

---

# An Incremental Subgradient Algorithm for Approximate MAP Estimation in Graphical Models

---

**Jeremy Jancsary**

Austrian Research Institute  
for Artificial Intelligence  
jeremy.jancsary@ofai.at

**Gerald Matz**

Institute of Communications  
and Radio-Frequency Engineering  
Vienna University of Technology

**Harald Trost**

Section for Artificial  
Intelligence of the CeMSIIS  
Medical University of Vienna

## Abstract

We present an incremental subgradient algorithm for approximate computation of maximum-a-posteriori (MAP) states in cyclic graphical models. Its most striking property is its immense simplicity: each iteration requires only the solution of a sequence of trivial optimization problems. The algorithm can be equally understood as a degenerated dual decomposition scheme or as minimization of a degenerated tree-reweighted upper bound and assumes a form that is reminiscent of message-passing. Despite (or due to) its conceptual simplicity, it is equipped with important theoretical guarantees and exposes strong empirical performance.

## 1 Introduction

In recent years, machine learning has given rise to a number of very-large-scale optimization problems. In this regime, conceptually simple algorithms can have an edge over their mathematically involved counterparts, in particular if accuracy is not at a premium. One reason for this phenomenon is that constant overhead matters a lot in the very-large-scale setting. The purpose of this paper is to assess whether the principle of simplicity extends to maximum-a-posteriori (MAP) estimation in cyclic graphical models. In general, this is an NP-hard combinatorial optimization problem (Chandrasekaran et al., 2010). There have been several attempts to solve the task approximately by optimizing a continuous relaxation, most prominently the so-called first-order linear programming (LP) relaxation (Koval and Schlesinger, 1976; Chekuri et al., 2005; Wainwright et al., 2005).

Industrial-strength general purpose solvers can be quite ineffective given the sheer size of the resulting programs (Yanover et al., 2006). Message passing algorithms, on the other hand, can exploit the special problem structure but are still a topic of ongoing research. For instance, the original tree-reweighted message passing algorithm of Wainwright et al. (2005) is not guaranteed to converge at all, while later improvements by Kolmogorov (2006) and a related formulation by Globerson and Jaakkola (2007) establish convergence, but not necessarily to the global optimum (except for binary variables). Consequently, recent work has focused on smoothing a dual formulation of the LP relaxation (Johnson et al., 2007; Jojic et al., 2010) and on obtaining strict convexity in the primal formulation (Ravikumar et al., 2010). These approaches provide global convergence and improved asymptotic convergence rates at the cost of greater complexity. However, it is not immediate that solving a relaxation very accurately should result in better solutions of the discrete problem.

A different approach was taken by Komodakis et al. (2007), who solve a dual decomposition formulation using the projected subgradient algorithm. Global convergence is guaranteed, although at a sublinear rate. At each iteration, their scheme involves max-product belief propagation on spanning trees of the graph. We found that in practice, this comes at a considerable cost. Depending on the structure of a graph, a substantial number of spanning trees can be required in order to cover all edges. The number of dual parameters is exceedingly large in these cases. Moreover, since each iteration requires repeated belief propagation, a significant computational overhead can accumulate.

Interestingly, the dual formulation of [Komodakis et al. \(2007\)](#) is equivalent to minimization of the tree-reweighted upper bounds of [Wainwright et al. \(2005\)](#). Whereas the first authors directly minimize this bound, the latter go on to determine a Lagrangian reformulation and devise message passing algorithms in order to solve it. An important finding in this context is that the choice of spanning trees does not matter as long as all edges are covered with non-zero probability. Moreover, as [Kolmogorov \(2006\)](#) later noted, the trees need not be spanning. We exploit this freedom in the choice of trees to define a lightweight iterative scheme that solves a dual formulation of the first-order LP relaxation. The resulting algorithm is easy to implement, efficient in practice, and guaranteed to converge to the global optimum of the relaxation.

## 2 Preliminaries

We consider undirected graphical models  $G$  with vertex set  $\mathcal{V}$  and edge set  $\mathcal{E}$  defined over discrete random variables with at most pairwise interactions. The potential of a particular variable configuration  $\mathbf{x} \in \mathcal{X}^n$  thus decomposes as

$$P(\mathbf{x}; \boldsymbol{\theta}) = \sum_{s \in \mathcal{V}} \theta_s(x_s) + \sum_{(s,t) \in \mathcal{E}} \theta_{st}(x_s, x_t), \quad (1)$$

where the parameters  $\boldsymbol{\theta} \in \mathbb{R}^d$  (consisting of node potentials  $\theta_s$  and edge potentials  $\theta_{st}$ ) are considered given. Subsequently, we will be concerned with computation of approximations to the maximum-a-posteriori (MAP) value and state,

$$\bar{P}(\boldsymbol{\theta}) = \max_{\mathbf{x} \in \mathcal{X}^n} P(\mathbf{x}; \boldsymbol{\theta}) \quad \text{and} \quad \bar{\mathbf{x}} = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}^n} P(\mathbf{x}; \boldsymbol{\theta}). \quad (2)$$

### 2.1 Tree-reweighted upper bounds

We next discuss the tree-reweighted upper bounds on  $\bar{P}(\boldsymbol{\theta})$  introduced by [Wainwright et al. \(2005\)](#) and show that minimization of these bounds is equivalent to the dual decomposition formulation by [Komodakis et al. \(2007\)](#). Consider the set  $\mathcal{T} = \{T\}$  of all spanning trees of a cyclic graph  $G$ . Each of the spanning trees is associated with a parameterization  $\boldsymbol{\theta}^T$  which is tractable by the structural assumption; that is, the potentials of  $\boldsymbol{\theta}^T$  corresponding to configurations  $x_s$  of vertices and  $(x_s, x_t)$  of edges that do not belong to a particular tree  $T$  are implicitly constrained to be zero. A convex combination  $\sum_T \rho^T \bar{P}(\boldsymbol{\theta}^T)$  over trees then yields a natural upper bound on  $\bar{P}(\boldsymbol{\theta})$  if the tractable parameters  $\{\boldsymbol{\theta}^T\}$  and the distribution over trees  $\boldsymbol{\rho}$  lie in the respective convex sets

$$\mathcal{C}(\boldsymbol{\theta}) = \left\{ \{\boldsymbol{\theta}^T\} \mid \sum_T \rho^T \boldsymbol{\theta}^T = \boldsymbol{\theta} \right\} \quad \text{and} \quad \Delta = \left\{ \boldsymbol{\rho} \mid \sum_T \rho^T = 1, \rho^T \geq 0 \text{ for all } T \right\}.$$

Note that the definition of  $\mathcal{C}(\boldsymbol{\theta})$  implies that each edge  $(s, t)$  must be covered with non-zero probability unless the potentials  $\theta_{st}$  are all zero. The upper bound now follows from Jensen's inequality:

$$\bar{P}(\boldsymbol{\theta}) = \bar{P}(\sum_T \rho^T \boldsymbol{\theta}^T) \leq \sum_T \rho^T \bar{P}(\boldsymbol{\theta}^T).$$

A natural question is then how to obtain the tightest upper bound possible within this framework. For a given distribution  $\boldsymbol{\rho}$  over spanning trees, and given target parameters  $\boldsymbol{\theta}$ , we want to find the minimum over the set of tractable parameterizations  $\{\boldsymbol{\theta}^T\}$ ,

$$\min_{\{\boldsymbol{\theta}^T\} \in \mathcal{C}(\boldsymbol{\theta})} \sum_{T \in \mathcal{T}} \rho^T \bar{P}(\boldsymbol{\theta}^T). \quad (3)$$

This is a convex optimization problem. For any feasible  $\boldsymbol{\rho} \in \Delta$ , the optimum attained in (3) will be the same, the reason being that the Lagrangian duals are all equivalent to the same LP relaxation ([Wainwright et al., 2005](#)). Hence, we can choose  $\boldsymbol{\rho}$  such that most coefficients  $\rho^T$  are zero, whereas the coefficients for a few selected trees needed to cover the edges, which we denote by  $S(\mathcal{T})$ , are equal to a common constant  $\rho = 1/|S(\mathcal{T})|$ . But then, the formulation in (3) reduces to

$$\min_{\{\boldsymbol{\lambda}^T\} \in \mathcal{S}(\boldsymbol{\theta})} \sum_{T \in S(\mathcal{T})} \bar{P}(\boldsymbol{\lambda}^T) \quad \text{with} \quad \mathcal{S}(\boldsymbol{\theta}) = \left\{ \{\boldsymbol{\lambda}^T\} \mid \sum_{T \in S(\mathcal{T})} \boldsymbol{\lambda}^T = \boldsymbol{\theta} \right\}, \quad (4)$$

where we moved the common constant  $\rho$  into the parameters by defining  $\boldsymbol{\lambda}^T = \rho \boldsymbol{\theta}^T$ . Due to the linearity of  $P(\cdot)$ , this changes neither the solution nor the corresponding optimum. But now the equivalence to the formulation obtained by [Komodakis et al. \(2007\)](#) is apparent.

### 3 Approach

One of the principal motivations of [Wainwright et al. \(2005\)](#) in deriving the Lagrangian dual of (3) is the associated reduction in dimensionality; specifically, the number of parameters of the optimization problem is reduced from  $|\mathcal{T}|d$  to  $d$ . However, as we shall point out, a similar dimensionality reduction is possible while maintaining the upper bound formulation in (3)–(4).

#### 3.1 Re-stating the Problem

We start out with the more convenient formulation given by (4). It is apparent that the objective does not depend on  $\rho$ ; moreover, as mentioned before, the optimum is independent of the choice of trees as long as each edge with non-zero potentials  $\theta_{st}$  is covered. Our main idea is now to choose each tree as a single edge, such that  $S(\mathcal{T})$  equals  $\mathcal{E}$ . To make this choice explicit, we will use  $E$  to refer to such a degenerated tree consisting of a single edge from now on. An important consequence of our limitation to single-edge trees is that the edge potentials  $\lambda_{st}^E$  of degenerated tree  $E$  must equal the edge potentials  $\theta_{st}$  of the target parameters  $\theta$ , otherwise we have that  $\{\lambda^E\} \notin \mathcal{S}(\theta)$ . Hence, the parameters  $\lambda_{st}^E$  are fully specified. Moreover, those components of  $\lambda^E$  corresponding to variables that are not part of  $E$  or edges other than  $E$  are implicitly constrained to be zero. Hence, the only remaining parameters of an edge  $E = (s, t)$  are the node parameters  $\lambda_s^E$  and  $\lambda_t^E$ . We conclude that each node  $s$  can be part of several edges whose parameters take the form  $\lambda^E = \{\lambda_s^E, \lambda_t^E\}$ .

The MAP value of each degenerated tree  $E = (s, t)$  is now easily obtained as

$$\bar{P}^E(\lambda^E) = \max_{(x_s, x_t) \in \mathcal{X} \times \mathcal{X}} \{\lambda_s^E(x_s) + \lambda_t^E(x_t) + \theta_{st}(x_s, x_t)\}. \quad (5)$$

Subsequently, we use  $(\bar{x}_s^E, \bar{x}_t^E)$  to denote the edge state<sup>1</sup> which attains the maximum in (5). It can be found by maximizing over  $|\mathcal{X} \times \mathcal{X}|$  sums of three scalar values. In contrast, computation of  $\bar{P}(\lambda^T)$  in (4) requires max-product belief propagation, which is a fairly elaborate procedure. Next, observe that the constraint set simplifies to

$$\mathcal{Q}(\theta) = \left\{ \{\lambda^E\} \mid \sum_{E:s \in E} \lambda_s^E = \theta_s \text{ for all } s \in \mathcal{V} \right\}. \quad (6)$$

By defining  $\lambda = \{\lambda^E\}$  and putting things together, we obtain the final formulation.

#### Optimization Problem 1:

$$\begin{aligned} & \text{minimize} && D(\lambda; \theta) = \sum_{E \in \mathcal{E}} \bar{P}^E(\lambda^E) \\ & \text{subject to} && \lambda \in \mathcal{Q}(\theta). \end{aligned} \quad (7)$$

This new problem is defined in terms of the parameters  $\lambda = \{\lambda^E\} \in \mathbb{R}^{2|\mathcal{X}||\mathcal{E}|}$ , the dimensionality of which is significantly lower than that of  $\{\lambda^T\}$ . In particular, since  $\lambda$  only involves parameters corresponding to node potentials, there is no quadratic dependence on  $|\mathcal{X}|$ . In terms of memory consumption, formulation (7) is thus on par with message passing algorithms. Indeed, as we shall see, each  $\lambda_s^E$  can be understood to carry messages between a node  $s$  and its containing edge  $E$ . Furthermore, comparing (7) and (4), we observe that the new objective replaces  $|S(\mathcal{T})|$  elaborate optimization problems by a large number  $|\mathcal{E}|$  of primitive optimization problems.

#### 3.2 Minimization of the Upper Bound

We next inspect the mathematical properties of (7). While we have a convex optimization problem, the objective function  $D(\lambda; \theta)$  is non-differentiable. Nonetheless, one can obtain a subgradient  $\mathbf{g} \in \mathbb{R}^{2|\mathcal{X}||\mathcal{E}|}$  with respect to  $\lambda$ . Its components are given by

$$g_s^E(x_s) = [x_s = \bar{x}_s^E], \quad g_t^E(x_t) = [x_t = \bar{x}_t^E] \quad \text{for all } E = (s, t), x_s, x_t, \quad (8)$$

where  $\bar{x}_s^E$  and  $\bar{x}_t^E$  belong to the edge MAP state  $(\bar{x}_s^E, \bar{x}_t^E)$  that maximizes (5) for edge  $E$ , and  $[\cdot]$  evaluates to 1 if the condition inside the bracket is true and 0 otherwise. This subgradient is trivially bounded since we have  $\|g^E\|_2^2 = 2$  for all  $E$ , and hence  $\|\mathbf{g}\|_2^2 = 2|\mathcal{E}|$ .

<sup>1</sup>To facilitate presentation, we will assume that the maximum is attained by exactly one edge state  $(\bar{x}_s^E, \bar{x}_t^E)$ .

```

Input : Graph  $G$ , target parameters  $\theta$ , initial feasible point  $\lambda$ 
Output: Feasible primal solution  $\tilde{x}$  that is an approximation to  $\bar{x}$ 
choose initial feasible primal solution  $\tilde{x}$  arbitrarily ;
repeat
  pick next step size  $\alpha$  and shuffle the set of edges  $\mathcal{E}$ ;
  foreach  $E = (s, t) \in \mathcal{E}$  do
    find MAP state:  $(\bar{x}_s^E, \bar{x}_t^E)$ ;
    subtract scaled subgradient:  $\lambda_s^E(\bar{x}_s^E) \leftarrow \lambda_s^E(\bar{x}_s^E) - \alpha, \lambda_t^E(\bar{x}_t^E) \leftarrow \lambda_t^E(\bar{x}_t^E) - \alpha$ ;
    foreach  $E' \in \mathcal{E}_s$  do project:  $\lambda_s^{E'}(\bar{x}_s^E) \leftarrow \lambda_s^{E'}(\bar{x}_s^E) + \alpha/|\mathcal{E}_s|$ ;
    foreach  $E' \in \mathcal{E}_t$  do project:  $\lambda_t^{E'}(\bar{x}_t^E) \leftarrow \lambda_t^{E'}(\bar{x}_t^E) + \alpha/|\mathcal{E}_t|$ ;
  foreach  $s \in \mathcal{V}$  do construct candidate  $\tilde{c}$ :  $\tilde{c}_s \leftarrow$  choose at random from  $\{\bar{x}_s^E \mid E \in \mathcal{E}_s\}$ ;
  if  $P(\tilde{c}; \theta) > P(\tilde{x}; \theta)$  then
    accept best primal solution so far:  $\tilde{x} \leftarrow \tilde{c}$ ;
  if  $D(\lambda; \theta) = P(\tilde{x}; \theta)$  then
    optimal primal solution found: return  $\tilde{x}$ ;
until converged ;
approximate primal solution found: return  $\tilde{x}$ ;

```

Algorithm 1: INCMP – Incremental subgradient algorithm for Optimization Problem 1

Consider now the constraint set  $\mathcal{Q}(\theta)$ . It turns out that there is an efficient way of projecting an infeasible point  $\lambda'$  onto this set. Formally, we search the solution to the following problem:

$$\mathcal{P}_\theta(\lambda') = \underset{\lambda \in \mathcal{Q}(\theta)}{\operatorname{argmin}} \|\lambda - \lambda'\|_2^2. \quad (9)$$

It is easily seen that among the admissible  $\{\lambda_s^E(x_s)\}$  for a given variable  $s$  and state  $x_s$ , which must sum to  $\theta_s(x_s)$ , the sum of squares is minimized if  $(\lambda_s^E(x_s) - \lambda_s^{E'}(x_s))^2$  is equal for all containing edges  $\mathcal{E}_s = \{E \mid s \in E\}$ . Hence, the optimal projection in the sense of (9) is given by

$$\mathcal{P}_\theta(\lambda') = \left\{ \lambda_s^E(x_s) \leftarrow \lambda_s^{E'}(x_s) - \left( \sum_{E' \in \mathcal{E}_s} \lambda_s^{E'}(x_s) - \theta_s(x_s) \right) / |\mathcal{E}_s| \text{ for all } E, s \in E, x_s \right\}.$$

Equivalently, after each modification of a component  $\lambda_s^E(x_s)$  of a feasible point, feasibility can be restored by distributing the amount of change uniformly over all components  $\{\lambda_s^{E'}(x_s) \mid E' \in \mathcal{E}_s\}$ .

Equipped with efficient ways of computing the subgradient and projecting onto the feasible set, we could use the projected subgradient algorithm to solve Optimization Problem 1, analogously to [Komodakis et al. \(2007\)](#). However, our problem differs from theirs in that the number of component functions can be expected to be significantly larger. Hence, the incremental subgradient algorithm ([Nedić and Bertsekas, 2001](#)) is an attractive option. Here, at each inner iteration, only the subgradient  $g^E$  of a single component function  $\bar{P}^E(\cdot)$  is subtracted, after which feasibility is restored using projection. The subgradient of the next component function is then computed using the adapted parameters. When the parameters of the component functions overlap or are coupled through the constraints, this can result in significantly faster initial convergence.

Algorithm 1 outlines the application of this method to Optimization Problem 1. Each component subgradient  $g^E$  is very sparse in our case. Only two indices are ever non-zero, namely those corresponding to the variable states  $\bar{x}_s^E$  and  $\bar{x}_t^E$  of the edge MAP state that maximizes (5) for edge  $E$ . Consequently, each inner update only affects a small number of parameters. The parameters of edges other than  $E$  are affected through the projection step, which only involves adjacent edges. Hence, the structure of the a graph determines how quickly parameter updates propagate through the graph, which mirrors the situation in message passing algorithms.

Two choices impact the convergence behavior of Algorithm 1 significantly. The first one is the order in which edges  $E$  are selected for the inner updates. We found that a random update order (implemented using a Fisher-Yates shuffle) consistently gives good results over a variety of graph structures. This is also supported by findings of [Nedić and Bertsekas \(2001\)](#), who give improved convergence rates for updates in random order.

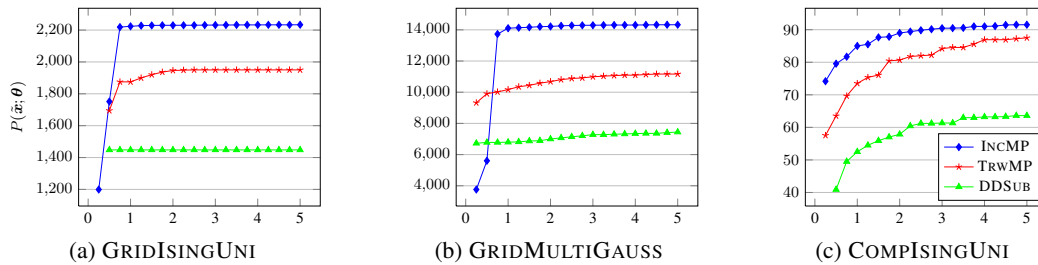


Figure 1: Best primal solution found by the solvers as a function of time (seconds)

The second choice concerns the sequence of step sizes  $\{\alpha^{(k)}\}$ . A variety of sequences are known for which convergence to the global optimum is guaranteed; however, these can be rather slow in practice. We implemented the following practical variant: Initially,  $\alpha^{(0)}$  is set to the sample standard deviation of the potentials in  $\theta$ . At each outer iteration, if  $D(\cdot)$  has decreased, we opt for a moderate decrease, say,  $\alpha^{(k+1)} = 0.95\alpha^{(k)}$ ; otherwise, we decrease aggressively, for instance  $\alpha^{(k+1)} = 0.5\alpha^{(k)}$ . Once  $\alpha^{(k)}$  drops below a tiny number  $\varepsilon$  in iteration  $k$ , we adopt a static schedule and choose  $\alpha^{(k')} = \varepsilon/(k' - k)$  in subsequent iterations  $k' = k + 1, k + 2, \dots$  of the algorithm.

**Proposition 1.** *With the sequence of step sizes  $\{\alpha^{(k)}\}$  chosen as described above, Algorithm 1 converges to the global optimum of Optimization Problem 1 as  $k$  approaches infinity.*

*Proof (Sketch).* Initially, the sequence of step sizes  $\{\alpha^{(k)}\}$  decreases such that we reach  $\varepsilon$  in a finite number of iterations. Subsequently,  $\{\alpha^{(k)}\}$  is chosen as a nonsummable diminishing sequence, which guarantees global convergence for bounded subgradients (Nedić and Bertsekas, 2001).  $\square$

### 3.3 Obtaining Primal Solutions

In general, we do not have strong duality between (2) and (7). Hence, it is not always possible to extract the exact MAP state  $\tilde{x}$  from an optimal solution  $\lambda$  of Optimization Problem 1. However, “good” feasible points can be expected to be obtained from the edge MAP states  $(\tilde{x}_s^E, \tilde{x}_t^E)$ . Specifically, at each iteration of Algorithm 1, for each node  $s$ , we choose the component  $\tilde{c}_s$  of a candidate primal solution uniformly at random from the set  $\{\tilde{x}_s^E \mid E \in \mathcal{E}_s\}$ . Those MAP states that appear in more edges adjacent to  $s$  thus have a higher chance of being picked. We keep track of the best primal solution  $\tilde{x}$  generated so far. In some cases, a certificate of optimality can be obtained for  $\tilde{x}$ .

**Proposition 2.** *Assume that at a given outer iteration of Algorithm 1, we have  $P(\tilde{x}; \theta) = D(\lambda; \theta)$ . It then follows that  $\lambda$  minimizes  $D(\cdot)$  and  $\tilde{x}$  maximizes  $P(\cdot)$ . This happens precisely if for each node  $s$ , the edge MAP states  $\{(\tilde{x}_s^E, \tilde{x}_t^E) \mid E \in \mathcal{E}_s\}$  all agree on a common node MAP state  $\tilde{x}_s^E$ .*

*Proof (Sketch).* By construction,  $P(\tilde{x}; \theta)$  gives a lower bound on  $\bar{P}(\theta)$ , whereas  $D(\lambda; \theta)$  gives an upper bound. For the bounds to coincide,  $\tilde{x}$  and  $\lambda$  must both be optimal. Agreement of the edge MAP states at the joint optimum follows from Wainwright et al. (2005, Proposition 1).  $\square$

## 4 Experiments

We generated three random graphs of varying structure and potentials  $\theta$ . First, GRIDISINGUNI was a  $50 \times 50$  grid with binary variables ( $\mathcal{X} = \{-1, +1\}$ ) and potentials given by  $\theta_s(x_s) = \gamma x_s$  and  $\theta_{st}(x_s, x_t) = \gamma x_s x_t$  with  $\gamma \sim \mathcal{U}(-1, +1)$  drawn independently for each node and edge. Second, GRIDMULTIGAUSS was a  $20 \times 20$  grid with variables of arity  $|\mathcal{X}| = 16$  and potentials chosen as  $\theta_s(x_s) = 0$  and  $\theta_{st}(x_s, x_t) \sim \mathcal{N}(0, 15)$  independently. Finally, COMPISINGUNI was a complete graph of 50 binary variables with potentials chosen akin to GRIDISINGUNI.

Subsequently, we compared Algorithm 1 (INCMP) to our implementations of two competing algorithms. We did not tune INCMP individually for each graph, but rather used the general step size schedule described in section 3.2. By construction, a choice of trees is not required by INCMP. For the dual decomposition scheme (DDSUB; Komodakis et al., 2007), we used a greedy algorithm to establish small sets of trees covering all edges and obtained the primal solutions similarly to INCMP. The step size schedule we used was similar to the one presented here and performed well

in previous experiments. For tree-reweighted message passing (TRWMP; Wainwright et al., 2005), we obtained edge occurrence probabilities analogously to DDSUB and constructed primal solutions from the maximizers of the node beliefs at each iteration. The messages were updated by iterating over factors in random order, akin to INCMP. For each graph and solver, we plotted the best primal function value found as a function of running time, averaged over 20 runs. We excluded the time for generation of the set of trees needed by TRWMP and DDSUB.

Figure 1 shows that Algorithm 1 (INCMP) dominates its competitors on the three graphs discussed above. Interestingly, all algorithms were able to minimize the dual  $D(\cdot)$  very effectively, but the quality of primal solutions found by the methods varied significantly. In particular, DDSUB suffered from this phenomenon. One explanation is that the LP relaxation need not be very tight for graphs of substantial size and complexity. In this regime, the low per-iteration cost of INCMP allows for guided construction and evaluation of a large number of candidate solutions, which clearly pays off.

## 5 Conclusion and Outlook

We derived an efficient algorithm for approximate MAP estimation in cyclic graphical models that is reminiscent of message passing. It is characterized by the following properties: (a) guaranteed convergence to the global optimum of the first-order LP relaxation of the MAP problem; (b) by construction, we obtain both an upper bound and a lower bound on the exact MAP value; (c) if the LP relaxation is tight, the bounds coincide and we obtain the exact MAP state; (d) the memory requirements are equal to those of belief propagation. In future work, we would like to employ the algorithm as the computational core in a branch-and-bound scheme. We believe that the above properties, along with the potential for warm-starting, render it an attractive choice in this setting.

### Acknowledgments

We thank our reviewers. OFAI acknowledges support by the Austrian Federal Ministry for Transport, Innovation, and Technology and by the Austrian Federal Ministry for Science and Research. JJ is supported by FIT-IT Grant 819567. GM acknowledges funding by FWF Grant N10606.

### References

- V. Chandrasekaran, N. Srebro, and P. Harsha. Complexity of inference in graphical models, May 2010. URL [http://ssg.mit.edu/~venkate/csh\\_compinf\\_preprint10.pdf](http://ssg.mit.edu/~venkate/csh_compinf_preprint10.pdf). Preprint.
- C. Chekuri, S. Khanna, J. Naor, and L. Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Mathematics*, 18(3):608–625, 2005.
- A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *Advances in Neural Information Processing Systems (NIPS)*, 2007.
- J. K. Johnson, D. M. Malioutov, and A. S. Willsky. Lagrangian relaxation for MAP estimation in graphical models. In *45th Annual Allerton Conference on Communication, Control and Computing*, 2007.
- V. Jojic, S. Gould, and D. Koller. Accelerated dual decomposition for MAP inference. In *27th International Conference on Machine Learning (ICML)*, 2010.
- V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(10):1568–1583, 2006.
- N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *IEEE 11th International Conference on Computer Vision (ICCV)*, pages 1–8, 2007.
- V. K. Koval and M. I. Schlesinger. Two-dimensional programming in image analysis problems. *Automatics and Telemekhanics*, 8:149–168, 1976. In Russian.
- A. Nedić and D. P. Bertsekas. Incremental subgradient methods for nondifferentiable optimization. *SIAM Journal on Optimization*, 12:109–138, 2001.
- P. Ravikumar, A. Agarwal, and M. J. Wainwright. Message-passing for graph-structured linear programs: proximal methods and rounding schemes. *Journal of Machine Learning Research*, 11:1043–1080, 2010.
- M. J. Wainwright, T. S. Jaakkola, and A. S. Willsky. MAP estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Transactions on Information Theory*, 51(11):3697–3717, 2005.
- C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation – an empirical study. *Journal of Machine Learning Research*, 7:1887–1907, 2006.