Neurocomputing 120 (2013) 453–460

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

A new criterion for choosing planar subproblems in MAP-MRF inference

Jianwu Dong [a], Feng Chen [a,*], Qiang Cheng [a], Song Wang [b]

[a] Tsinghua National Laboratory for Information Science and Technology, Department of Automation, Tsinghua University, Beijing 100084, China
[b] Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208, United States

ARTICLE INFO

Article history:
Received 2 July 2012
Received in revised form
2 April 2013
Accepted 15 April 2013
Communicated by X. He
Available online 10 June 2013

Keywords:
Markov random fields
MAP inference
Dual decomposition
LP relaxation
Planar subproblems

ABSTRACT

We propose an efficient algorithm for finding the maximum a posteriori (MAP) configuration in Markov random fields (MRFs) under the framework of dual decomposition. In the framework, tractable subproblems like binary planar subproblems (BPSPs) have been introduced to obtain more accurate solutions than that of tree-structured subproblems. However, since there are exponentially many BPSPs and they have very different effects on tightening the linear programming (LP) relaxation, the choice of the best BPSPs becomes an important open problem. In this paper, we find that cycles of BPSPs have the equivalent potential structure with the cycles where k-ary cycle inequalities are defined. We further prove that adding a BPSP in the dual decomposition is equivalent to enforcing a set of k-ary cycle inequalities in the LP relaxation, which gives a new insight to the procedure of adding BPSPs. In addition, a new criterion for choosing BPSPs is proposed by first selecting the violated k-ary cycle inequalities and then packing as many of these violated cycles as possible into a BPSP. Experimental results show the effectiveness of our criterion.

1. Introduction

Markov random fields are useful tools in modeling complex dependencies among random variables and are widely applied in the fields of computer vision, natural language processing, bioinformatics, machine learning, and so on [1,2]. Maximum a posteriori (MAP) inference is an important problem in both theory and applications of MRFs.

MAP inference is essentially an integer programming problem, and it is computationally infeasible if we directly solve it. A common approach is to convert the integer programming to a linear programming over the constraints of *marginal polytope* [3]. Since too many constraints are needed to describe the *marginal polytope*, researchers turn to relax the *marginal polytope* to the *local consistency polytope*, which only enforces pairwise consistency between edge marginals and node marginals (called pairwise LP relaxation). The technique of dual decomposition has been widely used for solving the LP relaxations. In dual decomposition, the problem is decomposed into a set of tractable subproblems and a global solution in then achieved by combining the solutions of subproblems [4,5]. Specifically, different tree-decomposition techniques have been developed for solving the pairwise LP relaxation and lead to many different MAP inference algorithms, such as tree-reweighted message passing (TRW) [6,7], max-sum diffusion (MSD) [8] and max product linear programming (MPLP) [9]. These algorithms are not always guaranteed to get the exact solution since the pairwise LP relaxation may not be sufficiently tight.

In order to obtain more accurate solutions, higher order constraints or more complex subproblems are introduced [4]. Sontag and Jaakkola [10] propose a more accurate inference algorithm by successively adding cycle inequality constraints to the *local consistency polytope*. They use a separation algorithm to iteratively find the most violated k-ary cycle inequality and then add it into the current constraint set. They use standard optimization solvers like the interior point method to solve the primal problem with the additional constraints. However, Yanover et al. [11] show that LP solvers cannot be applied to solve the large-scale MAP inference problems since they do not consider the special structure of the MRFs.

Batra et al. [12] give a new tighter approximate algorithm by decomposing a binary MRF into a set of outer-planar graphs, covering every node and edge of the original graph, and accurately solve each subproblem by finding minimum weight perfect matchings [13,14]. For the non-binary case, they use other approximate methods to solve the subproblems. They select the outer-planar subgraphs to cover every node and edge of the original graph, without considering the choice of extra outer-planar subgraphs to further tighten the LP relaxations. Yarkony et al. [15] propose a novel algorithm by decomposing a non-binary MRF into a "covering tree" and successively adding tractable binary planar subproblems (BPSP) based on current MAP solutions. However, they do not consider the problem of efficiently choosing BPSPs so as to tighten the LP relaxation as much as possible.

* Corresponding author. Tel: +86 10 62797145; fax: +86 10 62786911.
E-mail address: chenfeng@mail.tsinghua.edu.cn (F. Chen).

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.04.043

In this paper, we focus on the problem of choosing the best BPSPs to tighten the LP relaxation as much as possible in each step. We find the equivalence in potential structures between the cycles of BPSPs and the cycles of k-ary cycle inequalities. We then prove that adding a BPSP corresponds to enforcing a group of k-ary cycle inequalities. Based on this equivalence, a criterion for choosing BPSPs is proposed by selecting the violated k-ary cycle inequalities and then packing as many of these violated cycles as possible. We use the proposed criterion to choose BPSP, so as to ensure that adding such a BPSP in each time will repair many *violated* k-ary cycle inequalities. Our contributions are threefold. First, we prove the equivalence between adding BPSPs and enforcing k-ary cycle inequalities, which gives a new insight to the procedure of adding BPSPs. Second, we propose an efficient criterion to choose the BPSPs with which we get tighter LP relaxation than existing methods. Third, we propose a new MAP inference algorithm based on the criterion. Experimental results on different datasets show the effectiveness of the new MAP inference algorithm.

2. Preliminaries

In this section, we first briefly overview LP relaxation and dual decomposition, which are the basis of our method.

We denote an MRF $\mathcal{G}=(\mathcal{V},\mathcal{E})$ with a set of random variables $\mathcal{X}=\{X_1,X_2,\ldots,X_n\}$, $(X_p\in L=\{1,2,\ldots,K\})$. The problem of MAP inference is equivalent to the following energy minimization problem:

$$\min_{\mathbf{x}} E(\boldsymbol{\theta},\mathbf{x})=\sum_{p\in\mathcal{V}}\theta_p(x_p)+\sum_{(p,q)\in\mathcal{E}}\theta_{pq}(x_p,x_q), \tag{1}$$

where $\theta_p(\cdot):X_p\to\mathbb{R}$ and $\theta_{pq}(\cdot,\cdot):X_p\times X_q\to\mathbb{R}$ denote the unary and pairwise potential functions for each node and edge, respectively. The energy minimization problem (1) is an integer programming problem that is equivalent to the linear programming problem over the constraint of the *marginal polytope* $\mathcal{M}(\mathcal{G})$ [3]

$$\min_{\boldsymbol{\mu}\in\mathcal{M}(\mathcal{G})} E(\boldsymbol{\theta},\boldsymbol{\mu})=\sum_{p\in\mathcal{V}}\boldsymbol{\theta}_p\cdot\boldsymbol{\mu}_p+\sum_{(p,q)\in\mathcal{E}}\boldsymbol{\theta}_{pq}\cdot\boldsymbol{\mu}_{pq}. \tag{2}$$

In the above equation, $\boldsymbol{\theta}_p=\{\theta_p(\cdot)\}$ and $\boldsymbol{\theta}_{pq}=\{\theta_{pq}(\cdot,\cdot)\}$ represent the vectorized MRF-parameters. Similarly, $\boldsymbol{\mu}_p=\{\mu_p(\cdot)\}$ and $\boldsymbol{\mu}_{pq}=\{\mu_{pq}(\cdot,\cdot)\}$ denote the node marginal probabilities and edge marginal probabilities. The complex constraint $\mathcal{M}(\mathcal{G})$ can be relaxed to the *local consistency polytope* $\mathcal{L}(\mathcal{G})$ with the following formulation:

$$\mathcal{L}(\mathcal{G})=\left(0\le\boldsymbol{\mu}\le1\;\middle|\;\begin{array}{ll}\sum_{u\in L}\mu_p(u)=1, & \forall p\in\mathcal{V}\\ \sum_{v\in L}\mu_{pq}(u,v)=\mu_p(u), & \forall pq\in\mathcal{E}\end{array}\right).$$

The technique of dual decomposition has been widely used to solve the LP relaxation [6,9,16,12,4,5]. Suppose we decompose an original energy minimization problem into a set of tractable subproblems denoted by $\mathcal{SP}(\mathcal{G})=\{\mathcal{SP}_i\}$. The following constraints are satisfied within the decomposition:

$$\sum_{\mathcal{SP}_i\in\mathcal{SP}(p)}\boldsymbol{\theta}_p^{\mathcal{SP}_i}=\boldsymbol{\theta}_p,\qquad\sum_{\mathcal{SP}_i\in\mathcal{SP}(pq)}\boldsymbol{\theta}_{pq}^{\mathcal{SP}_i}=\boldsymbol{\theta}_{pq}, \tag{3}$$

where $\mathcal{SP}(p)$ and $\mathcal{SP}(pq)$ denote the set of subproblems that contain node p and edge pq, respectively. The dual function can be obtained as follows by appropriately introducing Lagrangian multipliers $\{\lambda^{\mathcal{SP}_i}\}=\{\lambda_p^{\mathcal{SP}_i},\lambda_{pq}^{\mathcal{SP}_i}\}$ [4]

$$\max_{\{\lambda^{\mathcal{SP}_i}\}\in\boldsymbol{\Lambda}}\sum_{\mathcal{SP}_i\in\mathcal{SP}(\mathcal{G})}\min_{\boldsymbol{\mu}^{\mathcal{SP}_i}}E(\boldsymbol{\theta}^{\mathcal{SP}_i}+\lambda^{\mathcal{SP}_i},\boldsymbol{\mu}^{\mathcal{SP}_i}), \tag{4}$$

where $\boldsymbol{\Lambda}$ satisfies

$$\boldsymbol{\Lambda}=\left(\{\lambda^{\mathcal{SP}_i}\}\;\middle|\;\sum_{\mathcal{SP}_i\in\mathcal{SP}(p)}\lambda_p^{\mathcal{SP}_i}=0,\ \sum_{\mathcal{SP}_i\in\mathcal{SP}(pq)}\lambda_{pq}^{\mathcal{SP}_i}=0\right).$$

Eq. (4) provides lower bounds of the original energy minimization problem and it can be optimized using the projected subgradient method or the block coordinate ascent method [4,17]. The update strategy of the projected subgradient method is

$$\lambda_p^{\mathcal{SP}_i}+=\alpha_t\cdot\left(\boldsymbol{\mu}_p^{\mathcal{SP}_i}-\frac{\sum_{\mathcal{SP}_j\in\mathcal{SP}(\mathcal{G})}\boldsymbol{\mu}_p^{\mathcal{SP}_j}}{|\mathcal{SP}(\mathcal{G})|}\right), \tag{5}$$

$$\lambda_{pq}^{\mathcal{SP}_i}+=\alpha_t\cdot\left(\boldsymbol{\mu}_{pq}^{\mathcal{SP}_i}-\frac{\sum_{\mathcal{SP}_j\in\mathcal{SP}(\mathcal{G})}\boldsymbol{\mu}_{pq}^{\mathcal{SP}_j}}{|\mathcal{SP}(\mathcal{G})|}\right). \tag{6}$$

It is well known that the tightest lower bound of the tree-structured subproblems is equivalent to the optimal value of the pairwise LP relaxation and with more complex subproblems, we may achieve tighter LP relaxations [4].

3. The equivalence between adding BPSPs and enforcing k-ary cycle inequalities

We know that the tree-structured subproblems equal to enforcing pairwise consistency constraints in the LP relaxations. Adding BPSPs in the dual decomposition can tighten the pairwise LP relaxation. However, it is unclear what kind of higher-order constraints are enforced by adding BPSPs. In this section, we prove that adding a BPSP corresponds to enforcing a group of k-ary cycle inequalities constraints.

3.1. BPSP and k-ary cycle inequalities

For binary planar Ising model with the following energy function:

$$E(\mathbf{x})=\sum_{(p,q)\in\mathcal{E}}[x_p\neq x_q]\cdot\theta_{pq}, \tag{7}$$

where $[\cdot]$ denotes the indicator function, there exists a very efficient implementation to conduct MAP inference by reducing the energy minimization problem to a minimum weight perfect matching problem [13,14].

For the non-binary case, the method of minimum weight perfect matching cannot be directly used. Yarkony et al. [15] show that if the potential of every edge in an MRF happens to take only two different values, we can project the edge potential to an equivalent binary form and then project the solution back. This way, BPSPs can be introduced in dual decomposition as tractable subproblems, with which we can obtain more accurate solution than the methods using tree-structured subproblems. We follow the notation in [15] and introduce the concept of binary planar subproblems (BPSP). We define a non-binary planar graph to be an equivalent BPSP if its edge potentials have the following form:

$$\theta_{pq}(x_p,x_q)=\begin{cases}\vartheta_{pq}:(x_p\in S_p)\oplus(x_q\in S_q)\\ 0:\text{otherwise}\end{cases}, \tag{8}$$

where $\oplus$ denotes the operation of exclusive-or. S_p ($S_p\cup\overline{S}_p=\{1,2,\ldots,K\}$) and S_q ($S_q\cup\overline{S}_q=\{1,2,\ldots,K\}$) denote the subsets of the state space of variable X_p and X_q, respectively. The state space of every node in BPSP is divided into two complementary parts. BPSP with the above potentials can be projected to a binary planar Ising model with energy function (7) and then be solved by minimum weight perfect matching.

In order to get tighter LP relaxations, we can add higher-order constraints to the *local consistency polytope*. K-ary cycle inequalities are one kind of higher-order constraints with the following formulation [10]:

$$\sum_{mn\in C\setminus F}(\mu_{mn}^{\pi}(1,0)+\mu_{mn}^{\pi}(0,1))+\sum_{mn\in F}(\mu_{mn}^{\pi}(0,0)+\mu_{mn}^{\pi}(1,1))\ge1,$$

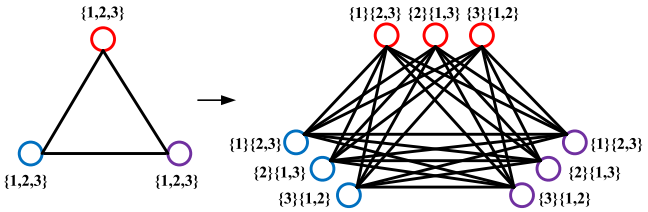


Fig. 1. Illustration of the projection graph of a three node cycle (left). Every node has three possible partitions and the projection graph is shown on right.

where C denotes a cycle and $F \subseteq C$ ($|F|$ is odd), m and n denote the binary nodes after projecting. The edge marginal probability in the projection graph is defined as

$$\mu^{\pi}_{mn}(x_m, x_n) = \sum_{u \in L, \pi_p(u) = x_m; v \in L, \pi_q(v) = x_n} \mu_{pq}(u, v),$$

where $x_m, x_n \in \{0, 1\}$, π_p and π_q denote partitions of the nodes' state space. Fig. 1 gives a simple example to show how to construct a projection graph. If every node's state space is divided into one state versus all the other states, it is called a *k-projection* graph. However, if a projection graph consists all possible partitions for every node, it is called *full projection* graph (For example, a node space $\{1, 2, 3, 4\}$ can be divided into $\{1, 3\}$ and $\{2, 4\}$ in a *full projection* graph).

3.2. Adding BPSP is equivalent to enforcing k-ary cycle inequalities

From the above definitions, we can see that the k-ary cycle inequalities are defined on the projection graph with every node's state space divided into two complementary subsets. Similarly, every node's state space is also partitioned into two complementary subsets in BPSPs.

Proposition 1. *Adding one cycle C of a BPSP in the dual decomposition is equivalent to enforcing all k-ary cycle inequalities that are defined on the projection cycle C'.*

Proof. Consider any cycle C in the BPSP, the potentials of every edge $pq \in C$ have the form of (8). For every edge $pq \in C$, denote the partitions of nodes p and q as π_p ($\pi_p : S_p \to x_m, \overline{S}_p \to \overline{x}_m$) and π_q($\pi_q : S_q \to x_n, \overline{S}_q \to \overline{x}_n$) respectively, where $x_m, x_n \in \{0, 1\}$. With the special potentials of C, we can equivalently project the cycle C to a binary projection cycle C' according to the following equation:

$$\mu^{\pi}_{mn}(x_m, x_n) = \sum_{s_p \in L, \pi_p(s_p) = x_m; s_q \in L, \pi_q(s_q) = x_n} \mu_{pq}(s_p, s_q).$$

So the procedure of adding the cycle C in the dual decomposition enforces the following *cycle consistency constraints* of C' in the LP relaxations:

$$CYCLE(C') = \left\{ \boldsymbol{\mu} \middle| \exists \tau_{C'} > 0, \begin{array}{c} \sum_{\mathbf{x}_{C' \setminus m,n}} \tau_{C'}(\mathbf{x}_{C'}) = \mu^{\pi}_{mn}(x_m, x_n) \\ \sum_{\mathbf{x}_{C'}} \tau_{C'}(\mathbf{x}_{C'}) = 1 \end{array} \right\},$$

where $mn \in C'$. Consider any binary pairwise MRF consisting only one cycle, Sontag et al. [18] prove that the cycle consistency constraints give the same relaxation with the k-ary cycle inequalities defined on the cycle. In their work, they analyze the dual of the LP relaxation considering all the k-ary cycle inequalities with the following formulation:

$$\begin{aligned} \max_{\lambda \geq 0, \delta} \min_{\mu \geq 0} & \sum_{p} \sum_{x_p} \mu_p(x_p)(\theta_p(x_p) + \sum_{q \in N(p)} \delta_{qp}(x_p)) - \sum_{C,F,\pi} \lambda_{C,F,\pi} \\ & + \sum_{pq} \sum_{x_p, x_q} \mu_{pq}(x_p, x_q)(\theta_{pq}(x_p, x_q) - \delta_{qp}(x_p) - \delta_{pq}(x_q) \\ & + \sum_{C,F: pq \in C \setminus F} \lambda_{C,F,\pi} 1[\pi_p(x_p) \neq \pi_q(x_q)] + \sum_{C,F: pq \in F} \lambda_{C,F,\pi} 1[\pi_p(x_p) = \pi_q(x_q)]), \end{aligned}$$

where $\delta_{qp}(x_p)$ and $\delta_{pq}(x_q)$ are Lagrange multipliers for the pairwise consistency constrains, and $\lambda_{C,F,\pi}$ are Lagrange multipliers for the k-ary cycle inequalities. They also analyze the dual of the cycle relaxation and prove the equivalence using LP duality. For more details of the proof, please see the supplementary material to [18] (or [19], Chapter 7).

Using the above conclusion, the cycle consistency constraints $CYCLE(C')$ of the projected cycle C' is equivalent to the k-ary cycle inequalities defined on the cycle C'. So we conclude that adding one cycle C of a BPSP in the dual decomposition is equivalent to enforcing all k-ary cycle inequalities that are defined on the projection cycle C' (all possible F'):

$$\sum_{mn \in C' \setminus F'} (\mu^{\pi}_{mn}(1,0) + \mu^{\pi}_{mn}(0,1)) + \sum_{mn \in F'} (\mu^{\pi}_{mn}(0,0) + \mu^{\pi}_{mn}(1,1)) \geq 1. \quad \square$$

Theorem 1. *For a BPSP, consider the projection graph where we only include the nodes that have the same partitions with the BPSP. Adding a single BPSP then corresponds to enforcing or repairing all the k-ary cycle inequalities that can be derived using this projection graph.*

Proof. From Proposition 1, we see that adding one cycle of a BPSP in the dual decomposition is equivalent to enforcing all k-ary cycle inequalities that are defined on its corresponding projection cycle. Considering all the cycles contained in the BPSP, then adding a single BPSP corresponds to enforcing or repairing all the k-ary cycle inequalities that can be derived using the projection graph, in which we only include the nodes that have the same partitions with the BPSP. □

4. A new criterion for choosing BPSPs based on a separation algorithm

In this section, we propose a new criterion for choosing BPSPs and present an inference algorithm applying this criterion. We try to add the BPSP that can tighten the current LP relaxation as much as possible in each step. Observing the equivalence between adding BPSPs and enforcing k-ary cycle inequalities, we try to add the BPSPs that corresponds to enforcing the *violated* cycle inequalities. To this end, we propose a criterion by using a separation algorithm to find the violated k-ary cycle inequalities and then pack these cycles in a single BPSP to be added.

4.1. Separation algorithms

There are a large number of cycle inequalities, while adding all cycle inequalities is infeasible and also unnecessary. It has been demonstrated that MAP inference in a binary Markov random fields is equivalent to a linear programming over the cut polytope [20]. Barahona and Mahjoub [21] give an efficient separation algorithm to find the violated cycle inequalities on the cut polytope. Sontag and Jaakkola [10] extend the separation algorithm to the non-binary case, and propose a new class of tighter outer bounds than the *local consistency polytope* by iteratively adding cycle inequality constraints. In their method, they first solve the pairwise LP relaxation to get the solution (usually not accurate), and then use the separation algorithm to adaptively find the k-ary cycle inequality that is most violated by the current solution. Fig. 2 illustrates the relationship of the *marginal polytope*, the *local consistency polytope* and the violated k-ary cycle inequalities. The two red lines refer to k-ary cycle inequalities that are violated by the current factional solutions $\tilde{\boldsymbol{\mu}}$. Adding the violated k-ary cycle inequalities will obtain tighter outer bound so as to get more accurate solutions.

The separation algorithm takes current primal solution $\tilde{\boldsymbol{\mu}}$ and the projection graph as input and returns the violated k-ary cycle inequalities. In our paper, we use the separation algorithm described by Sontag and Jaakkola [10]. However, there are some differences in using the algorithm. In the paper of [10], only the most violated k-ary cycle inequality is added in each iteration. Specifically, they find up to one

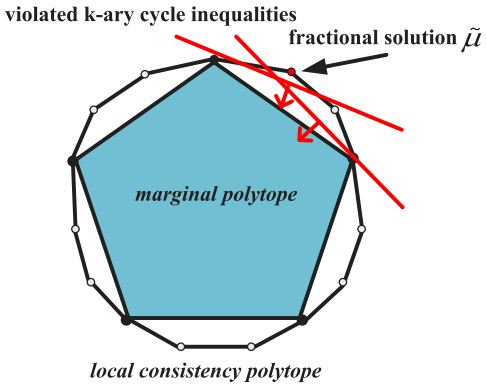


Fig. 2. Illustration the relationship of the *marginal polytope*, the *local consistency polytope* and k-ary cycle inequalities.

violated k-ary cycle inequality by running the shortest-path algorithm once for each node in the projection graph. The returned shortest path for one node defines a cycle and gives the minimum value of $\sum_{mn\in C\setminus F}(\mu^{\pi}_{mn}(1,0)+\mu^{\pi}_{mn}(0,1))+\sum_{mn\in F}(\mu^{\pi}_{mn}(0,0)+\mu^{\pi}_{mn}(1,1))$. If this value is less than 1, then they find a violated k-ary cycle inequality. If not, they cannot find any violated k-ary cycle inequality on this node. After running the shortest-path algorithm for all the nodes, they select the shortest path among all these paths and obtain the most violated k-ary cycle inequality. In our paper, we find not only the most violated cycle inequality, but also all the violated cycle inequalities, from the most violated to the least violated.

The number of violated cycles found by the separation algorithm depends on the current primal solution and the structure of the projection graph. If the current primal solution is close to the optimal solution, we will find fewer violated cycles. In order to use the separation algorithm, we need to compute the current primal solution based on the dual solution. In our algorithm, we use the projected subgradient method to solve the dual function (4). Computing the primal value from the current dual solution when using the subgradient method has been well studied in the field of optimization [22,23]. We use the following weighted average of primal sequence $\{\boldsymbol{\mu}_t\}$ generated by the subgradient method to compute the current approximate primal solutions

$$\tilde{\boldsymbol{\mu}}_T=\frac{\sum_{t=1}^{T}\alpha_t\boldsymbol{\mu}_t}{\sum_{t=1}^{T}\alpha_t},\quad T=1,2,\ldots, \tag{9}$$

where $\{\alpha_t\}$ denotes the stepsize sequence of the subgradient method. Nedic and Ozdaglar [23] demonstrate that the estimation $\tilde{\boldsymbol{\mu}}_T$ will converge to the primal solution, as $T\to\infty$.

4.2. Choosing BPSP by packing violated k-ary cycle inequalities

Suppose we have found a set of violated k-ary cycle inequalities using the separation algorithm. We then try to pack as many of these cycles as possible within a single BPSP. Since every node's state space in the violated projected cycles is partitioned into two complementary parts, we can determine the partitions of the nodes in the new added BPSP by packing these violated cycles in a greedy manner. The initial edge potentials of the added BPSP are zero-valued and then they are updated by the subgradient method. For the convenience of explanation, we use a simple example to show the procedure.

Example 1. Fig. 3(a) shows a 4×4 grid MRF and the node state space is $\{1,2,3\}$. Our method first decomposes the original MRF into a set of "row" and "column" trees, and then use dual decomposition technique to get the primal solution. We use the separation algorithm to find the violated k-ary cycle inequalities based on the current primal solution. Assume the violated cycles we find are $\{C\}=\{C_1,C_2,C_3\}$, where

$C_1=\{1_{(1;2,3)}:2_{(1;2,3)}:3_{(2;1,3)}:7_{(3;1,2)}:6_{(1;2,3)}:5_{(1;2,3)}:1_{(1;2,3)}\}$,

$C_2=\{9_{(2;1,3)}:10_{(1;2,3)}:11_{(3;1,2)}:15_{(2;1,3)}:14_{(2;1,3)}:$

$13_{(1;2,3)}:9_{(2;1,3)}\}$,

$C_3=\{2_{(3;1,2)}:3_{(2;1,3)}:4_{(1;2,3)}:8_{(2;1,3)}:12_{(2;1,3)}:16_{(1;2,3)}:$

$15_{(2;1,3)}:14_{(2;1,3)}:10_{(1;2,3)}:6_{(1;2,3)}:2_{(3;1,2)}\}$.

The notations like $2_{(1;2,3)}$ denotes that node 2 has partition $S_2=\{1\}$, $\bar{S}_2=\{2,3\}$. We consider the problem of how to pack these cycles into a single BPSP (see Fig. 3(b)).

Different cycles may have different partitions at one node. As shown in Example 1, C_1 and C_3 disagree on the partition of node 2. In cycle C_1, the partition of node 2 is $S_2=\{1\}$, which is different from that of $S'_2=\{3\}$ in cycle C_3. Besides, if the MRF is a non-planar graph, then we must ensure that constructed BPSP to be planar. For a graph with N nodes, Boyer and Myrvold [24] give a simple $O(N)$ time-complexity algorithm that can decide if the graph is planar. The planar-testing algorithm takes the basic operation of edge addition while preserving planarity. Observing these factors, we propose the following efficient packing-criterion to pack the violated cycles into BPSPs.

Packing-criterion: Suppose we have obtained m cycles denoted by $C_1,C_2,\ldots,C_m$. Traverse every cycle C_i (from 1 to m) and find the partitions of the nodes appeared in C_i. If C_i disagrees with previous cycles in the partition of any node, do not pack C_i into the current BPSP. Besides, use the planar-testing algorithm to check whether packing C_i into the current BPSP makes it non-planar. If so, do not pack C_i into the current BPSP.

This packing-criterion may discard some violated cycles. If the number of discarded cycles is small, we just discard them and start the next iteration. If the number is large, we can pack them into other BPSPs and then add these BPSPs simultaneously. In our experiment, we empirically find that the number of disagreeing cycles is usually small and we choose to add one BPSP each time. We propose an adaptive MAP inference algorithm, which uses the criterion of choosing BPSPs described above, as shown in Algorithm 1. We can terminate the algorithm if we cannot get more accurate MAP solutions after adding a certain number of BPSPs. In Algorithm 1, the "tolerance" is usually set as 10^{-4} or even smaller.

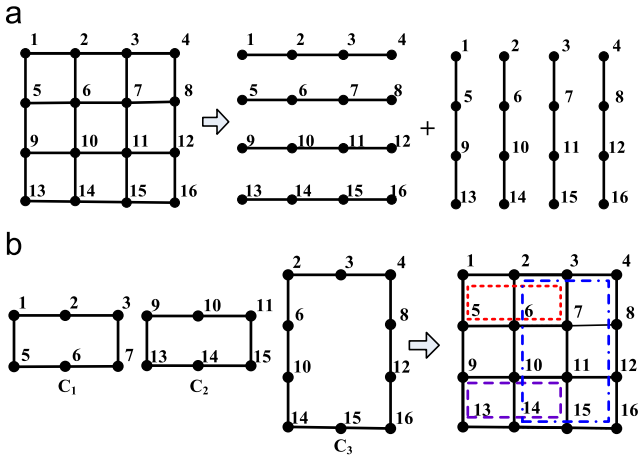


Fig. 3. (a) The original MRF is decomposed into a set of "row" and "column" trees and (b) we use the separation algorithm to find three violated cycles and try to pack these cycles into a single BPSP: (a) Tree decomposition and (b) packing cycles.

Algorithm 1. Adding BPSPs with separation algorithm

1: Start with tree decomposition.
2: Set Gap = primal-dual;
3: **while** Gap > *tolerance* **do**
4: Get the solution $\tilde{\boldsymbol{\mu}}$ using the weighted average strategy (9);
5: Violated cycle inequalities $\{C\} \leftarrow$ *Separation algorithm*$(\mathcal{G}, \tilde{\boldsymbol{\mu}})$;
6: Pack these violated cycles into a BPSP;
7: Add the BPSP and use the projected subgradient method (5–6) to optimize the dual function;
8: **end while**
9: Return MAP $\mathbf{x}^*$.

Theorem 2. *The procedure of choosing and adding BPSPs, which are constructed by packing the violated k-ary cycle inequalities based on current primal solution, will enforce these violated k-ary cycle inequalities at once.*

Proof. From Theorem 1 we know that adding a BPSP is equivalent to enforcing all the k-ary cycle inequalities that can be derived using the projection graph (only include the nodes that have the same partitions with the BPSP). Conversely, if we find a set of violated k-ary cycle inequalities and pack these cycles into a single BPSP to be added as a subproblem, it will enforce these violated k-ary cycle inequalities. Since the choice of binary partitions for the nodes in the violated cycles is accomplished automatically by the separation algorithm, we can easily determine the partitions of the nodes in the BPSP by packing these violated k-ary cycle inequalities.

However, since k-ary cycle inequalities are slightly weaker than the cycle relaxation when the number of discrete states $K \geq 3$ (Theorem 3.4.3 in [19]), the tightest bound that our method can obtain is looser than that of cycle relaxation. □

As we know, it is impossible for large-scale inference problem to explicitly enforce all the cycle consistency constraints. So iteratively adding selected higher order constraints is an efficient strategy. Although the tightest bound that our method can obtain is looser than that of cycle relaxation, we can obtain accurate MAP configuration after adding only a small number of BPSPs in most problems. It benefits from the fact that adding a BPSP is equivalent to enforcing a large number of violated k-ary cycle inequalities.

5. Related work

Many different approaches can be used for tightening the pairwise LP relaxation. Komodakis and Paragios [25] present an algorithm for tightening the relaxation by repairing inconsistent cycles. Sontag et al. [16] propose a message passing algorithm to tighten the pairwise LP relaxation by iteratively adding cluster consistency constraints. They define a scoring function to decide the clusters to be added and this algorithm is guaranteed to improve the approximation successively. Werner [26] uses a method of cutting plane to tighten the pairwise LP relaxation by adding short cycles.

Our work is closely related to the work of Sontag and Jaakkola [10]. Different from their work, we try to solve the problem via dual decomposition, which takes advantage of the special structure of the LP and makes the problem easier. When we add a single BPSP, it is equivalent to enforcing many violated k-ary cycle inequalities. Sontag et al. [18] propose a new algorithm to find the violated k-ary cycle inequalities in the dual and add the full cycles to the dual decomposition iteratively. Different from their work, we find the violated k-ary cycle inequalities in primal and enforce a group of these violated cycles inequalities by adding the corresponding BPSP. The work of Yarkony et al. [15] is also closely related. One major difference is that we focus on how to choose BPSPs and present an efficient criterion based on the separation algorithm. Besides, we give a new insight of the procedure of adding BPSPs and prove that adding BPSPs is equivalent to enforcing the k-ary cycle inequalities. Last but not least, our method not only adds *one-versus-all* BPSPs, but also adds other BPSPs that are constructed from the *full projection* graph.

6. Experiments

To assess the effectiveness of our criterion, we compare our algorithm against four algorithms: convergent tree-reweighted message passing (TRW-S) [7], tightening MRF relaxation using planar subproblems [15], tightening LP relaxations using message-passing (MPLP) [16] and choosing frustrated cycles in MAP inference [18]. In the work of [15], they construct a *one-versus-all* BPSP by dividing every node's state space into $S_p = \{x_p^*\}$ and $\overline{S}_p$, where $\{x_p^*\}$ is the current MAP solution. In our method, we use the proposed criterion to adaptively choose the BPSPs.

All the algorithms except MPLP and the algorithm of [18] are implemented using MATLAB with version 7.10.0.499 (R2010a). We use the code of MPLP and the code of [18] provided by the authors (from the authors' websites). We run the experiments on a computer with i3–370M CPU and 6 GB RAM.

6.1. Synthetic problems

We construct our problem instances following that of [15]. We consider $N \times N$ grid MRFs with $K (K \in \{3, 4\})$ states per node. The nodes and edges' random potentials are described as

$$\theta_{p;u} \sim U[-1, 1], \quad \theta_{pq;uv} \sim \begin{cases} U[-a, a] & |u-v| = 1 \\ 0 & u = v \\ 16 & \text{otherwise} \end{cases},$$

where $U[-1, 1]$ and $U[-a, a]$ are uniform distributions on $[-1, 1]$ and $[-a, a]$, respectively. We use the values $a \in \{2, 8\}$ in the experiment. For $K = 3, a = 2$, we randomly generate 100 data samples according to the above distributions. We run all the algorithms on these 100 samples and then report the average results over these 100 samples. We do the same for the other values of a and K. We use the code of Batra and Chen [27] to solve the BPSP, which is a mex-wrapper for the inference method in the binary planar Ising model [13,14]. For the separation algorithm, we use the code of Sontag and Jaakkola [10] to find the violated k-ary cycle inequalities. We refer to our algorithm as "BPSP +SEP" and denote the algorithm of Yarkony et al. [15] as "BPSP+MAP". Both of these two algorithms start with decomposing the original MRF into a set of "row" and "column" trees, which is mathematically equivalent to that of "covering tree" [15]. For both algorithms, we choose the step size of the subgradient method as $\{\alpha_t = \alpha/\sqrt{t}\}$. All settings of these two algorithms are strictly the same except that they use different strategies for choosing BPSPs.

We refer to the MPLP algorithm as "MPLP08" and denote the algorithm of the paper [18] as "MPLP12". For all these algorithms except TRW-S, we perform 200 iterations for tree structured subproblems. After that, we begin to tighten the LP relaxation for different algorithms. For "BPSP+SEP" and "BPSP+MAP", when we add a BPSP, we perform 100 iterations of the subgradient optimization and then add a new one.

As a measure of performance, we plot the values of dual functions versus the iterations and also show the results versus running time. The values of dual function provide lower bounds of the primal energy function and higher values provide more accurate approximations. Fig. 4(a–c) shows the average results over 100 problem instances on 30×30 grid MRFs with different potentials. In the experiment, we adaptively add 7 *one-versus-all* BPSPs for both "BPSP+SEP" and "BPSP+MAP". As Fig. 4 shows, our algorithm can obtain higher dual objective values than that of [15]

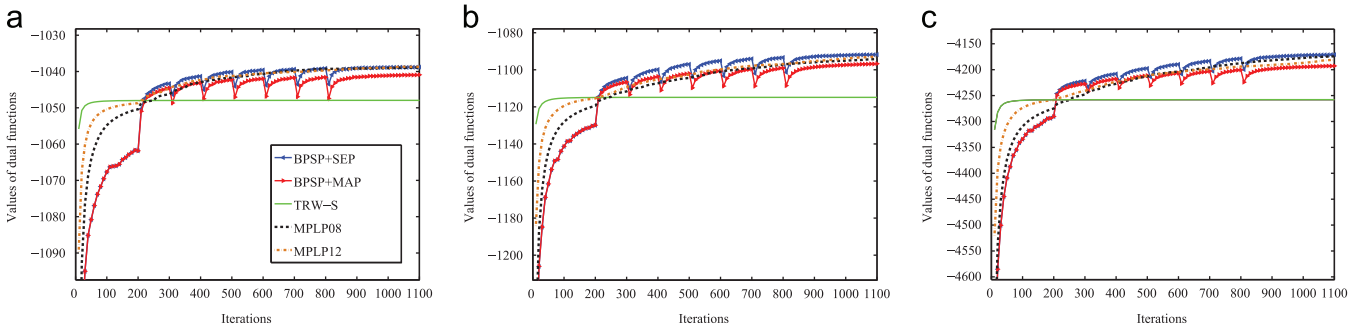


Fig. 4. Average results on 30×30 grids with different state space and parameters. Horizontal axis: numbers of iterations. Vertical axis: values of dual functions that provide lower bounds of the energy function: (a) 30×30, $K=3$, $a=2$; (b) 30×30, $K=4$, $a=2$ and (c) 30×30, $K=4$, $a=8$.

on all these problems. The dual value decreases a little when we add a new BPSP because of the re-initialization of the step size of the subgradient method. We believe that better strategies of choosing the step size will make this decrease smaller.

Our algorithm starts out with a lower energy than that of TRW-S. The main reason is that we are using the subgradient method to optimize the dual function, which is slower than that of the block coordinate ascent method used by TRW-S. Besides, if we decompose the original MRF into more complex tree-structured subproblems than "row" and "column" trees, our algorithm will also converge faster. However, even though our algorithm start out with a lower energy than the TRW-S, the dual objective values of our algorithm can quickly surpass those of TRW-S after we add the first BPSP. Compared with "MPLP08" and "MPLP12", our algorithm can get higher dual objective values after we add a small number of BPSPs. This is benefit from the fact that adding a selected BPSP is equivalent to enforcing a large number of violated k-ary cycle inequalities each time. To make a quantitative comparison of different algorithms' performance in accuracy, we introduce a scoring procedure following the paper of [28]. We denote the dual objective values of the compared algorithms as $e_1, \ldots, e_J$, and set the score of the i-th algorithm as

$$s_i = \frac{e_i - \min_{1 \le j \le J} e_j}{\max_{1 \le j \le J} e_j - \min_{1 \le j \le J} e_j}. \tag{10}$$

In this way, we assign the best and the worst methods with scores of one and zero respectively. The remaining algorithms are assigned with fraction values, and algorithms with higher dual objective values will be assigned with larger fraction values according to Eq. (10). The scores of Table 1 are computed from the average results in Fig. 4. We can see that the dual objective values of TRW-S are much lower than that of the other four algorithms after we perform 1100 iterations for all these algorithms. The accuracy scores of "BPSP+SEP" are significantly higher than that of "BPSP+MAP", and are comparable with the algorithms of "MPLP08" and "MPLP12". The advantage of our algorithm, compared with "MPLP08" and "MPLP12", is that our algorithm usually converges faster since each time we adding a selected BPSP corresponds to enforcing a large number of violated k-ary cycle inequalities.

Compared with "BPSP+MAP", our algorithm takes more time because we use the separation algorithm to choose BPSPs. Although we spend extra time in selecting BPSPs, we can enforce more violated k-ary cycle inequalities each time. So the total performance is still better than the comparison methods. As shown in Fig. 5, "BPSP+SEP" is worse than "BPSP+MAP" in the early iterations because our algorithm spends extra time in selecting BPSPs. However, our algorithm quickly outperforms "BPSP+MAP" as we add better BPSPs. Note that we are currently using the subgradient method with the typical diminishing step $\{\alpha_t = \alpha/\sqrt{t}\}$ to optimize the dual functions of both "BPSP+MAP" and "BPSP+SEP". We believe that better ways to choose the step length of the subgradient method, using other variants of subgradient methods (e.g. bundle trust system) [29], or using block coordinate ascent method [17] will further speed up our algorithm. We will try these methods in our future work.

Table 1
Averaged scores of different algorithms in solving 30×30 grid MRFs. The scores are computed according to (10), and algorithms with higher dual objective values will be assigned with higher score values.

Algorithm	$K=3$, $a=2$	$K=4$, $a=2$	$K=4$, $a=8$
BPSP+SEP	0.9912	1	1
BPSP+MAP	0.7623	0.7873	0.7496
MPLP08	0.9760	0.9046	0.9628
MPLP12	1	0.9521	0.9070
TRW-S	0	0	0

The state space of every node in *one-versus-all* BPSP is partitioned into one state versus all the other states. *One-versus-all* BPSPs can be constructed from packing the violated k-ary cycle inequalities derived from *k-projection* graph. Different from the work of [15], we add not only *one-versus-all* BPSPs but also other BPSPs that are constructed from the *full projection* graph. In the experiment on 10×10 grid MRFs, we first add the same 6 *one-versus-all* BPSPs for both "BPSP+SEP" and "BPSP+SEP-Full". "BPSP+SEP-Full" denotes the result where we add a 7th BPSP constructed from packing the violated cycles that are derived from the *full projection* graph. "BPSP+SEP" refers to the result where we only consider the *k-projection* graph. Fig. 6 shows the values of dual functions when we add the 7th BPSP (the early iterations are not shown). We can see that adding BPSPs that are constructed from the *full projection* graph can obtain tighter LP relaxation than that of only adding *one-versus-all* BPSPs. The reason is that after we add a few of *one-versus-all* BPSPs, most violated k-ary cycle inequalities that can be derived from the *k-projection* graph are repaired. So we need to consider the *full projection* graph to repair other violated k-ary cycle inequalities.

6.2. Real problem

In order to further assess the effectiveness of our criterion on real problems, we test our algorithm on the image denoising problem. Specifically, we create a "CVPR" image with noise and formulate the image denoising as an energy minimization problem in a pairwise MRF as described in [30,31]. The "CVPR" image is a 24×62 gray image with 16 intensity values. Fig. 7(a) shows the input image of "CVPR". Fig. 7(b) and (c) shows the denoised images using the algorithms of "BPSP+SEP" and "BPSP+MAP", respectively. From the result shown in Fig. 7(d), we can see that our algorithm achieves a more accurate result than "BPSP+MAP". However, the improvement is not so apparent compared to the synthetic problems. The reason is that the tree-structured subproblems are nearly sufficient to get the accurate solution in many computer vision problems, as mentioned in [7].

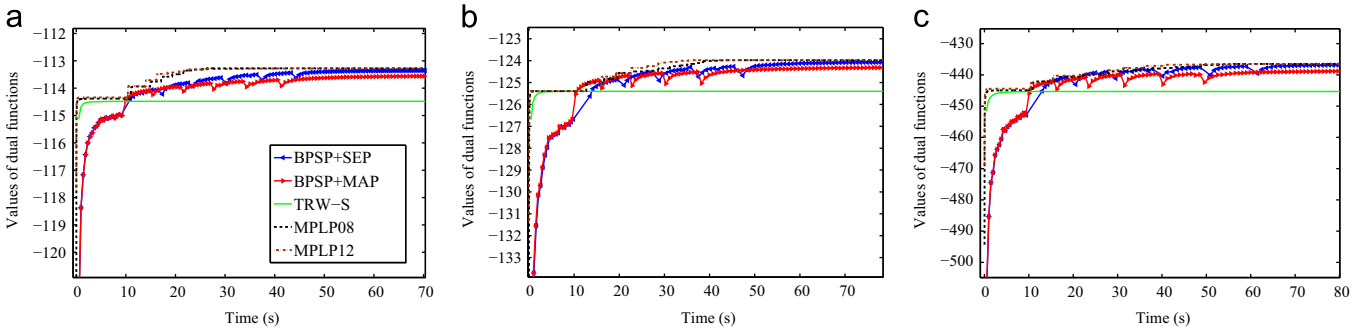


Fig. 5. Average results on 10×10 grids with different state space and parameters. Horizontal axis: running time. Vertical axis: value of dual function that provides lower bound of the energy function: (a) 10×10, $K=3$, $a=2$; (b) 10×10, $K=4$, $a=2$ and (c) 10×10, $k=4$, $a=8$.

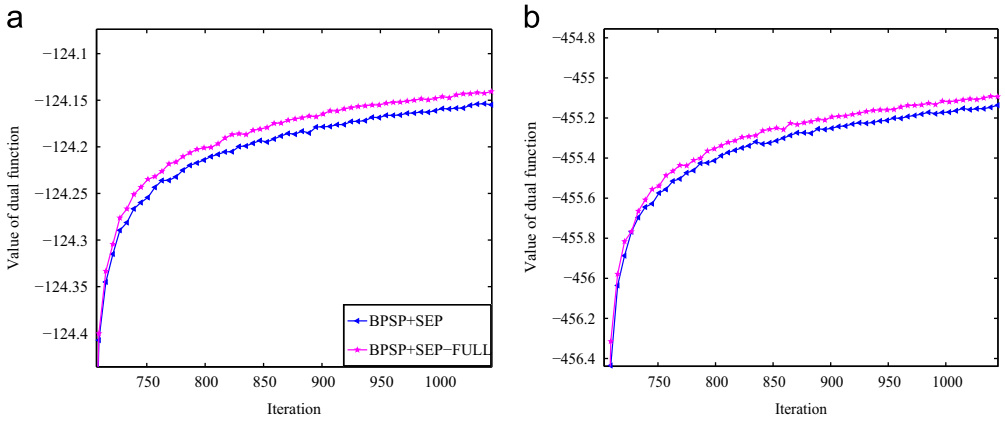


Fig. 6. The performance difference between adding *one-versus-all* BPSPs and adding BPSPs that are constructed from the *full projection* graph: (a) 10×10, $K=4$, $a=2$ and (b) 10×10, $K=4$, $a=8$.

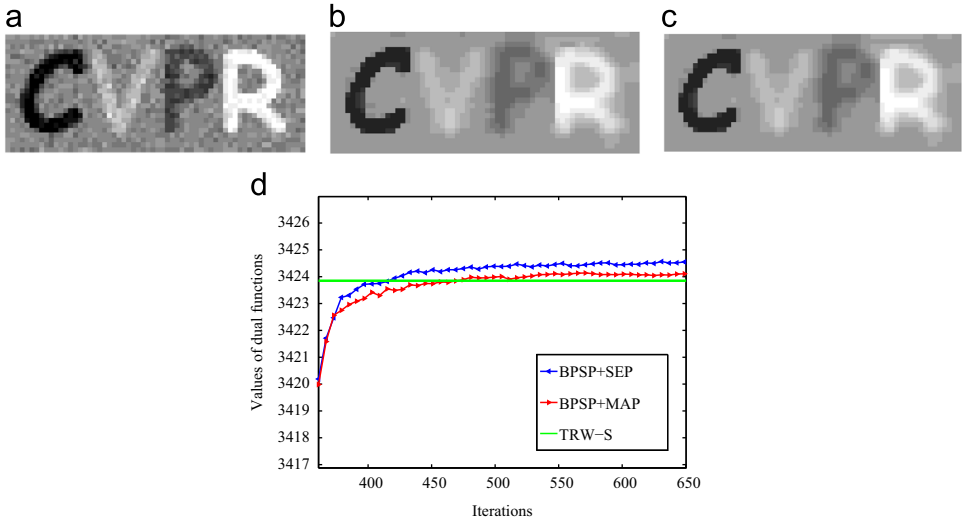


Fig. 7. Performance comparisons of "BPSP+SEP" and "BPSP+MAP" on an image denoising problem. Note that in (d), we show the values of dual functions from iteration 350 to focus on visualizing the performance after the initial tightening: (a) Input image; (b) "BPSP+SEP"; (c) "BPSP+MAP" and (d) values of dual functions.

7. Conclusion and future work

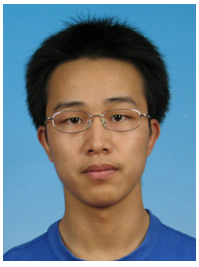
In this paper, we prove the equivalence between adding a BPSP and enforcing the k-ary cycle inequalities. We then propose a criterion for choosing BPSPs in MAP-MRF inference by using the separation algorithm to find the violated k-ary cycle inequalities and packing as many of these cycles as possible in a BPSP. Experimental results validate that different BPSPs have different effects on tightening the LP relaxation and show the effectiveness of our criterion. In this paper, we use the subgradient method to optimize the dual functions because it is a widely used technique in MAP inference and it is also easy to implement. However, some other optimization methods can also be tried, such as the block coordinate ascent method, the bundle trust method [29], the alternating direction method (ADLP) [32], and the recent method of [33].

Acknowledgments

We thank David Sontag and Julian Yarkony for helpful discussions and useful suggestions. This work was supported by National Natural Science Foundation of China (Project No. 61071131 and Project No. 61271388), Beijing Natural Science Foundation (No. 4122040), Research Project of Tsinghua University (No. 2012Z01011) and Doctoral Fund of Ministry of Education of China (No. 20120002110036). This work was supported, in part, by AFOSR FA9550-11-1-0327, NSF IIS-1017199, and ARL under Cooperative Agreement Number W911NF-10-2-0060 (DARPA Mind's Eye).

References

[1] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, MIT Press, 2009.
[2] Y. ming Cheung, M. Li, X. Cao, Lip segmentation and tracking under MAP-MRF framework with unknown segment number, Neurocomputing 104 (2013) 155–169.
[3] M.J. Wainwright, M.I. Jordan, Graphical models, exponential families, and variational inference, Found. Trends Mach. Learn. 1 (2008) 1–305.
[4] N. Komodakis, N. Paragios, G. Tziritas, MRF energy minimization and beyond via dual decomposition, IEEE Trans. Pattern Anal. Mach. Intell. 33 (2011) 531–552.
[5] D. Sontag, A. Globerson, T. Jaakkola, Introduction to dual decomposition for inference, in: Optimization for Machine Learning, MIT Press, 2011.
[6] M.J. Wainwright, T.S. Jaakkola, A.S. Willsky, MAP estimation via agreement on trees: message-passing and linear programming, IEEE Trans. Inf. Theory 51 (11) (2005) 3697–3717.
[7] V. Kolmogorov, Convergent tree-reweighted message passing for energy minimization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (2006) 1568–1583.
[8] T. Werner, A linear programming approach to max-sum problem: a review, IEEE Trans. Pattern Anal. Mach. Intell. 29 (2007) 1165–1179.
[9] A. Globerson, T. Jaakkola, Fixing max-product: convergent message passing algorithms for MAP LP-relaxations, in: Advances in Neural Information Processing Systems, MIT Press, 2008, pp. 553–560.
[10] D. Sontag, T. Jaakkola, New outer bounds on the marginal polytope, in: Advances in Neural Information Processing Systems, MIT Press, 2008, pp. 1393–1400.
[11] C. Yanover, T. Meltzer, Y. Weiss, Linear programming relaxations and belief propagation-an empirical study, J. Mach. Learn. Res. 7 (2006) 1887–1907.
[12] D. Batra, A. Gallagher, D. Parikh, T. Chen, Beyond trees: MRF inference via outer-planar decomposition, in: IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2496–2503.
[13] N.N. Schraudolph, D. Kamenetsky, Efficient exact inference in planar Ising models, in: Advances in Neural Information Processing Systems, vol. 21, 2009, pp. 1417–424.
[14] V. Kolmogorov, Blossom V: a new implementation of a minimum cost perfect matching algorithm, Math. Programming Comput. 1 (2009) 43–67.
[15] J. Yarkony, R. Morshed, A. Ihler, C. Fowlkes, Tightening MRF relaxations with planar subproblems, in: Proceedings of the Twenty-Seventh Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-11), AUAI Press, 2011, pp. 770–777.
[16] D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, Y. Weiss, Tightening LP relaxations for MAP using message passing, in: Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-08), AUAI Press, 2008, pp. 503–510.
[17] D.P. Bertsekas, Nonlinear Programming, Athena Scientific, 1999.
[18] D. Sontag, D.K. Choe, Y. Li, Efficiently searching for frustrated cycles in MAP inference, in: Proceedings of the Twenty-Eighth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-12), AUAI Press, 2012, pp. 795–804.
[19] D. Sontag, Approximate Inference in Graphical Models Using LP relaxations, Ph.D. Thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2010.
[20] F. Barahona, M. Grötschel, M. Jünger, G. Reinelt, An application of combinatorial optimization to statistical physics and circuit layout design, Oper. Res. 36 (1988) 493–513.
[21] F. Barahona, A.R. Mahjoub, On the cut polytope, Math. Programming 36 (1986) 157–173.
[22] H.D. Sherali, G. Choi, Recovery of primal solutions when using subgradient optimization methods to solve Lagrangian duals of linear programs, Oper. Res. Lett. 19 (3) (1996) 105–113.
[23] A. Nedic, A.E. Ozdaglar, Approximate primal solutions and rate analysis for dual subgradient methods, SIAM J. Optim. 19 (4) (2009) 1757–1780.
[24] J.M. Boyer, W.J. Myrvold, On the cutting edge: simplified o(n) planarity by edge addition, J. Graph Algorithms Appl. 8 (2004) 241–273.
[25] N. Komodakis, N. Paragios, Beyond loose LP-relaxations: optimizing MRFs by repairing cycles, in: Proceedings of the 10th European Conference on Computer Vision: Part III, ECCV '08, 2008, pp. 806–820.
[26] T. Werner, High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF), in: IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 109–116.
[27] D. Batra, T. Chen, Dynamic planar-cuts: efficient computation of min-marginals for outer-planar models, in: Workshop on Discrete Optimization in Machine Learning (DISCML), NIPS, 2009.
[28] P. Pletscher, S. Wulff, LPQP for MAP: putting LP solvers to better use, in: Proceedings of the 29th International Conference on Machine Learning (ICML-12), ICML '12, 2012, pp. 783–790.
[29] H. Schramm, J. Zowe, A version of the bundle idea for minimizing a nonsmooth function: conceptual idea, convergence analysis, numerical results, SIAM J. Optim. 2 (1) (1992) 121–152.
[30] C. Rother, V. Kolmogorov, V. Lempitsky, M. Szummer, Optimizing binary MRFs via extended roof duality, in: IEEE Conference on Computer Vision and Pattern Recognition, 2007.
[31] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient belief propagation for early vision, Int. J. Comput. Vision 70 (1) (2006) 41–54.
[32] O. Meshi, A. Globerson, An alternating direction method for dual MAP LP relaxation, in: Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML PKDD'11, 2011, pp. 470–483.
[33] A. Schwing, T. Hazan, M. Pollefeys, R. Urtasun, Globally convergent dual MAP LP relaxation solvers using Fenchel–Young margins, in: Advances in Neural Information Processing Systems, vol. 25, 2012, pp. 2393–2401.



Jianwu Dong received his B.S. degree in Automation from Central South University in 2010. He is currently a Ph.D. candidate in the Department of Automation at Tsinghua University, China. His research interests cover inference in probabilistic graphical models, machine learning and their applications.

Feng Chen received the B.S. and the M.S. degrees in Automation from Saint-Petersburg Polytechnic University, Russia, in 1994 and 1996, respectively, the Ph.D. degree in Automation Department from Tsinghua University, China, in 2000. He is currently a professor of Tsinghua University. His research interests are mainly in the area of computer vision, video processing and inference in graphical models. Corresponding author of this paper.

Qiang Cheng received his B.S. degree in Automation from Nanjing University in 2009. He is currently a Ph.D. candidate in the Department of Automation at Tsinghua University, China. His research area is machine learning, focusing on problems of inference, learning, decision making and various applications under the framework of graphical models.

Song Wang received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC) in 2002. From 1998 to 2002, he also worked as a research assistant in the Image Formation and Processing Group at the Beckman Institute of UIUC. In 2002, he joined the Department of Computer Science and Engineering at the University of South Carolina, where he is currently an associate professor. His research interests include computer vision, medical image processing, and machine learning. He is currently serving as the Publicity/Web portal Chair of the Technical Committee of Pattern Analysis and Machine Intelligence, IEEE Computer Society and an Associate Editor of Pattern Recognition Letters. He is a member of the IEEE and the IEEE Computer Society.