Subproblem-Tree Calibration: A Unified Approach to Max-Product Message Passing

Huayan Wang HUAYANW@CS.STANFORD.EDU
Computer Science Department, Stanford University, Palo Alto, CA 94305 USA

Daphne Koller KOLLER@CS.STANFORD.EDU
Computer Science Department, Stanford University, Palo Alto, CA 94305 USA

Abstract

Max-product (max-sum) message passing algorithms are widely used for MAP inference in MRFs. It has many variants sharing a common flavor of passing "messages" over some graph-object. Recent advances revealed that its convergent versions (such as MPLP, MSD, TRW-S) can be viewed as performing block coordinate descent (BCD) in a dual objective. That is, each BCD step achieves dual-optimal w.r.t. a block of dual variables (messages), thereby decreases the dual objective monotonically. However, most existing algorithms are limited to updating blocks selected in rather restricted ways. In this paper, we show a "unified" message passing algorithm that: (a) subsumes MPLP, MSD, and TRW-S as special cases when applied to their respective choices of dual objective and blocks, and (b) is able to perform BCD under much more flexible choices of blocks (including very large blocks) as well as the dual objective itself (that arise from an arbitrary dual decomposition).

1. Introduction

MAP-MRF (finding the most probable assignments for MRFs) is one of the most important components in learning and applying structured probabilistic models. In general this problem is NP-hard (Shimony, 1994). Many different methods have been proposed to approximate or solve it under specific circumstances. A large family of these methods is based on solving a dual problem of an LP relaxation. Different duals of different LP relaxations (with different tightness) have been used. These methods are usually formulated as max-product (max-sum) message passing over the MRF or its cluster (region) graph.

Recent advances revealed that convergent versions of these algorithms can often be interpreted as block coordinate descent (BCD) in the dual (Meltzer et al., 2009; Sontag et al., 2011). However, most of them operate on local (small) blocks, such as MPLP (Globerson & Jaakkola, 2007) and its generalizations (Sontag et al., 2008), max-sum diffusion (MSD) (Werner, 2007), and TRW-S (Kolmogorov, 2006). Given a block of dual variables (messages), these algorithms work by enforcing some *consistency constraint* over the block, thereby achieve dual-optimal w.r.t. these dual variables. Meltzer et al. (2009) noted that it was difficult to generalize them to larger blocks while enforcing the same consistency constraints. Sontag et al. (2009) proposed a method (tree-BCD) that updates much larger blocks for a specific choice of dual. But it was not clear how to apply it to duals of tighter LP relaxations (e.g., a dual decomposition with cycle subproblems).

We observe that the difficulties in generalizing these methods arise from the fact that they all impose too strong consistency constraints, which are *sufficient but not necessary* for the dual objective to be optimal on the blocks being updated. By loosing these constraints, we are able to perform BCD on much larger blocks.

Indeed, we show that dual-optimality on blocks (of messages) can be established on a much broader basis—with quite general choices of the dual objective itself as well as the blocks to be updated. Specifically, we illustrate this by deriving a "unified" message passing algorithm in the most general setup—an

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28.

arbitrary dual decomposition. The resulted algorithm (*subproblem-tree calibration*, or STC) has the following properties:

1. It is formulated as message passing on a graph-object (*subproblem multi-graph*, or SMG) that generalizes traditional cluster (region) graphs.
2. It subsumes MPLP, MSD and TRW-S as special cases when applied to their respective choices of dual objectives and blocks. We will precisely characterize the choices made by these existing methods under our framework.
3. It achieves dual-optimal on blocks that can be chosen in an extremely flexible manner, including very large blocks. For example, we can choose a block that spans all subproblems in an arbitrary dual decomposition.

In other words, our framework attempts to characterize the *degrees of freedom* we have in designing a message passing algorithm (for MAP inference). Understanding these flexibilities could help us better understand existing methods, as well as design more powerful ones. In practice, we observe that we often get stuck in sub-optimal dual states when only updating blocks chosen in a restricted manner, whereas being able to choose blocks with more flexibility could lead to better dual (and primal) states.

2. Background

2.1. MAP inference, LP relaxation, and dual decomposition

The MAP inference problem over $\mathbf{X} = \{X_{1:N}\}$ and graph structure $G = \{V, E\}$ can be formulated as:

$$\underset{\mathbf{X}}{\text{maximize}}\ \boldsymbol{\Theta}(\mathbf{X}) \tag{1}$$

where $\boldsymbol{\Theta}(\mathbf{X}) = \sum_{\alpha\in\mathcal{A}} \boldsymbol{\theta}_\alpha(X_\alpha)$; $\mathcal{A}$ is the set of MRF cliques. Without loss of generality we choose $\mathcal{A} = V \cup E$ for a parametrization with unary and pairwise potentials[1]. We use lowercase $x_i \in Val(X_i)$ and $\mathbf{x} = \{x_{1:N}\}$ to denote assignments to the variables. Problem (1) is NP-hard in general (Shimony, 1994).

A large family of MAP inference methods builds on solving a linear programming (LP) relaxation (Wainwright & Jordan, 2008; Koller & Friedman, 2009):

$$\underset{\boldsymbol{\mu}\in\mathcal{M}}{\text{maximize}}\quad \boldsymbol{\Theta}\cdot\boldsymbol{\mu} \tag{2}$$

[1] A high-order potential can be converted into unary and pairwise potentials by introducing auxiliary variables.

where $\boldsymbol{\mu} = \{\mu_i(x_i), \mu_{ij}(x_i, x_j) \mid \forall i,\ x_i,\ (i,j),\ (x_i, x_j)\}$; $\boldsymbol{\Theta}$ is all MRF parameters $\{\boldsymbol{\theta}_i,\ \boldsymbol{\theta}_{ij}\}$ concatenated in same ordering as $\boldsymbol{\mu}$. Choosing different polytopes $\mathcal{M}$ results in different LP relaxations. One choice is the *marginal polytope*:

$$\begin{aligned}\mathcal{M}_G = \{\boldsymbol{\mu} \mid \exists p(\mathbf{X}), \sum_{\mathbf{x}\in Val(\mathbf{X})} p(\mathbf{x}) = 1,\ p(\mathbf{x}) \geq 0,\\ p(X_i) = \mu_i,\ p(X_i, X_j) = \mu_{ij},\ \forall i,\ (i,j)\}\end{aligned} \tag{3}$$

where $p(X_i)$ and $p(X_i, X_j)$ denote marginal distributions of $p(\mathbf{X})$. When $\mathcal{M} = \mathcal{M}_G$ the LP relaxation (2) is equivalent to the original problem (1). Therefore it is also NP-hard—the marginal polytope is defined by exponentially many faces (constraints).

A widely used choice for $\mathcal{M}$ is the *local polytope*:

$$\begin{aligned}\mathcal{M}_L =\{\boldsymbol{\mu} \mid \mu_{ij} \geq 0,\ \sum_{x_j} \mu_{ij}(X_i, x_j) = \mu_i(X_i),\\ \sum_{x_i} \mu_i(x_i) = 1,\ \forall i \in V, (i,j) \in E\}\end{aligned} \tag{4}$$

which only has a polynomial number of constraints, and the LP relaxation (2) with $\mathcal{M}_L$ is tractable. In general $\mathcal{M}_L$ is a loose outer bound of $\mathcal{M}_G$. Any solution to (2) with $\mathcal{M} = \mathcal{M}_L$ is a vertex of $\mathcal{M}_L$. If it happens to be a vertex of $\mathcal{M}_G$, we have the exact solution to (1). This is usually not the case except for several special families such as the problems with tree structures or sub-modular potentials etc.

Solving (2) using a standard LP solver is very inefficient (Yanover et al., 2006) even with a tractable choice of $\mathcal{M}$. In practice we usually solve its dual LP. Indeed, it is more convenient to start with a *dual decomposition*, which directly gives us a dual LP (of some underlying primal LP that we do not explicitly deal with).

Specifically, consider a decomposition of $\boldsymbol{\Theta}(\mathbf{X})$ into subproblems $c \in \mathcal{C}$, parametrized by $\{\boldsymbol{\Theta}^c\}$, which altogether give rise to a reparametrization of $\boldsymbol{\Theta}(\mathbf{x})$:

$$\forall \mathbf{x},\quad \sum_{c\in\mathcal{C}} \boldsymbol{\Theta}^c(\mathbf{x}|_c) = \boldsymbol{\Theta}(\mathbf{x}) \tag{5}$$

where $\mathbf{x}|_c$ denotes restricting the joint assignment to the scope of subproblem c. The subproblems could be single nodes, edges, trees, or cycles that are assumed to be tractable.

Note that (5) has exponentially many constraints. A succinct way to enforce them is to express the reparametrization in terms of messages:

$$\boldsymbol{\Theta}^c = \boldsymbol{\Theta}^c_0 + \sum_{c':\, \mathbf{X}_c \cap \mathbf{X}'_c \neq \emptyset} \boldsymbol{\delta}_{c'\to c}(\mathbf{X}_c \cap \mathbf{X}'_c) \tag{6}$$

where the messages satisfy $\boldsymbol{\delta}_{c'\to c} = -\boldsymbol{\delta}_{c\to c'}$. The initial subproblem potentials $\boldsymbol{\Theta}_0^c$ are assumed to satisfy (5). They can be constructed, for example, by splitting the original potentials. It is easy to check that subproblem potentials defined by (6) satisfy (5) for any choice of the messages.

Note that the scopes of messages are defined by $\mathbf{X}_c \cap \mathbf{X}'_c$, which is the intersections among $\mathbf{X}_c$ (with a subscript): the original *MRF variables* covered by subproblem c. This is to distinguish from $\mathbf{X}^c$ (with a superscript), which are *subproblem variables*. For example, if subproblems c and c' both cover X_1 (orginal MRF variable), we would have X_1^c and $X_1^{c'}$: two different variables with independent assignments.

Since each subproblem has its own copy of variables $\mathbf{X}^c$, summing over their optimal values gives rise to an upper bound of (1) because each subproblem is maximized independently. This upper bound is a function of the messages:

$$\mathcal{D}(\{\boldsymbol{\delta}_{c'\to c}\}) = \sum_{c\in\mathcal{C}} \max_{\mathbf{X}^c} \boldsymbol{\Theta}^c(\mathbf{X}^c). \tag{7}$$

It turns out that (7) is a dual problem of (2) under some choice of $\mathcal{M}$. The tightness of the underlying $\mathcal{M}$ depends on our choice of the subproblems. It has been shown (Komodakis et al., 2011) that choosing only tree structured subproblems leads to a dual of (2) with $\mathcal{M} = \mathcal{M}_L$. Duals of tightened LP relaxations can be constructed by introducing complex subproblems such as cycles.

Given a decomposition, our goal is to minimize (7) by rearranging subproblem potentials (via message passing), thereby solving the underlying LP relaxation.

2.2. Bethe cluster (region) graph

In principle the messages $\boldsymbol{\delta}_{c'\to c}$ in (6) can be defined between all pairs of overlapping subproblems. However, a rather restricted form has been dominantly used in defining such messages: the Bethe cluster graph. It has a bipartite structure with one layer of "factor" nodes and one layer of small (usually unary) nodes encoding intersections between the factors. Specifically, consider using the set of subproblem $\mathcal{C} = \{i\}\cup\mathcal{F}$ consisting of unary subproblems $\{i\}$ and larger subproblems (factors) $\mathcal{F}$. So the dual (7) takes a restricted form:

$$\mathcal{D}(\{\boldsymbol{\delta}_{f\to i}\}) = \sum_i \max_{X^i} \boldsymbol{\theta}^i(X^i) + \sum_f \max_{\mathbf{X}^f} \boldsymbol{\Theta}^f(\mathbf{X}^f) \tag{8}$$

where the messages are only defined *between* the two layers because of the bipartite structure. This construction is shown in Fig. 1 (a) and (c). Note that we used superscripts (instead of subscripts) in $\boldsymbol{\theta}^i$ and X^i to indicate that these are subproblem potentials and variables to distinguish from the orginal MRF potentials $\boldsymbol{\theta}_i$ and variables X_i.

It has been shown (Meltzer et al., 2009; Sontag & Jaakkola, 2009; Sontag et al., 2011) that many existing algorithms, including (Globerson & Jaakkola, 2007; Sontag et al., 2008; 2011; Werner, 2007; Sontag & Jaakkola, 2009; Tarlow et al., 2011; Komodakis & Paragios, 2008; Meltzer et al., 2009), can be interpreted as passing messages on the Bethe cluster graph and thereby optimizing the dual objective (8). This "restricted" design had arisen from the historical concern of satisfying the *running intersection property*, thus alleviating double-counting in loopy BP (Weiss, 2000). However, this is no longer relevant under the modern view of message passing as a BCD algorithm. To this end, we will introduce a more general graph-object: *subproblem multi-graph* (Sec. 3.1) to serve the role of traditional cluster graphs. This more general setup allows us to pass messages in more flexible ways and achieve better dual (and primal) states in many situations (as shown in experiments).

3. The STC Framework

3.1. Subproblem multi-graph and subproblem trees

For notation simplicity, in this paper we treat graphs $\mathcal{G}$ and trees $\mathcal{T}$ as *sets* consisting of nodes and edges. So we will use $\mathcal{T}\subset\mathcal{G}$ to denote that $\mathcal{T}$ is a subgraph of $\mathcal{G}$. And we will use $e\in\mathcal{T}$ or $c\in\mathcal{T}$ to denote that edge e or node c is in the tree.

Given a dual decomposition with subproblems $\mathcal{C}$, we build a graph-object as follows:

Definition 1 (Subproblem Multi-Graph/Tree)**.** *Given $\mathcal{C}$, the* **subproblem multi-graph (SMG)** *$\mathcal{G} = (\mathcal{V}, \mathcal{E})$ has one node for each $c\in\mathcal{C}$, and one edge between c and c' for each tuple (c, c', φ), where $\varphi\in V\cup E$ is shared by c and c'. A* **subproblem tree** *is a tree $\mathcal{T}\subset\mathcal{G}$.*

Note that we use (V, E) for the MRF graph and $(\mathcal{V}, \mathcal{E})$ for SMG.

This construction is illustrated in Fig. 1 (b) where we decompose the MRF (Fig. 1 (a)) into four subproblems.

Note that if we include all unary subproblems into the decomposition, we would get a SMG similar to Fig. 1 (c) but with extra edges among the non-unary

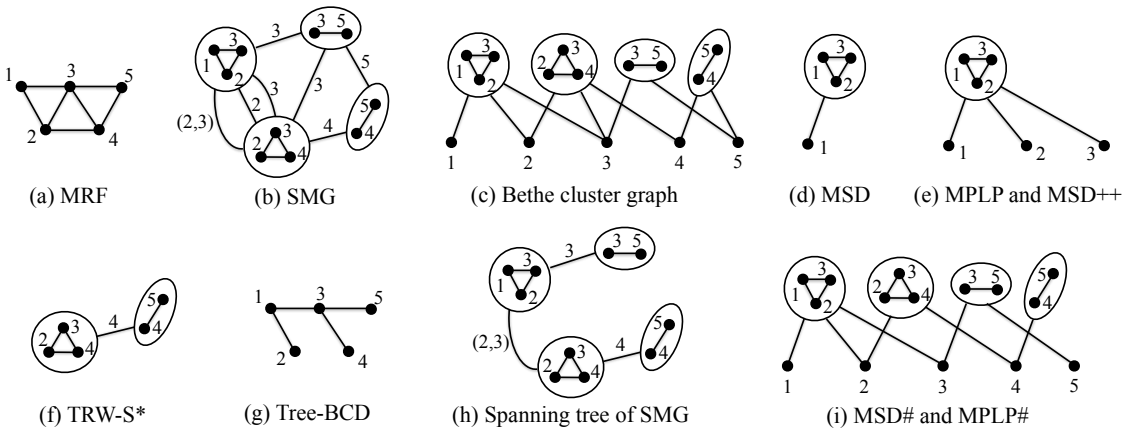

Figure 1. Examples of subproblem multi-graph and subproblem trees. See text for explanation.

subproblems. So a tree in the Bethe cluster graph (which we call a *Bethe tree*) is also a *subproblem tree* by definition. Therefore all conclusions in this paper on subproblem trees apply to Bethe trees.

Fig. 1 (d) to (i) are examples of subproblem trees. They correspond to blocks updated by different algorithms as explained later. Note that (d) (e) and (i) are Bethe trees corresponding to the Bethe cluster graph (c); (f) and (h) are trees in the SMG (b); (g) is a Bethe tree of a different dual decomposition (with all edges and nodes as subproblems).

For each SMG edge $(c, c', \varphi) \in \mathcal{E}$, we have messages[2] $\boldsymbol{\delta}_{c' \to c}(X_\varphi) = -\boldsymbol{\delta}_{c \to c'}(X_\varphi)$. Therefore the block (of dual variables) associated with subproblem tree $\mathcal{T}$ is given by

$$\mathcal{B}^{\mathcal{T}} = \{\boldsymbol{\delta}_{c' \to c}(X_\varphi) : (c, c', \varphi) \in \mathcal{T}\}. \tag{9}$$

3.2. Max-consistency and dual-optimal on trees

Given a block $\mathcal{B}^{\mathcal{T}}$ associated with some subproblem tree $\mathcal{T}$, we want to achieve dual-optimal w.r.t. that block:

Definition 2 (Dual-optimal on $\mathcal{T}$)**.** *The subproblem potentials $\{\boldsymbol{\Theta}^c\}$ are dual-optimal on $\mathcal{T}$ if we can not further decrease the dual objective by changing messages in $\mathcal{B}^{\mathcal{T}}$.*

A message passing algorithm achieves this by enforcing some *consistency constraint*. We first identify a constraint that is equivalent to dual-optimal on $\mathcal{T}$.

[2]The messages defined in (6) appear to have larger scopes. However, if the MRF is parametrized over $\mathcal{A} = V \cup E$, we can always reparametrize the messages in (6) using messages over unary or pairwise scopes. For high-order MRFs we could allow φ to be larger scopes as well.

Definition 3 (Assignments agree on $\mathcal{T}$)**.** *Assignments to all subproblems $\{\mathbf{x}^c\}_{c \in \mathcal{T}}$ agree on $\mathcal{T}$, denoted as $\{\mathbf{x}^c\} \sim \mathcal{T}$, if for $\forall (c, c', \varphi) \in \mathcal{T}$ we have $x^c_\varphi = x^{c'}_\varphi$.*

Definition 4 (Weak max-consistency on $\mathcal{T}$)**.** *$\{\boldsymbol{\Theta}^c\}_{c \in \mathcal{T}}$ satisfies weak max-consistency if*

$$\sum_{c \in \mathcal{T}} \max_{\mathbf{X}^c} \boldsymbol{\Theta}^c(\mathbf{X}^c) = \max_{\{\mathbf{X}^c\} \sim \mathcal{T}} \sum_{c \in \mathcal{T}} \boldsymbol{\Theta}^c(\mathbf{X}^c) \tag{10}$$

That is, maximizing each subproblem independently gets to the same optimal value as maximizing them while requiring the assignments to agree on the tree. This condition turns out to be equivalent to dual optimal on $\mathcal{T}$ (Proposition 1). We will show that our message passing algorithm (Sec. 3.3) enforces this constraint for arbitrary $\mathcal{T}$.

We now identify two other stronger constraints (enforced by existing algorithms) that are *sufficient but not necessary* for dual-optimal on $\mathcal{T}$.

Let $\boldsymbol{M}^c_\varphi$ be the (log)-max-marginals of c on φ:

$$\boldsymbol{M}^c_\varphi(x_\varphi) = \max_{\mathbf{X}^c | \varphi = x_\varphi} \boldsymbol{\Theta}^c(\mathbf{X}^c), \tag{11}$$

Note that if $\varphi = (i, j) \in E$, $\mathbf{X}^c|_\varphi = x_\varphi$ means $X^c_i = x_i$ and $X^c_j = x_j$.

The following consistency constraint requires the subproblem to agree on their max-marginals over the tree:

Definition 5 (Strong max-consistency[3] on $\mathcal{T}$)**.** *$\{\boldsymbol{\Theta}^c\}_{c \in \mathcal{T}}$ satisfies strong max-consistency if $\boldsymbol{M}^c_\varphi = \boldsymbol{M}^{c'}_\varphi$, $\forall (c, c', \varphi) \in \mathcal{T}$.*

Another consistency constraint requires that the

[3]This corresponds to "max-consistency" in (Meltzer et al., 2009).

subproblem potentials match the max-marginals of the tree distribution:

Definition 6 (MPLP max-consistency on $\mathcal{T}$)**.** *For Bethe tree $\mathcal{T}$ with N unary clusters, $\{\boldsymbol{\theta}^i, \boldsymbol{\Theta}^f\}_{i,f\in\mathcal{T}}$ satisfies MPLP max-consistency if $\boldsymbol{\theta}^i = \frac{1}{N}\boldsymbol{\mu}_i^{\mathcal{T}}$ and $\boldsymbol{\Theta}^f = \boldsymbol{\mu}_f^{\mathcal{T}} - \sum_{i\in f,\, i\in\mathcal{T}} \frac{N_i^f}{N}\boldsymbol{\mu}_i^{\mathcal{T}}$ where $\boldsymbol{\mu}_i^{\mathcal{T}}$ and $\boldsymbol{\mu}_f^{\mathcal{T}}$ are max-marginals[4] of the tree $\mathcal{T}$, and N_i^f is the number of unary clusters in the subtree rooted by i on the opposite side of f.*

Note that MPLP max-consistency is only defined for Bethe trees. With a slight abuse of notation, we used $i \in f$ in the above definition to denote that the scope of unary subproblem i is contained in the scope of factor subproblem f in a Bethe cluster graph.

The relations among these consistency constraints are given below.

Proposition 1. *For any Bethe tree $\mathcal{T}$,*

$$\textit{MPLP max-consistency} \implies \textit{weak max-consistency}$$

For any subproblem tree $\mathcal{T}$ (including Bethe trees),

$$\textit{strong max-consistency} \implies \textit{weak max-consistency}.$$
$$\textit{weak max-consistency} \iff \textit{dual-optimal on } \mathcal{T}.$$

Proof for this (**as well as all other propositions in the rest of this paper**) are given in Appendix (supplement material), where we also give an example that satisfies weak max-consistency but not the other two.

The two stronger constraints are enforced by existing methods:

Proposition 2. *Max-sum diffusion (MSD) (Werner, 2007) performs BCD on the Bethe dual (8). Each BCD step enforces strong max-consistency for a Bethe tree consisting of one $f \in \mathcal{F}$ and one $i \in f$ (as in Fig. 1 (d)).*

Proposition 3. *TRW-S (Kolmogorov, 2006) performs BCD on the dual (7) where each subproblem $c \in \mathcal{C}$ is a tree (of the MRF). Each BCD step enforces strong max-consistency for a subproblem tree $\mathcal{T}$ with nodes $\mathcal{V}_{\mathcal{T}} = \{c : \text{scope}(c) \supset \varphi\}$ for a given $\varphi \in V \cup E$, and edges (c, c', φ) that constitute a tree over $\mathcal{V}_{\mathcal{T}}$*[5]*.*

Proposition 4. *MPLP (Globerson & Jaakkola, 2007; Sontag et al., 2011) performs BCD on the Bethe dual (8). Each BCD step enforces MPLP max-consistency for a Bethe tree consisting of one $f \in \mathcal{F}$ and all $i \in f$ (as in Fig. 1 (e)).*

Proposition 5. *Tree-BCD (Sontag & Jaakkola, 2009) performs BCD on the Bethe dual (8) where all f are pairwise factors (i, j). Each BCD step enforces MPLP max-consistency for a Bethe tree corresponding to a spanning tree of the MRF (as in Fig. 1 (g)).*

Note that a spanning tree of the MRF (with $|V| - 1$ edges) is much smaller than a spanning tree of the SMG (of a dual decomposition into all edges and nodes). The latter contains all (possibly $O(|V|^2)$) edges of the MRF.

3.3. The STC algorithm

Now we show how to attain weak max-consistency for any subproblem tree (such as Fig. 1 (h) and (i)). To express the algorithm concisely, we use $x \xrightarrow{m} y$ to denote two updates: $x \leftarrow x - m$ and $y \leftarrow y + m$. We assume subproblem solvers for all c that output $\boldsymbol{M}_\varphi^c$. For tree-structured subproblems this is straightforward. For cycle subproblems we use (Felzenszwalb & McAuley, 2011) which provides a fast way of computing the junction-tree messages.

Our algorithm (Alg. 1) calibrates a subproblem-tree by an upstream pass and a downstream pass; both update subproblem potentials "in place" without storing any message (although conceptually it can be viewed as updating the messages in (7) as well). The

Algorithm 1 Subproblem tree calibration (STC)

1: INPUT: $\mathcal{T}$; allocation weight a_c for each $c \in \mathcal{T}$ satisfying $a_c \geq 0$, $\sum_{c\in\mathcal{T}} a_c = 1$.
2: / * upstream pass * /
3: **for** edges (c, c', φ) in upstream order **do**
4: $\quad \boldsymbol{\Theta}^c \xrightarrow{\boldsymbol{M}_\varphi^c} \boldsymbol{\Theta}^{c'}$, where c is a child of c'
5: / * downstream pass * /
6: $w_c \leftarrow \sum_{\tilde{c}\in\mathcal{T}_c} a_{\tilde{c}}$ for all $c \in \mathcal{T}$, where $\mathcal{T}_c$ is the subtree rooted by c
7: **for** nodes c' in downstream order **do**
8: $\quad$ (P-STC only) precompute all $\boldsymbol{M}_\varphi^{c'}$ used below
9: $\quad$ **for** all c (children of c') **do**
10: $\quad\quad \boldsymbol{\Theta}^{c'} \xrightarrow{(w_c/w_{c'})\cdot\boldsymbol{M}_\varphi^{c'}} \boldsymbol{\Theta}^c$
11: $\quad\quad$ (S-STC only) $w_{c'} \leftarrow w_{c'} - w_c$

downstream pass of STC can be performed for each children of a node either *sequentially* or in *parallel*. They differ in line 8 and 11 of Alg. 1. We call these two alternatives S-STC and P-STC, respectively. Any statement about STC in the following, if not specified, applies to both P-STC and S-STC.

Overall we repeatedly choose different trees and

[4] Max-marginals of subproblem tree $\mathcal{T}$ over $\tilde{c}$ is defined as $\boldsymbol{\mu}_{\tilde{c}}^{\mathcal{T}}(\boldsymbol{x}_{\tilde{c}}) = \max_{\{\boldsymbol{X}^c\}\sim\mathcal{T},\, \boldsymbol{X}^{\tilde{c}}=\boldsymbol{x}_{\tilde{c}}} \sum_{c\in\mathcal{T}} \boldsymbol{\Theta}^c(\boldsymbol{X}^c)$.

[5] In TRW-S the subproblems (MRF trees) can be connected in arbitrary ways into $\mathcal{T}$. Because all edges in $\mathcal{T}$ have the same scope φ, strong max-consistency essentially requires the max-marginals on φ to be the same for all $c \in \mathcal{T}$.

perform the calibration.

Proposition 6. *STC enforces weak max-consistency for any $\mathcal{T}$ and any allocation weights a_c satisfying $a_c \geq 0$, $\sum_{c\in\mathcal{T}} a_c = 1$.*

Note that this (together with Proposition 1) implies monotonicity and convergence of the overall algorithm.

We now clarify the relations between STC and existing methods.

Proposition 7. *Applying P-STC to the Bethe tree specified in Proposition 4 with allocation weights $a_i = \frac{1}{|f|}$, $a_f = 0$ is equivalent to MPLP.*

Proposition 8. *Applying STC to the Bethe tree specified in Proposition 2 with allocation weights $a_f = 0.5$, $a_i = 0.5$ is equivalent to MSD.*

Proposition 9. *Applying STC to the subproblem tree specified in Proposition 3 with uniform allocation weights $a_c = \frac{1}{|\mathcal{T}|}$ is equivalent to TRW-S.*

This implies that STC actually enforces strong max-consistency when applied to settings of MSD and TRW-S. Indeed, we have:

Proposition 10. *If all edges of $\mathcal{T}$ have same scope φ, P-STC is equivalent to S-STC. Both achieve strong max-consistency when applied with uniform allocation weights $a_c = \frac{1}{|\mathcal{T}|}$*

This condition (all edges of $\mathcal{T}$ have the same scope) is satisfied in MSD and TRW-S.

Given Proposition 9, we can easily generalize TRW-S beyond tree subproblems[6]. We call this generalized algorithm TRW-S*: applying STC to a subproblem tree consists of all subproblems sharing a given MRF node or edge, as in Fig. 1 (f). It will appear in experiments.

3.4. Fixed-point characterization

In this part we give a fixed-point characterization of the STC algorithm, and we show that it is in fact equivalent to the weak tree agreement (WTA) condition (Kolmogorov, 2006). Indeed, the fact that TRW-S (Kolmogorov, 2006) is a special case of STC suggests that the latter should be at least as powerful as the former.

Definition 7 (WTA2)**.** *Subproblem potentials $\{\boldsymbol{\Theta}^c\}$ satisfy weak tree-agreement-two (WTA2) if for any subproblem tree $\mathcal{T}$, $\exists\{\mathbf{x}^c\} \sim \mathcal{T}$ that are optimal for each $c \in \mathcal{T}$ individually.*

[6]However, the acceleration trick by monotonic chains is not applicable in general.

Proposition 11 (STC Fixed-Point)**.** *If $\{\boldsymbol{\Theta}^c\}$ do not satisfy WTA2, we can always find $\mathcal{T}$ such that applying STC to $\mathcal{T}$ decreases the dual objective. If $\{\boldsymbol{\Theta}^c\}$ satisfy WTA2, applying STC to any $\mathcal{T}$ does not change the dual objective and preserves the WTA2 condition.*

We re-state the WTA condition from (Kolmogorov, 2006) in a slightly generalized form (allowing c to be arbitrary subproblems, not restricted to trees).

Definition 8 (WTA)**.** *Subproblem potentials $\{\boldsymbol{\Theta}^c\}$ satisfy weak tree-agreement (WTA) if we can find a* **subset** *of optimal assignments for each c, denoted as $OPT(c)$, such that: for any small subproblem tree $\mathcal{T}$ consists of two nodes c and c' connected by (c, c', φ), we have: for $\forall \mathbf{x}^c \in OPT(c)$, $\exists \mathbf{x}^{c'} \in OPT(c')$, $\{\mathbf{x}^c, \mathbf{x}^{c'}\} \sim \mathcal{T}$.*

Comparing to WTA, WTA2 appears to assert a weaker constraint on more general $\mathcal{T}$. This originates from the fact that STC calibrates more general $\mathcal{T}$ and enforces a weaker consistency constraint than that of MSD and TRW-S. However asymptotically they are equally powerful:

Proposition 12. *WTA2 is equivalent to WTA.*

3.5. Choosing allocation weights

In this part we clarify the role of allocation weights a_c, which are responsible for the message coefficients in the downstream pass of STC. As we have seen in Proposition 8 and 7, there are subtle (but important) differences between MPLP and MSD in this aspect. (This has also been noted in (Sontag et al., 2011).)

In order to understand the role of allocation weights a_c, we first show some detailed characterization of STC.

Proposition 13 (STC Allocation)**.** *After STC, for each subproblem c we have:*

$$\max_{\mathbf{X}^c} \boldsymbol{\Theta}^c(\mathbf{X}^c) = a_c \cdot \max_{\{\mathbf{X}^{\tilde{c}}\}\sim\mathcal{T}} \sum_{\tilde{c}\in\mathcal{T}} \boldsymbol{\Theta}^{\tilde{c}}(\mathbf{X}^{\tilde{c}}) \tag{12}$$

Intuitively, the downstream pass allocate "energy" to all subproblems according to their allocation weights. We can further show that:

Proposition 14 (Detailed Monotonicity)**.** *Let $\mathcal{D}_0$, $\mathcal{D}_1$, $\mathcal{D}_2$ be the dual objective value before STC, after upstream pass, and after downstream pass, respectively. We have $\mathcal{D}_0 \geq \mathcal{D}_1 = \mathcal{D}_2$. Moreover, the dual objective remains constant in each single step of the downstream pass.*

That is, the downstream pass essentially moves around in a plateau of the dual, preparing for the next

downhill move. Therefore the allocation weights determine where to settle on that plateau.

When $\mathcal{T}$ is a Bethe tree, two styles of allocation have been used. One is the MPLP-style allocation, which assigns zero weights to all non-unary subproblems, and uniformly allocate among unary subproblems. This is the case for MPLP and Tree-BCD[7]. The other is the MSD-style allocation, which assigns uniform allocation weights to all subproblems. This is the case for MSD and its generalized version MSD++ (Sontag et al., 2011).

Given our framework, it is straightforward to further generalize these two styles of allocation to spanning trees of the Bethe cluster graph (as in Fig. 1 (i)). We call these two algorithms MPLP# and MSD#, respectively[8]. They will appear in experiments.

3.6. Generate primal solutions

Given subproblem potentials, solutions to the original MAP inference problem can be constructed in different ways (Komodakis et al., 2011). For example, when unary subproblems are present, we could simply take the assignments that minimize each unary subproblem. Or we could visit all subproblems in turn, and for each one commit the variables in its scope to the subproblem solution.

In order to better leverage "beliefs" of different subproblems as well as "smoothness" across subproblems, we propose the following method, which works better in practice comparing to other heuristics. (The comparison is not shown due to space limit.) In practice, one could construct multiple assignments from different heuristics and choose the one with the best objective value.

We visit the variables (in the original MRF) in some ordering, for example, $X_1, X_2, \cdots X_N$. And for X_i we choose the assignment:

$$x_i = \text{argmax} \sum_{c: i \in scope(c)} \max_{\boldsymbol{X}^c \setminus X_i} \boldsymbol{\Theta}^c(\boldsymbol{X}^c | X_j = x_j, \forall j < i) \tag{13}$$

That is, when visiting each X_i, we choose its assignment to maximize the sum of all max-marginals from all subproblems covering X_i. And then we fix $X_i = x_i$ in all subproblems (this will affect subsequent visits to these subproblems).

[7] Although Tree-BCD is not a special case of STC. We can define its choice of allocation weights according to the distribution of "energy" in its calibrated tree.

[8] Specifically, MPLP# means applying STC with MPLP-style allocation weights to randomly selected spanning trees as in Fig. 1 (i), and similarily for MSD#.

4. Experimental Results

[Experiment Setup] We used three MAP inference tasks in experiments: (1) The protein design benchmark (Yanover et al., 2006). We use the 20 largest problems from that dataset, with number of variables from 101 to 180, number of edges from 1,973 to 3,005, variable cardinality up to 154. (2) Synthetic 20-by-20 grid with variable cardinality of 100. Potentials are dawn from $\mathcal{N}(0, 1)$. (3) The "object detection" task from PIC-2011[9]. There are 37 problem instances, each has 60 variables and 1,770 edges, variable cardinality from 11 to 21. Each task corresponds to one row in Fig. 2.

For each tasks, we use two settings: with or without cycle subproblems (loose or tightened LP relaxation). The former corresponds to the left two columns in Fig. 2, and the latter corresponds to the right two columns. Each setting for each task is shown by a primal(right)-dual(left) pair of figures (averaged over all problem instances). Each primal-dual pair share one legend.

The cycles in dual decomposition are selected in a static manner by applying the criterion of (Sontag et al., 2008) to the original edge potentials. We did this (instead of dynamically adding cycles) because our goal is to compare different dual methods and we want them to always operate on the same dual problem. For protein design and PIC, we selected 500 triangles for each problem instance. For grid, we used all 381 squares of size four.

[Methods Compared] Among the methods compared, MPLP (Globerson & Jaakkola, 2007), MSD++ (Sontag et al., 2011), Tree-BCD (Sontag & Jaakkola, 2009), TRW-S MonoChain (Kolmogorov, 2006) are existing methods. TRW-S* is a straightforward generalization of a existing method. MPLP#, MSD#, and STC are new methods derived from our framework. Here STC means calibrating randomly chosen spanning trees of the SMG without unary subproblems (as in Fig. 1 (h)). For TRW-S MonoChain we used the code of (Szeliski et al., 2008). Note that it is only applicable to the grid problem with long monotonic chains. For TRW-S* we use MRF edges (and cycles for the tightened setting) as subproblems. All other methods have been explained in earlier sections.

[Result Analysis] When cycle subproblems are present, we observe that STC and TRW-S* performs significantly better in all tasks. The crucial difference

[9] http://www.cs.huji.ac.il/project/PASCAL/index.php

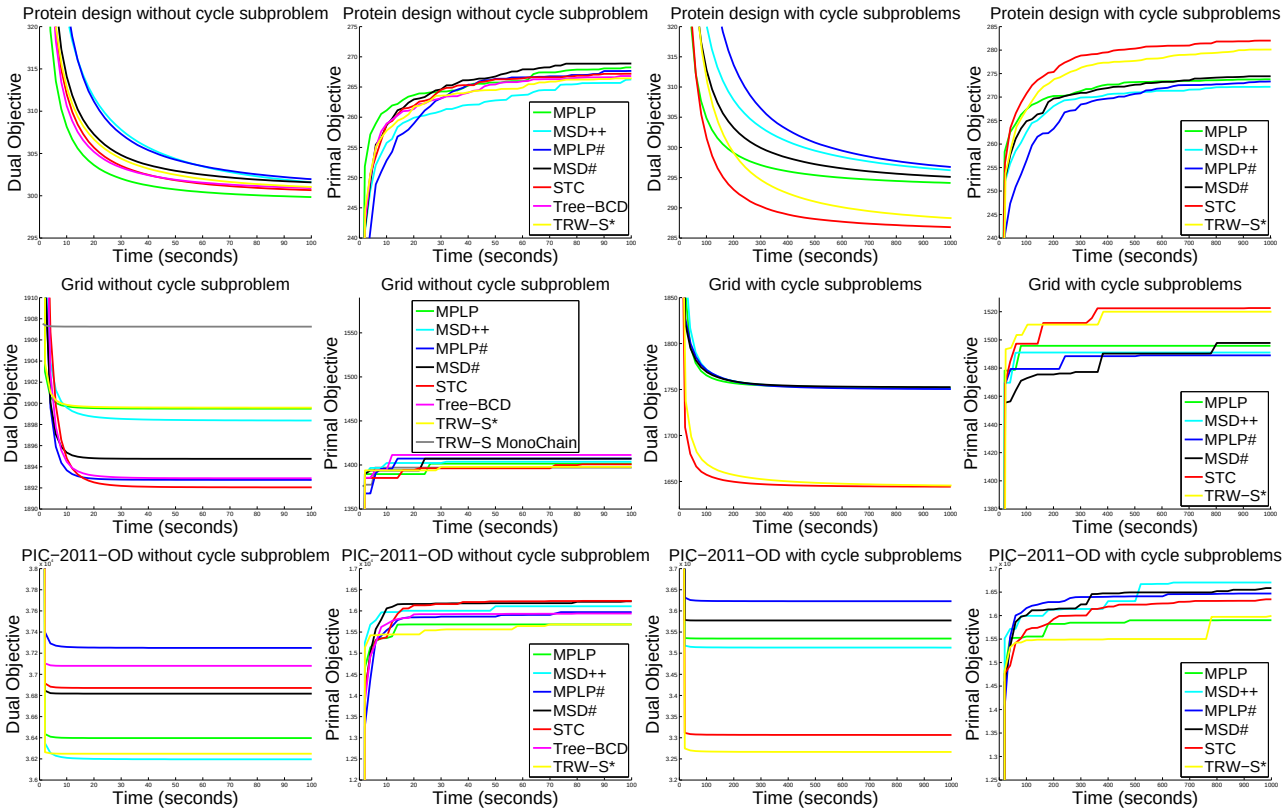

Figure 2. Experimental results. This figure is better viewed in color.

is that: these two methods allow the subproblems communicating through MRF edges, whereas all other methods are restricted to passing messages over MRF nodes (they use the Bethe cluster graph as in Fig. 1 (c)). This reveals the limitation of this bipartite construction.

For sparse MRFs (the grid case), getting more subproblems involved in each BCD steps turns out to be important. As we can see from the second plot in the first column of Fig. 2. The methods that updates "global" blocks (STC, Tree-BCD, MPLP#, MSD#) performs significantly better than the ones with "local" blocks (MPLP, MSD++, and TRW-S*). Note that TRW-S MonoChain get stuck at an even worse dual state than the "local" methods. This is because its block selection (and ordering) is even more restrictive due to using monotonic chains.

Overall, we observe that different methods tend to "converge" to different dual objectives, Even though the dual objectives in each plot should have exactly the same optimal value. Note that all these methods perform BCD—they achieve dual optimal on a block in each step. Therefore choosing blocks (as well as choosing plateau dual states) is very important to the performance (and final result) of a message passing algorithm. Generally speaking, being able to choose blocks and plateau states in a more flexible manner (as our framework reveals) could help us get to better dual states, whereas making these choices in a restricted manner (as most existing methods do) could easily get stuck.

5. Conclusion

Our framework revealed two dimensions of flexibility in designing a message passing algorithm for MAP inference: choosing blocks to update, and choosing a dual state on a plateau in each BCD step. The STC algorithm can be applied with extreme flexibility in these choices. Although these choices appear to be important to performance, any known fixed strategy in making them does not seem to be optimal across different scenarios. If we could find principled and adaptive strategies in making these choices, we will be able to design much more powerful message passing algorithms.

Acknowledgement

This work has been supported by the Max Planck Center for Visual Computing and Communication.

References

Felzenszwalb, P. F. and McAuley, J. J. Fast inference with min-sum matrix product. *PAMI*, 33(12):2549–2554, 2011.

Globerson, Amir and Jaakkola, Tommi. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *NIPS*, 2007.

Koller, D. and Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

Kolmogorov, V. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10): 1568–1583, October 2006.

Komodakis, N. and Paragios, N. Beyond loose LP-relaxations: Optimizing MRFs by repairing cycles. In *ECCV*, pp. III: 806–820, 2008.

Komodakis, Nikos, Paragios, Nikos, and Tziritas, Georgios. MRF energy minimization and beyond via dual decomposition. *PAMI*, 33(3):531–552, 2011.

Meltzer, Talya, Globerson, Amir, and Weiss, Yair. Convergent message passing algorithms - a unifying view. In *UAI*, pp. 393–401, 2009.

Shimony. Finding MAPs for belief networks is NP-hard. *AIJ: Artificial Intelligence*, 68, 1994.

Sontag, David and Jaakkola, Tommi. Tree block coordinate descent for MAP in graphical models. *AISTATS*, 5:544–551, 2009.

Sontag, David, Meltzer, Talya, Globerson, Amir, Jaakkola, Tommi, and Weiss, Yair. Tightening LP relaxations for MAP using message passing. In *UAI*, pp. 503–510, 2008.

Sontag, David, Globerson, Amir, and Jaakkola, Tommi. Introduction to dual decomposition for inference. In Sra, Suvrit, Nowozin, Sebastian, and Wright, Stephen J. (eds.), *Optimization for Machine Learning*. MIT Press, 2011.

Szeliski, R. S., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(6):1068–1080, June 2008.

Tarlow, Daniel, Batra, Dhruv, Kohli, Pushmeet, and Kolmogorov, Vladimir. Dynamic tree block coordinate ascent. In *ICML*, 2011.

Wainwright, Martin J. and Jordan, Michael I. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

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

Werner, T. A linear programming approach to max-sum problem: A review. *PAMI*, 29(7):1165–1179, July 2007.

Yanover, Chen, Meltzer, Talya, and Weiss, Yair. Linear programming relaxations and belief propagation - an empirical study. *JMLR*, 7:1887–1907, 2006.