to the true optimum.

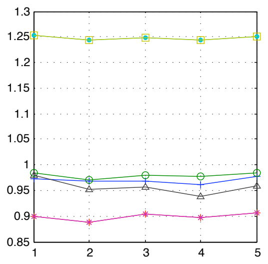
The last two columns provide runtime comparisons between CCQP and MPLP. The total outer loop iterations in CCQP was fixed to 1100 and inner loop iterations to 60. MPLP was run for 2 hours. The time in these columns shows when the best integral solution quality was first achieved for each algorithm. In addition, for CCQP, the performance was dependent on the choice of spanning trees. Therefore we reported the best of 12 runs. The r value in ‘T_{ccqp}/r’ denotes the best run. For most instances, CCQP was quite robust. For the ‘250-i’ dataset, a higher number of runs was required. These instances are also the most difficult of all the instances even for the optimal approach of [14], which required ≈ 4500 sec. at the minimum and up to 316000 sec. for some instances. CCQP on the other hand terminates within ≈ 650 sec. for all



(a)



(b)



(c)

Figure 2: Quality comparison for 50×50 grids: a) Potts model with variable domain size 4 and b) with domain size 8; c) Ising model (domain size = 2). The x-axis shows the coupling parameter β and y-axis the normalized quality.

the '250-i' instances. The complete set of results for max-product in shown in the supplementary material. MP converges and achieves good quality for the '100-i', '250-i' instances, but for 'gka' instances it performs poorly achieving a quality of 0 for most instance.

We also experimented with grid graphs with edge potentials set using the Potts model [11] and the Ising model [18]. For both models, the edge potentials were sampled from $\mathcal{U}[-\beta, \beta]$ for each edge independently where β is the coupling strength parameter. The unary potentials were sampled from $\mathcal{U}[-1, 1]$. For these graphs, MPLP performs quite well in minimizing the upper bound. Therefore we report normalized values for all the algorithms in Fig. 2(a–c) where the best MPLP upper bound (U.B.) is denoted as 1. Fig. 2(a,b) show that CCQP again achieves better solution quality than the LP relaxation (CCCP, EP). The decoded quality is within 98.5% of optimal for all the settings. For the Potts graphs, CCQP used 2 spanning trees for the set L with the fraction of LP edges being $\approx .75$. The CCQP fractional objective ('CCQP*') is very close to the MPLP U.B., which further confirms that the outer bound is tightened significantly using QP edges. The plots for 'CCCP*' and 'EP*' almost overlap, which shows that CCCP converges to the same quality as EP for the LP relaxation. For the Ising graphs (Fig. 2(c)), CCQP provides better quality (within 97% of the optimal on average) than all the other algorithms including MPLP. Interestingly, the performance of CCQP improved with a lower fraction of LP edges ($\approx .45$) than in the Potts model. This can be explained by the relaxed LP objective ('CCQP*' or 'EP*'), which is relatively loose (≈ 1.25) as opposed to (≈ 1.06) in Potts. Therefore, a higher fraction of QP edges is required to tighten the outer bound. This can also help explain the deteriorated performance of MPLP, which may need even higher order clusters to further tighten the outer bound. MP provided the worse quality for all these problems and is not plotted for clarity sake. The normalized quality MP achieves for each coupling strength parameter setting is: (0.99, 0.93, 0.82, 0.77, 0.75) for the Potts model with domain size 4 and (0.99, 0.94, 0.89, 0.82, 0.77) with size 8. For the Ising graphs, it was (0.60, 0.63, 0.62, 0.61, 0.61). Thus CCQP provided consistent and much better performance than MP.

6 Conclusion and Future Work

We present a flexible framework for tightening the outer bound on the MAP marginal polytope by combining it with the inner bound. This hybrid bound is parameterized by a set of edges such that on one extreme it resembles the *exact* but non-convex inner bound and on the other extreme it becomes the well-known LP relaxation for MAP. We formulated the optimization over this hybrid bound as a DC program and used the concave-convex procedure to derive a convergent message-passing algorithm, guaranteed to solve the LP relaxation optimally. We also identified a close relationship between the MAP DC formulations and the Bregman divergence based proximal methods. Our experiments show that the hybrid bound approximates the MAP significantly better than the marginal polytope despite its non-convexity, and it achieves the optimal solution on a range of problems.

Our approach presents notable opportunities for future work. Our current scheme of tightening the outer bound is *static* in that the QP edges set is predetermined before the optimization procedure. This can be significantly improved by taking an adaptive tightening approach similar to cluster pursuit based methods [19]. Furthermore, the DC programming framework and CCCP allow for including additional constraints for the given optimization problem. We plan to develop message-passing algorithms that tighten the outer bound using additional constraints based on the cut polytope [17]. These enhancements could further improve the performance of our approach.

Acknowledgments

This work was funded in part by the NSF grants IIS-1116917, IIS-1160424; Marc Toussaint was supported by the German Research Foundation (DFG), Emmy Noether fellowship TO 409/1-3 and SPP grant TO 409/7-1.

References

[1] Dhruv Batra, Sebastian Nowozin, and Pushmeet Kohli. Tighter relaxations for MAP-MRF inference: A local primal-dual gap based separation algorithm. In *International Conf. on Artificial Intelligence and Statistics*, pages 146–154, 2011.

[2] Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Cambridge, MA, USA, 1999.

[3] Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B*, 36(2):192–236, 1974.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[5] R. M. Corless, G. H. Gonnet, D. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5:329–359, 1996.

[6] Amir Globerson and Tommi Jaakkola. Fixing Max-Product: Convergent message passing algorithms for MAP LP-relaxations. In *Advances in Neural Information processing Systems*, pages 553–560, 2007.

[7] Vladimir Jojic, Stephen Gould, and Daphne Koller. Accelerated dual decomposition for MAP inference. In *International Conf. on Machine Learning*, pages 503–510, 2010.

[8] V Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28: 1568–1583, 2006.

[9] N. Komodakis, N. Paragios, and G. Tziritas. MRF energy minimization and beyond via dual decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(3):531 –552, 2011.

[10] Judea Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann Publishers Inc., 1988.

[11] Pradeep Ravikumar, Alekh Agarwal, and Martin J. Wainwright. Message-passing for graph-structured linear programs: Proximal methods and rounding schemes. *Journal of Machine Learning Research*, 11: 1043–1080, 2010.

[12] Pradeep Ravikumar and John Lafferty. Quadratic programming relaxations for metric labeling and Markov random field MAP estimation. In *International Conf. on Machine Learning*, pages 737–744, 2006.

[13] Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele. Biq Mac Solver – binary quadratic and max-cut solver, 2010. URL `http://biqmac.uni-klu.ac.at/`.

[14] Franz Rendl, Giovanni Rinaldi, and Angelika Wiegele. Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations. *Mathematical Programming*, 121(2):307–335, 2010.

[15] Ruslan Salakhutdinov, Sam Roweis, and Zoubin Ghahramani. On the convergence of bound optimization algorithms. In *International Conf. on Uncertainty in Artificial Intelligence*, pages 509–516, 2003.

[16] David Sontag, Amir Globerson, and Tommi Jaakkola. Introduction to dual decomposition for inference. *Optimization for Machine Learning*, 2010.

[17] David Sontag and Tommi Jaakkola. New outer bounds on the marginal polytope. In *Advances in Neural Information Processing Systems*, 2007.

[18] David Sontag and Tommi Jaakkola. New outer bounds on the marginal polytope. In *Advances in Neural Information Processing Systems*, pages 1393–1400, 2008.

[19] David Sontag, Talya Meltzer, Amir Globerson, Tommi Jaakkola, and Yair Weiss. Tightening LP relaxations for MAP using message passing. In *International Conf. on Uncertainty in Artificial Intelligence*, pages 503–510, 2008.

[20] Bharath Sriperumbudur and Gert Lanckriet. On the convergence of the concave-convex procedure. In *Advances in Neural Information Processing Systems*, pages 1759–1767, 2009.

[21] Martin Wainwright, Tommi Jaakkola, and Alan Willsky. MAP estimation via agreement on (hyper)trees: Message-passing and linear programming approaches. *IEEE Transactions on Information Theory*, 51:3697–3717, 2002.

[22] Martin J. Wainwright and Michael I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1:1–305, 2008.

[23] T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1165–1179, 2007.

[24] Tomás Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *Conf. on Computer Vision and Pattern Recognition*, 2008.

[25] Chen Yanover, Talya Meltzer, and Yair Weiss. Linear programming relaxations and belief propagation – an empirical study. *Journal of Machine Learning Research*, 7:1887–1907, 2006.

[26] A. L. Yuille and Anand Rangarajan. The concave-convex procedure. *Neural Computation*, 15:915–936, 2003.

[27] Alan L. Yuille. CCCP algorithms to minimize the bethe and kikuchi free energies: Convergent alternatives to belief propagation. *Neural Computation*, 14: 1691–1722, 2002.

Message-Passing Algorithms for MAP Estimation Using DC Programming (Supplementary Material)

Akshat Kumar
Univ. of Massachusetts Amherst

Shlomo Zilberstein
Univ. of Massachusetts Amherst

Marc Toussaint
Freie Universität Berlin

A Detailed Proofs and Derivations

A.1 Proposition 3

For the QP edges, it holds that $\mu_{ij}(x_i, x_j) = \mu_i(x_i)\mu_j(x_j)$. Instead of having a linear objective function $(\boldsymbol{\mu}\cdot\boldsymbol{\theta})$, we can substitute $\mu_{ij}(x_i, x_j)$ by $\mu_i(x_i)\mu_j(x_j)$ in the objective. Thus we no longer need to store the parameter $\mu_{ij}(x_i, x_j)$ nor the mean-field constraint explicitly for QP edges. Therefore, the total number of parameters is $O(k^2|L| + nk)$ and the total number of constraints is $O(2|L|k + n)$ where $L = E\backslash Q$. As the size of Q increases by 1, the size of the set L decreases by 1. This proves the proposition.

A.2 Proposition 4

The optimization problem involving the function $g(\boldsymbol{\mu}, \boldsymbol{y}; \theta, Q)$ over Ω' is given by[1]:

$$\min_{\boldsymbol{\mu},\boldsymbol{y}} -\sum_{(i,j)\in Q}\sum_{x_i,x_j}\theta(x_i,x_j)e^{y(x_i)+y(x_j)} - \sum_{(i,j)\in L}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i,x_j) \tag{19}$$

$$\textit{subject to}: \sum_{x_i,x_j}\mu(x_i,x_j) = 1\ \forall(i,j)\in L;$$

$$\sum_{\hat{x}_j}\mu_{ij}(x_i,\hat{x}_j) = e^{y(x_i)}\ \forall i\in V, \forall x_i, \forall Nb_l(i) \tag{20}$$

The Lagrangian $L(\boldsymbol{\mu}, \boldsymbol{y}, \lambda)$ is given by:

$$L(\boldsymbol{\mu},\boldsymbol{y},\lambda) = -\sum_{(i,j)\in Q}\sum_{x_i,x_j}\theta(x_i,x_j)e^{y(x_i)+y(x_j)} - \sum_{(i,j)\in L}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i,x_j) + \sum_{ij}\lambda_{ij}\Big\{\sum_{x_i,x_j}\mu(x_i,x_j) - 1\Big\} + \sum_{i\in V}\sum_{j\in Nb_l(i)}\sum_{x_i}\lambda_{ji}(x_i)\Big\{\sum_{x_j}\mu_{ij}(x_i,x_j) - e^{y(x_i)}\Big\} \tag{21}$$

Notice that in the above Lagrangian, we ignored the nonnegativity constraints associated with $\mu(x_i, x_j) \geq 0$ for simplicity as they remain the same for both versions of the optimization problem. Now the KKT conditions for the stationary point are given by:

$$\frac{\partial L}{\partial \mu(x_i,x_j)} = -\theta(x_i,x_j) + \lambda_{ij} + \lambda_{ji}(x_i) + \lambda_{ij}(x_j) = 0 \tag{22}$$

The second KKT condition is:

$$\frac{\partial L}{\partial y(x_i)} = -\sum_{j\in Nb_q(i)}\sum_{x_j}\theta(x_i,x_j)e^{y(x_i)+y(x_j)} - \sum_{j\in Nb_l(i)}\lambda_{ji}(x_i)e^{y(x_i)} = 0 \tag{23}$$

which can be further simplified to:

$$\sum_{j\in Nb_q(i)}\sum_{x_j}\theta(x_i,x_j)e^{y(x_j)} + \sum_{j\in Nb_l(i)}\lambda_{ji}(x_i) = 0 \tag{24}$$

Now consider the second optimization problem of optimizing $g(\boldsymbol{\mu}; \theta, Q)$ over Ω given as:

$$\min_{\boldsymbol{\mu},\boldsymbol{y}} -\sum_{(i,j)\in Q}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i)\mu(x_j) - \sum_{(i,j)\in L}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i,x_j) \tag{25}$$

$$\textit{subject to}: \sum_{x_i,x_j}\mu(x_i,x_j) = 1\ \forall(i,j)\in L;$$

$$\sum_{\hat{x}_j}\mu_{ij}(x_i,\hat{x}_j) = \mu(x_i)\ \forall i\in V, \forall x_i, \forall Nb_l(i) \tag{26}$$

Notice that every feasible point $(\boldsymbol{\mu'}, \boldsymbol{y'}) \in \Omega'$ of the first optimization problem can be transformed into a unique feasible point $\boldsymbol{\mu} \in \Omega$ of the second problem simply by setting $\mu(\cdot,\cdot) = \mu'(\cdot,\cdot)$ and setting $\mu(\cdot) = e^{y(\cdot)}$. Thus each stationary point of the first optimization problem is a feasible point of the second problem. We now show that this feasible point indeed satisfies the KKT conditions of the second optimization problem too, thus prov-

[1]Equation numbers continue the main paper.

ing the proposition. The Lagrangian $L(\boldsymbol{\mu}, \lambda)$ is given by:

$$L(\boldsymbol{\mu}, \lambda) = -\sum_{(i,j)\in Q}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i)\mu(x_j) - \sum_{(i,j)\in L}\sum_{x_i,x_j}\theta(x_i,x_j)\mu(x_i,x_j) + \sum_{ij}\lambda_{ij}\Big\{\sum_{x_i,x_j}\mu(x_i,x_j)-1\Big\} + \sum_{i\in V}\sum_{j\in Nb_l(i)}\sum_{x_i}\lambda_{ji}(x_i)\Big\{\sum_{x_j}\mu_{ij}(x_i,x_j)-\mu(x_i)\Big\} \tag{27}$$

Now the KKT conditions for the stationary point are:

$$\frac{\partial L}{\partial \mu(x_i,x_j)} = -\theta(x_i,x_j) + \lambda_{ij} + \lambda_{ji}(x_i) + \lambda_{ij}(x_j) = 0 \tag{28}$$

The above KKT condition matches exactly with the KKT condition of Eq. (22).

The second KKT condition is:

$$\frac{\partial L}{\partial \mu(x_i)} = -\sum_{j\in Nb_q(i)}\sum_{x_j}\theta(x_i,x_j)\mu(x_j) - \sum_{j\in Nb_l(i)}\lambda_{ji}(x_i) = 0 \tag{29}$$

The above condition is also exactly the same as of the condition in Eq. (24) by noting that $\mu(x_j) = e^{y(x_j)}$. Thus we have shown that every stationary point of the first optimization problem is a feasible point of the second optimization and also satisfies the KKT conditions.

A.3 Proposition 5

The above proposition can be proved using Zangwill's global convergence theorem [2] which has been used to prove the convergence of CCCP for the convex constraints [1, Thm. 4]. They also show another variant of CCCP with non-convex constraints also converges to a stationary point [1, Sec. 4.1]. In our case, we wish to prove the convergence of CCCP for the optimization problem $\min g(\boldsymbol{\mu}, \boldsymbol{y}; \theta, Q)$ of Eq. (19) over constraints Ω' of Eq. (20). In principle, we can use the analysis of [1, Sec. 4.1] that handles CCCP with non-convex constraints, however we do not use the variant of CCCP presented in that section as we have non-convex equality constraints rather than D.C. inequality constraints. For a brief overview of the Zangwill's theorem, we refer to [1, Thm. 2]. We will also use some background terms from [1] such as the notion of point-to-set map, details can be found in that paper.

Roughly speaking, the CCCP iteration of Eq. (12) defines a point-to-set map $x^{l+1} = A_{cccp}(x^l)$ where A_{cccp} is the optimization problem of Eq. (12). The main idea to prove the convergence of CCCP is two fold. First we show that a fixed point of A_{cccp} is also a stationary point of the D.C. program of Eq. (19). The fixed point $x^\star$ of A_{cccp} is given by the condition $x^\star = A_{cccp}(x^\star)$. This can be easily shown by writing the KKT conditions for A_{cccp} at $x^\star$ and showing that they also satisfy the KKT conditions for the D.C. program of Eq. (19) similar to Appendix A.2. This condition holds in our case; we skip the proof for brevity.

The second step is to show that the limit points of any sequence $\{x^l\}_{l=0}^{\infty}$ generated by A_{cccp} are the fixed points of A_{cccp}. This can be shown by using the conditions of [1, Thm. 2]. We do not show the proof in detail as it is similar to the proof of convergence of CCCP with *convex* constraints [1, Thm. 4]. We provide high level arguments as follows. The main reason is that although our original D.C. program of Eq. (19) has non-convex constraints, the CCCP iteration A_{cccp} we proposed in Eq. (12) is a convex optimization problem with linear equality constraints. Therefore the convergence of A_{cccp} is implied by [1, Thm. 4], which only requires A_{cccp} to be a convex optimization problem and be uniformly compact on the constraint set. We also highlight that the [1, Remark 7] holds in our case as the constraint set for A_{cccp} in Eq. (13) is compact.

A.4 Reinterpretation of the dual updates in terms of primal parameters

This appendix derives the updates used in the inner loop of Alg. 1. Let each step of dual coordinate ascent be indexed by superscripts τ starting from zero. Initially, we set all multipliers λs to zero. Let the outer loop iterations be indexed by subscripts n and let the current outer loop iteration be $n+1$. So we have:

$$\mu(x_i,x_j) = e^{\left\{\theta(x_i,x_j)+\nabla_{\mu(x_i,x_j)}v+\lambda_{ij}(x_j)+\lambda_{ji}(x_i)-\lambda_{ij}-1\right\}}$$
$$\boldsymbol{\mu^0(x_i,x_j) = \mu_n(x_i,x_j)\exp\{\theta(x_i,x_j)\}}$$
$$\mu(x_i) = \frac{\nabla_{y(x_i)}v}{1+\sum_{k\in Nb_l(i)}\lambda_{ki}(x_i)}$$
$$\boldsymbol{\mu^0(x_i) = \nabla_{y(x_i)}v}$$

For any inner loop iteration τ, we can realize the intermediate beliefs as:

$$\mu^\tau(x_i) = \frac{\nabla_{y(x_i)}v}{1+\sum_{k\in Nb_l(i)}\lambda^\tau_{ki}(x_i)}$$
$$\mu^\tau(x_i,x_j) = \mu_n(x_i,x_j)e^{\left\{\theta(x_i,x_j)+\lambda^\tau_{ij}(x_j)+\lambda^\tau_{ji}(x_i)-\lambda^\tau_{ij}\right\}}$$

Let us first consider the dual update for $\lambda^{\tau+1}_{ij}(x_j)$.

$$\lambda^{\tau+1}_{ij}(x_j) = W\Big[\frac{\nabla_{y(x_j)}v\; e^{\sum_{k\in Nb_l(j)\setminus i}\lambda^\tau_{kj}(x_j)+1}}{\sum_{x_i}\mu_n(x_i,x_j)\exp\{\theta(x_i,x_j)+\lambda^\tau_{ji}(x_i)-\lambda^\tau_{ij}\}}\Big] - 1 - \sum_{k\in Nb_l(j)\setminus i}\lambda^\tau_{kj}(x_j)$$

We can simplify the argument Z^τ of the lambert W-function as follows:

$$Z^\tau = \frac{\nabla_{y(x_j)} v \; e^{\sum_{k\in Nb_l(j)\setminus i} \lambda^\tau_{kj}(x_j)+1}}{\sum_{x_i} \mu_n(x_i,x_j) \exp\{\theta(x_i,x_j) + \lambda^\tau_{ji}(x_i) - \lambda^\tau_{ij}\}} \frac{e^{\lambda^\tau_{ij}}(x_j)}{e^{\lambda^\tau_{ij}}(x_j)}$$

$$Z^\tau = \frac{\nabla_{y(x_j)} v \; e^{\sum_{k\in Nb_l(j)} \lambda^\tau_{kj}(x_j)+1}}{\sum_{x_i} \mu_n(x_i,x_j) \exp\{\theta(x_i,x_j) + \lambda^\tau_{ij}(x_j) + \lambda^\tau_{ji}(x_i) - \lambda^\tau_{ij}\}}$$

$$Z^\tau = \frac{\nabla_{y(x_j)} v \;\exp\{\frac{\nabla_{y(x_j)}}{\mu^\tau(x_j)}\}}{\sum_{x_i} \mu^\tau(x_i,x_j)}$$

So now we have new $\lambda^{\tau+1}_{ij}(x_j)$ as:

$$\lambda^{\tau+1}_{ij}(x_j) = W\big[Z^\tau\big] - 1 - \sum_{k\in Nb_l(j)\setminus i} \lambda^\tau_{kj}(x_j)$$

Using the above equation to calculate the new $\mu^{\tau+1}(x_j)$, we have:

$$\mu^{\tau+1}(x_j) = \frac{\nabla_{y(x_j)} v}{1 + \sum_{k\in Nb_l(j)\setminus i} \lambda^\tau_{kj}(x_j) + \lambda^{\tau+1}_{ij}(x_j)}$$

$$\mu^{\tau+\mathbf{1}}(\mathbf{x_j}) = \frac{\nabla_{\mathbf{y(x_j)}}\mathbf{v}}{\mathbf{W}[\mathbf{Z}^\tau]}$$

The only other quantity affected by $\lambda^{\tau+1}_{ij}(x_j)$ is $\mu^{\tau+1}(x_i,x_j)$. We have:

$$\mu^{\tau+1}(x_i,x_j) = \mu_n(x_i,x_j) e^{\left\{\theta(x_i,x_j)+\lambda^\tau_{ji}(x_i)-\lambda^\tau_{ij}\right\}} e^{\lambda^{\tau+1}_{ij}(x_j)}$$

$$= \mu_n(x_i,x_j) e^{\left\{\theta(x_i,x_j)+\lambda^\tau_{ji}(x_i)-\lambda^\tau_{ij}\right\}} \frac{e^{W[Z]}}{e^{1+\sum_{k\in Nb_l(j)\setminus i} \lambda^\tau_{kj}(x_j)}}$$

Multiplying and dividing the above expression by $e^{\lambda^\tau_{ij}(x_j)}$, we get

$$= \mu_n(x_i,x_j) e^{\left\{\theta(x_i,x_j)+\lambda^\tau_{ij}(x_j)+\lambda^\tau_{ji}(x_i)-\lambda^\tau_{ij}\right\}} \frac{e^{W[Z]}}{e^{1+\sum_{k\in Nb_l(j)} \lambda^\tau_{kj}(x_j)}}$$

$$\mu^{\tau+\mathbf{1}}(\mathbf{x_i},\mathbf{x_j}) = \mu^\tau(\mathbf{x_i},\mathbf{x_j}) \exp\left(\mathbf{W}[\mathbf{Z}^\tau] - \frac{\nabla_{\mathbf{y(x_j)}}\mathbf{v}}{\mu^\tau(\mathbf{x_j})}\right)$$

A.5 Proposition 7

Substituting the definition of Bregman function in Eq. (18) we get the proximal iteration as:

$$\mu^{n+1} = \arg\min_{\mu\in\Omega} \left\{ g(\mu) + \frac{1}{\omega}\Big(f(\mu) - f(\mu^n) - \nabla f(\mu^n)\mu + \nabla f(\mu^n)\mu^n \Big) \right\}$$

$$= \arg\min_{\mu\in\Omega} \left\{ g(\mu) + \frac{1}{\omega}\big(f(\mu) - \nabla f(\mu^n)\cdot\mu \big) \right\}$$

Consider the D.C. program:

$$\min_{\mu\in\Omega}\{u(\mu) - v(\mu)\}$$

equivalent to the original problem $\min_{\mu\in\Omega} g(\mu)$ with $u(\mu) = g(\mu) + \frac{1}{\omega} f(\mu)$ and $v(\mu) = \frac{1}{\omega} f(\mu)$. The CCCP iteration of Eq. (7) is given as:

$$\arg\min_{\mu\in\Omega}\{g(\mu) + \frac{1}{\omega} f(\mu) - \frac{1}{\omega}\nabla f(\mu^n)\cdot\mu\}$$

which is equivalent to the previous proximal scheme iteration.

B Experimental Results for Max-Product for Biq Instances

Table 1 shows the complete set of results, detailing the solution quality achieved by max-product.

Table 1: Solution quality comparisons for max-product

Instance	Optimal	MP
100-1	7970	7822
100-2	11036	11036
100-3	12723	12723
100-4	10368	10368
100-5	9083	9083
100-6	10210	10065
100-7	10125	10034
100-8	11435	11435
100-9	11455	11455
100-10	12565	12565
250-1	45607	45607
250-2	44810	44810
250-3	49037	49037
250-4	41274	41270
250-5	47961	47961
250-6	41014	41014
250-7	46757	46757
250-8	35726	34450
250-9	48916	48916
250-10	40442	40442
1b.20	133	0
2b.30	121	0
3b.40	118	0
4b.50	129	0
5b.60	150	0
6b.70	146	61
7b.80	160	0
8b.90	145	0
9b.100	137	0
10b.125	154	0

References

[1] Bharath Sriperumbudur and Gert Lanckriet. On the convergence of the concave-convex procedure. In *Advances in Neural Information Processing Systems*, pages 1759–1767, 2009.

[2] W. Zangwill. *Nonlinear Programming: A Unified Approach.* Prentice Hall, Englewood Cliffs, N.J., 1969.