Tighter Relaxations for MAP-MRF Inference: A Local Primal-Dual Gap based Separation Algorithm

Dhruv Batra
TTI-Chicago

Sebastian Nowozin
Microsoft Research Cambridge

Pushmeet Kohli
Microsoft Research Cambridge

Abstract

We propose an efficient and adaptive method for MAP-MRF inference that provides increasingly tighter upper and lower bounds on the optimal objective. Our method starts by solving the first-order LOCAL(G) linear programming relaxation. This is followed by an adaptive tightening of the relaxation where we incrementally add higher-order interactions to enforce proper marginalization over groups of variables. Computing the best interaction to add is an NP-hard problem. We show good solutions to this problem can be readily obtained from "local primal-dual gaps" given the current primal solution and a dual reparameterization vector. This is not only extremely efficient, but in contrast to previous approaches, also allows us to search over prohibitively large sets of candidate interactions to add. We demonstrate the superiority of our approach on MAP-MRF inference problems encountered in computer vision.

1 Introduction

A number of problems in computer vision, computational biology and machine learning are naturally formulated as discrete labelling problems. Markov Random Fields (MRFs) (Wainwright and Jordan, 2008; Koller and Friedman, 2009) provide a principled framework for modelling and solving these problems. Maximum *a posteriori* (MAP) inference in MRFs is thus of fundamental importance in these domains but known to be NP-hard in general (Shimony, 1994).

LP Relaxations. Linear Programming (LP) relaxations of MAP-MRF are powerful techniques that have have been independently (re-)discovered by different communities in pattern recognition (Schlesinger, 1976), constraint satisfaction (Koster et al., 1998; Chekuri et al., 2004) and machine learning (Wainwright et al., 2005) (See Werner (2007) for a review). From a polyhedral perspective, LP relaxations for MAP correspond to approximating the intractable *marginal polytope* with an outer-bound, called the *local polytope* (Wainwright and Jordan, 2008).

Unfortunately, as various authors have noted (Meltzer et al., 2005; Kolmogorov, 2006; Yanover et al., 2006; Sontag et al., 2008b; Komodakis and Paragios, 2008; Werner, 2008) the standard LP relaxation is rarely tight on real applications. This motivates the need for cluster-based LPs (Sontag et al., 2008b,a), where local consistency is enforced on clique or cluster marginals. The relaxation becomes tighter as larger cliques are added, but since solving the clique-relaxation is exponential in the size of the clique, the problem of finding the *right* cliques to add, also referred to as *cluster-pursuit*, becomes a critical ones.

Cluster Pursuit. Most approaches for cluster-pursuit can be understood as *Dictionary-Enumeration* techniques. They define a small dictionary of clusters and enumerate over all members in this dictionary to add the member with the highest "score", which is typically a measure of the benefit of adding this cluster to the relaxation. For example, Sontag et al. (2008b) define their dictionary to be all 3-cycles in a graph (or 4-cycles for grids); Werner (2008) consider all 4-cycles for grids and Komodakis and Paragios (2008) consider all 1x1, 1x2, 2x1-cycles for grids.

Separation Problem and Hardness Result. Let $\mathcal{F} = \{A\}$ denote a dictionary of clusters A, and $w(A)$ denote the score of cluster A. Then, the cluster-pursuit problem can be written as:

$$A^* = \operatorname*{argmax}_{A \in \mathcal{F}} w(A) \tag{1}$$

Ideally, we would like $w(A)$ to be the gain in the tightness of the relaxation due to the addition of A. However, for a general cluster A, this value is hard to compute without actually adding A to the relaxation

Preliminary work. Under review by AISTATS 2010. Do not distribute.

and re-solving, defeating the very purpose of cluster-pursuit. Therefore dictionary-enumeration methods typically choose $w(A)$ to be a lower-bound on the improvement (Sontag et al., 2008b), and enumerate over small cycles in graphs. However, Sontag (2010) has shown that even for this "easier" scoring function, the separation problem (1) is NP-hard for arbitrary length cycles in the graph for general MRFs. In the view of this hardness result, it becomes important to look for surrogate or approximate scoring functions that are quickly computable and allow for efficient search.

Contribution. In this work we propose an approximate separation algorithm for the marginal polytope based on a novel cluster-scoring function we call, *Local Primal-Dual Gap (LPDG)*. LPDG is a generalization of the complementary slackness condition in the primal-dual programs of the LP relaxations of MAP-MRF. Intuitively, LPDG quantifies by how much the complementary slackness condition is violated, and attributes violations to individual potentials. Moreover, LPDG can be computed as a byproduct in dual message-passing algorithms, essentially *for free*, and also enables search over arbitrary length cycles in a graph, which cannot be done via lower-bound based scoring functions (Sontag, 2010). We demonstrate improved performance in two distinct regimes: for the case of having a small dictionary of possible clusters – such as all small cycles – we show improved overall performance with respect to time; and for the case of prohibitively large classes of clusters we show performance on hard, previously unsolvable instances.

In the following we first introduce the basic concepts in Section 2. Our novel contribution, the local primal-dual gap and a separation algorithm are discussed in Section 3. The performance of the proposed algorithm is demonstrated experimentally in Section 4 before we conclude in Section 5.

2 LP relaxation of MAP-MRF

Notation. We try to closely follow the notation of Werner (2008). For any positive integer n, let $[n]$ be shorthand for the set $\{1, 2, \ldots, n\}$. We consider a set of discrete random variables $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$, each taking value in a finite label set, $x_i \in X_i$. For a set $A \subseteq [n]$, we use x_A to denote the tuple $\{x_i \mid i \in A\}$, and X_A to be the joint label space $\times_{i \in A} X_i$. Let $G = (\mathcal{V}, \mathcal{E})$ be a hypergraph defined over these variables (*i.e.*, $\mathcal{V} = [n]$, $\mathcal{E} = \{A \mid A \subseteq \mathcal{V}\}$), and let $\theta_A : X_A \to \mathbb{R}$ be a function defining the cost/energy at each hyperedge for the labelling of variables in scope.

The goal of MAP inference can then be succinctly written as: $\min_{\mathcal{X} \in X_\mathcal{V}} \sum_{A \in \mathcal{E}} \theta_A(x_A)$. For the special case of pairwise MRFs, G is a simple graph (with no hyper-edges), and the energy can be written as: $\min_{\mathcal{X} \in X_\mathcal{V}} \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i, x_j)$.

Schlesinger's LP. It is well-known (Wainwright and Jordan, 2008) that the above discrete optimization problem can be written as a linear programming problem over the so-called marginal polytope $\mathcal{M}(G)$:

$$\min_{\boldsymbol{\mu} \in \mathcal{M}(G)} \boldsymbol{\theta} \cdot \boldsymbol{\mu} \tag{2}$$

$$\mathcal{M}(G) = \left\{ \boldsymbol{\mu} \;\middle|\; \exists p(\mathcal{X}) \; s.t. \; \begin{array}{c} \mu_i(x_i) = \sum_{X_{\mathcal{V} \setminus i}} p(\mathcal{X}) \\ \mu(x_i, x_j) = \sum_{X_{\mathcal{V} \setminus i,j}} p(\mathcal{X}) \end{array} \right\},$$

where $\boldsymbol{\theta} \cdot \boldsymbol{\mu} \doteq \sum_{i \in \mathcal{V}} \theta_i(x_i)\mu_i(x_i) + \sum_{(i,j) \in \mathcal{E}} \theta_{ij}(x_i, x_j)\mu_{ij}(x_i, x_j)$, $p(\mathcal{X})$ is a Gibbs distribution that factorizes over the graph G, and $\boldsymbol{\mu}$ is a vector holding node and edge marginal distributions, *i.e.*, $\boldsymbol{\mu} = \{\mu_i(\cdot), \mu_e(\cdot) \mid i \in \mathcal{V}, e \in \mathcal{E}\}$.

Problem (2) is NP-hard to solve in general and the marginal polytope cannot be characterized with a polynomial number of inequalities (Wainwright and Jordan, 2008). The standard LP *relaxation* of the MAP problem (2), also known as Schlesinger's bound (Schlesinger, 1976; Werner, 2007), is given by $\min_{\boldsymbol{\mu} \in \mathcal{L}(G)} \boldsymbol{\theta} \cdot \boldsymbol{\mu}$, with

$$\mathcal{L}(G) = \left\{ \boldsymbol{\mu} \geq 0 \;\middle|\; \begin{array}{c} \sum_{x_i} \mu_i(x_i) = 1, \\ \sum_{x_i, x_j} \mu_{ij}(x_i, x_j) = 1, \\ \sum_{x_i} \mu_{ij}(x_i, x_j) = \mu_j(x_j) \end{array} \right\}, \tag{3}$$

where $\boldsymbol{\mu}$ is now a vector of local *beliefs*, which are forced to be "edge-consistent", meaning that the marginalized edge beliefs agree with the node beliefs (known as marginalization constraints). Optimizing over (3) is possible in polytime, and by $\mathcal{L}(G) \supseteq \mathcal{M}(G)$ this provides a lower bound on the optimal objective. Unfortunately, this edge-consistent relaxation is guaranteed to be tight only for special classes of graphs and energies (Wainwright and Jordan, 2008).

Tighter LPs. In practice, this LP is rarely tight, and various authors have suggested improved relaxations in the form of cluster-based LPs which involve beliefs over cliques μ_A. Cluster-based LPs generalize marginalization constraints from (3) to cliques as follows: Let $J = \{(A, B, x_B) \mid B \subset A \subseteq [n]\}$ be a collection of triplets (called *pencil*). Although the members of J are triplets (A, B, x_B), with a slight abuse of notation we use $A \in J$ to mean there exists a pencil/triplet in J containing A, *i.e.*, $A \in J \doteq \{A \mid \exists (A, B, x_B), (Z, A, x_A) \in J\}$.

A cluster-based local polytope $\mathcal{L}(J)$ is defined as:

$$\mathcal{L}(J) = \left\{ \boldsymbol{\mu} \geq 0 \;\middle|\; \begin{array}{c} \sum_{x_A} \mu_A(x_A) = 1, \; A \in J, \\ \sum_{x_{A \setminus B}} \mu_A(x_A) = \mu_B(x_B), \\ (A, B, x_B) \in J \end{array} \right\}. \tag{4}$$

Clearly, the local polytope $\mathcal{L}(G)$ in (3) is a special case of $\mathcal{L}(J)$ for $J = \{(A, B, x_B) \mid A \in \binom{[n]}{2} \cap \mathcal{E}, B \subset A\}$,

$$\begin{array}{ll|ll|l}
\min\limits_{\boldsymbol{\mu}} & \boldsymbol{\theta}\cdot\boldsymbol{\mu} & \max\limits_{\boldsymbol{h},\boldsymbol{y}} & \mathbf{1}\cdot\boldsymbol{h} & \\
\text{sb.t.} & \sum\limits_{x_A}\mu_A(x_A)=1, & \text{sb.t.} & h_A\in\mathcal{R}, & \forall A\in J\\
& \sum\limits_{x_{A\setminus B}}\mu_A(x_A)=\mu_B(x_B), & & y_{A\to B}(x_B)\in\mathcal{R}, & \forall(A,B,x_B)\in J\\
& \mu_A(x_A)\geq 0. & & h_A\leq\theta_A(x_A)+\sum\limits_{Z|(Z,A,x_A)\in J}y_{Z\to A}(x_A)-\sum\limits_{B|(A,B,x_B)\in J}y_{A\to B}(x_B), & \forall A\in J, x_A.
\end{array}$$

Figure 1: Cluster-based linear programming relaxation $\mathcal{L}(J)$. Both the primal and dual LP are shown.

where $\binom{[n]}{2}$ is the set of all 2-element subsets of $[n]$. The quality of the relaxation is determined by the choice of J. The larger J is, the better the relaxation will be; in the extreme case of $J=\{(A,B,x_B)\mid A\in 2^{\mathcal{V}}, B\subset A\}$, we have $\mathcal{L}(J)|_{\mathcal{V},\mathcal{E}}=\mathcal{M}(G)$, that is, the *projection* of the local polytope onto the vertex and edge marginals yields the true marginal polytope (Werner, 2008).

We are now faced with conflicting goals. In order to have a strong relaxation, we would like the set J to be as large as possible. However, solving the relaxation over $L(J)$ has exponential dependence in the size of the largest clique in J. Thus, we follow the approach of Sontag et al. (2008b): we start with Schlesinger's LP relaxation and then incrementally add elements to it that will improve the bound.

Search Problem. As defined in (1), this search problem may be written as: $\text{argmax}_{A\in\mathcal{F}}\, w(A)$, where a natural choice for $\mathcal{F}$ is the set of all cliques with clique-width smaller than a parameter m, *i.e.*, $\mathcal{F}=\{A\mid A\subset[n], |A|\leq m\}$. Ideally, we would like to define the weight of a clique as the gain in the relaxation value caused by adding marginalization constraints over A and its subsets to J, *i.e.*, $w^*(A)=\min_{\boldsymbol{\mu}\in\mathcal{L}(J\cup A)}\boldsymbol{\theta}\cdot\boldsymbol{\mu}-\min_{\boldsymbol{\mu}\in\mathcal{L}(J)}\boldsymbol{\theta}\cdot\boldsymbol{\mu}$, where we use $J\cup A$ to denote the set $J\cup\{(A,B,x_B)\mid B\subset A\}$. However, for this ideal definition, even *computing* the weight of a clique requires solving the tightened LP, defeating the very purpose of incremental cluster-pursuit.

The next section describes our proposed weight function, which is both quickly computable and allows for (approximate) fast search over cliques.

3 Local Primal-Dual Gap

We propose a novel cluster-scoring function that we call *Local Primal-Dual Gap (LPDG)*. Before we define LPDG, we consider the dual-LP of the cluster-based LP. Figure 1 shows both the primal and dual programs corresponding to the cluster-based polytope $\mathcal{L}(J)$.

Reparameterization. Given dual-feasible vectors $\{h_A, y_{A\to B}\}$, a *reparameterized cost vector* $\tilde{\theta}_A(\cdot)$ is defined as:

$$\tilde{\theta}_A(x_A)\doteq\theta_A(x_A)+\sum_{Z|(Z,A,x_A)\in J}y_{Z\to A}(x_A)-\sum_{B|(A,B,x_B)\in J}y_{A\to B}(x_B). \quad (5)$$

This is a generalization of the reparameterization operations discussed in (Kolmogorov, 2006; Sontag and Jaakkola, 2009). Intuitively, the $\tilde{\theta}_A(x_A)$ can be thought of as the new cost for state x_A at hyperedge A after incorporating the dual variables $\{y_{Z\to A}, y_{A\to B}\}$. Energy is preserved under reparameterization, *i.e.*, for any labeling $\mathcal{X}$, we have $\sum_{A\in\mathcal{E}}\tilde{\theta}_A(x_A)=\sum_{A\in\mathcal{E}}\theta_A(x_A)$. Note, however, that this does not imply that $\tilde{\theta}_A(x_A)=\theta_A(x_A)$.

Complementary Slackness. Let $\{\mu_A^*\}$ and $\{h_A^*, y_{A\to B}^*\}$ be a pair of optimal primal and dual solutions of the LPs in Figure 1. The complementary slackness condition (Bertsimas and Tsitsiklis, 1997) for this pair is given by:

$$\mu_A^*(x_A)\cdot\left(\theta_A(x_A)+\sum_{Z|(Z,A,x_A)\in J}y_{Z\to A}^*(x_A)-\sum_{B|(A,B,x_B)\in J}y_{A\to B}^*(x_B)-h_A^*\right)=0. \quad (6)$$

We can re-write the complementary slackness in terms of the reparameterized vector as follows:

$$\mu_A(x_A)\cdot\left(\tilde{\theta}_A(x_A)-\min_{\hat{x}_A}\tilde{\theta}_A(\hat{x}_A)\right)=0. \quad (7)$$

We now define the LPDG for a hyperedge as a generalization of this complementary slackness condition:

Definition (Local Primal-Dual Gap). *Given a reparameterized cost vector $\tilde{\theta}$ and a (possibly non-optimal) integral primal labelling $\mathcal{X}^p=\{x_1^p, x_2^p, \ldots, x_n^p\}$, the* local primal-dual gap *$l(A)$ for each hyperedge $A\in J$ is:*

$$l(A)=\underbrace{\tilde{\theta}_A(x_A^p)}_{\text{primal}}-\underbrace{\min_{x_A}\tilde{\theta}_A(x_A)}_{\text{dual}}. \quad (8)$$

Comparing (7) and (8), we can see that intuitively LPDG quantifies by how much the complementary slackness condition is violated, and attributes violations to individual potentials. More precisely, we state the following properties of LPDG:

Proposition 1 *Non-negativity: $l(A)\geq 0$, $\forall A$.*

Proposition 2 *Distributed Primal-Dual Gap: If $P=\sum_{A\in\mathcal{E}}\theta_A(x_A^p)$ is the energy of the primal labelling, and $D=\mathbf{1}\cdot\boldsymbol{h}$ the current dual value, then LPDG for all cliques sums to the integrality gap,* i.e., *$\sum_{A\in\mathcal{E}}l(A)=P-D$.*

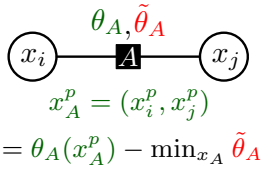


Figure 2: Illustration of the quantities used for computing the local primal-dual gap $l(A)$ for a hyperedge A. The primal contribution θ_A is evaluated at the primal integral iterate $x_A^p = (x_i^p, x_j^p)$. The dual part of the LPDG is computed by locally minimizing the reparameterized cost $\tilde{\theta}_A(x_A)$.

Proposition 3 *Slackness: If LPDG for the best clique is zero,* i.e. $\max_{A \in J} l(A) = 0$, *then complementary slackness conditions hold and thus the LP over* $\mathcal{L}(J)$ *is tight,* i.e. *we have solved the problem.*

Proof Sketches. Prop. 1 follows directly from the definition (8). Prop. 2 just utilizes that $h_A = \min_{x_A} \tilde{\theta}(x_A)$ (for the dual) and conservation of total energy under reparameterization (for the primal). Prop. 3 follows from complementary slackness.

Most interesting from a computational perspective is the fact that for each clique A, LPDG $l(A)$ can be computed essentially *for free* from standard message-passing algorithms, as we already have access to dual variables or messages.

3.1 LPDG-based Cluster-Pursuit

We now describe our approach for finding clusters based on LPDG scores. Recall that our goal is to find a cluster A from a family of clusters $\mathcal{F}$ that maximizes a cluster-scoring function, *i.e.*, $\operatorname{argmax}_{A \in \mathcal{F}} \omega(A)$. Also recall that the ideal cluster scoring function $\omega^*(A)$ is intractable as it requires actually solving the tightened LP to find out the usefulness of A. We define a surrogate cluster scoring function $\omega^l(A)$, which is based on LPDG:

$$\omega^l(A) = \sum_{B \subset A | B \in J} l(B). \tag{9}$$

Our proposed function is simply the sum of the local primal-dual gaps of subsets of A that are already in our relaxation.

Example 1: Triplet Search of Sontag et al. (2008b). If we consider a pairwise MRF, solve Schesinger's LP and look for triplet cliques to add, then $\mathcal{F} = \{(i,j,k) \mid (i,j),(j,k),(k,i) \in \mathcal{E}\}$ and our proposed scoring function would be: $\omega^l(\{i,j,k\}) = l(i)+l(j)+l(k)+l(i,j)+l(j,k)+l(k,i)$.

Example 2: Square Search of Werner (2008). If we consider a grid MRF and search for 4-cycles, $\mathcal{F} = \{(a,b,c,d) \mid (a,b),(b,c),(c,d),(d,a) \in \mathcal{E}\}$ and our proposed scoring function would be: $\omega^l(\{a,b,c,d\}) = l(a)+l(b)+l(c)+l(d)+l(a,b)+l(b,c)+l(c,d)+l(d,a)$.

Comparison with Sontag et al. (2008b). Sontag et al. (2008b) define the weight $w^s(A)$ of a clique A to be the improvement in the relaxation achieved by adding A and passing one round of messages (as opposed to passing messages till convergence). This can be shown to be a lower-bound on the ideal scoring function, $\omega^s(A) \leq \omega^*(A)$. In contrast our LPDG-based scoring function $w^l(A)$ is neither a lower-bound nor an upper-bound on $w^*(A)$ (we have empirically observed violations of both), but rather is simply well-correlated with the ideal score in practice. More importantly, LPDG is significantly more efficient to compute than $w^s(A)$. Concretely, using the triplet search in a pairwise MRF example, let $A = \{i,j,k\}$ be a triplet-clique, and e be an edge.

$$w^s(A) = \sum_{e \in A} \max_{x_e} b_e(x_e) - \max_{x_A} \left[\sum_{e \in A} b_e(x_e) \right], \tag{10}$$

where b_e is the equivalent of our $\tilde{\theta}_{ij}$. The exact form of b_e is not important for this discussion. The important point is that computing the second part of $\omega^s(A)$ takes $O(k^3)$ time, while computing LPDG-based $w^l(A)$ takes $O(k^2)$ time, where k is the number of labels. Moreover, in the case of LPDG, computations done on an edge can be shared for all triplets involving that edge, but for $w^s(A)$ this cannot be done, thus the actual comparison is $O(n^3 \cdot k^3)$ vs $O(n^3 + k^2)$. More generally, for an arbitrary family of clusters $\mathcal{F}$, the score computation time for each cluster will be $O(k^{|A|})$ for Sontag et al. (2008b) vs $O(k^{|B|})$ for us ($|B|$ will always be smaller than $|A|$). As we show in our experiments, these savings can result in significant speed-ups.

Algorithm. Our cluster-pursuit algorithm may be described as follows: We start by solving Schlesinger's LP. We then consider a family $\mathcal{F}$ of clusters to add. For small families like all triplets in a graph or 4-cycles in a grid, we follow the dictionary-enumeration paradigm and compute LPDG-based cluster score for each member of $\mathcal{F}$. We then add the cluster with the highest score of the relaxation $\mathcal{L}(J)$ and re-solve the relaxation (by "warm starting" the the previous solution). For larger families, where dictionary-enumeration is impractical, we use a greedy search algorithm that constructs a "heavy" cluster by incrementally adding parts (nodes, edges) with the highest local-primal dual gap. MAX-Clique is an NP-hard problem and even approximating the clique number within a factor of $O(n^{1-\epsilon})$ is NP-hard for any $\epsilon > 0$ (Zuckerman, 2007), so approximations are our best hope. Alg. 1 summarizes our approach.

Hybrid Methods. LPDG is efficient to compute, but $w^s(A)$ has strong lower-bound guarantees on improvement of the dual value. It is natural to consider hybrid methods that use LPDG to quickly propose candidate clusters with high LPDG-scores, and then only com-

Algorithm 1 IterativeMAP

$(\mu^*_{\mathcal{V},\mathcal{E}}) = \textsc{IterativeMAP}(G, J^{(0)}, \theta, \mathcal{F})$
Input:
  $G = (\mathcal{V}, \mathcal{E})$, graph instance
  $J^{(0)}$ initial set of pencils {Typically $J^{(0)} = \{(A, B, x_B) \mid A = \mathcal{E}, B \subset A\}$}
  $\boldsymbol{\theta} = \{\theta_A\}$ energy vector
  $\mathcal{F} \subseteq 2^{\mathcal{V}}$, family of clusters to search over
Output:
  $\mu^*_A \in [0,1]^{X_A}$, (relaxed) solution vector
Algorithm:
for $t = 0, 1, \ldots$ **do**
  $\mu^{(t)}_{J^{(t)}} \leftarrow \underset{\mu \in \mathcal{L}(J^{(t)})}{\operatorname{argmin}} \boldsymbol{\theta} \cdot \boldsymbol{\mu}$ {Current relaxation}
  if integral($\mu^{(t)}_{J^{(t)}}$) **then**
    break {Optimal solution}
  end if
  $A \leftarrow \underset{A \in \mathcal{F}}{\operatorname{argmax}}\, \omega(A)$ {Scoring}
  if $\omega(A) = 0$ **then**
    break {LP tight over $\mathcal{F}$}
  end if
  $J^{(t+1)} \leftarrow J^{(t)} \cup A$
end for
$\mu^*_{\mathcal{V},\mathcal{E}} \leftarrow \mu^{(t)}_{J(t)}|_{\mathcal{V} \cup \mathcal{E}}$

pute the expensive $w^s(A)$ scores for these candidates. We consider two such hybrids:

1. *LPDGcrisp+Sontag08*: We can ignore actual LPDG scores and only use LPDG to *prune* clusters. We only need to compute expensive $w^s(A)$ scores for all clusters that have non-zero LPDG, because zero LPDG indicates that $w^s(A)$ will be zero. Interestingly, we found that Sontag et al. (2008b) already use such a measure in their publicly available implementation.
2. *LPDG+Sontag08*: The second option is to use the LPDG score to propose a set of K candidates of highest LPDG score, and to compute the expensive $w^s(A)$ scores only for these top candidates.

4 Experiments and Results

We worked with synthetic problems and two real world computer vision applications: stereo vision and image de-convolution. Both the applications are well-studied vision problems, and are known to be difficult both from a vision and an inference perspective, consisting of thousands of nodes, tens of thousands of edges and in one case half a million triplets.

In all cases, we solve LP relaxations using the generalized MPLP message-passing algorithm of Globerson and Jaakkola (2007); Sontag et al. (2008b), which is a block co-ordinate ascent algorithm on the dual LP (shown in Fig. 1). To ensure our experiments assess the relative performance of our proposed LPDG score and are not influenced by the particular details of the dual message-passing implementation, we use the MPLP implementation provided by Sontag et al. (2008b) for all our experiments. Therefore the relative performance differences can be directly attributed to the different scoring functions.

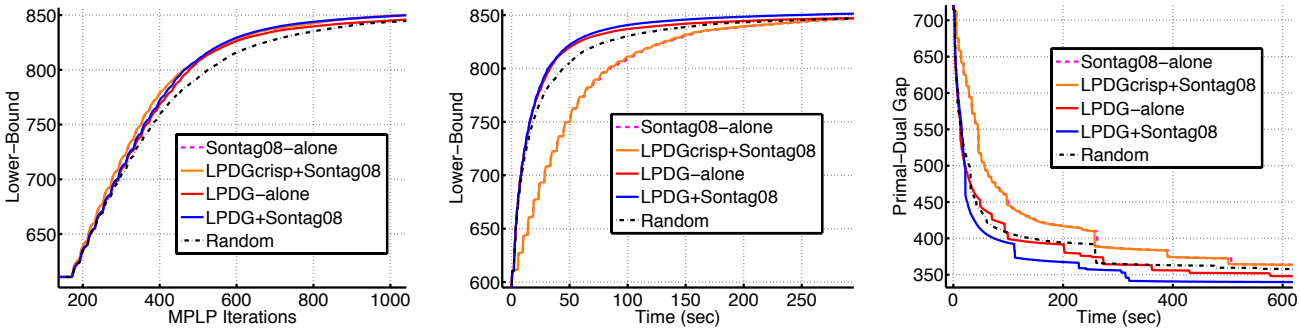
In all experiments, we first solve Schlesinger's edge-consistent LP for at most 1000 MPLP iterations with increase in dual of at least 10^{-4} at each iteration. For all experiments except one de-convolution MRF (with half a million triplets), we perform dictionary-enumeration. We test the following five scoring functions:

- Sontag08-alone (where $\omega(A) = \omega^s(A)$ is computed for all dictionary members),
- LPDGcrisp+Sontag08 (where $\omega(A) = \omega^s(A)$ is computed only for dictionary members with non-zeros LPDG score),
- LPDG-alone (where $\omega(A) = \omega^l(A)$ for all dictionary members),
- LPDG+Sontag08 (where $\omega^l(A)$ is computed for all dictionary members, and only for the top K scoring candidates we compute $\omega^s(A)$), and
- Random (where $\omega(A)$ is sampled from a uniform distribution).

For synthetic experiments, we search over all 3-cycles in K_{30} and all 4-cycles in a 20x20 grid. For stereo vision, our graphs are 4-connected grid-graphs, and our family of clusters is all 4-cycles in the graph. For de-convolution, our graphs are densely-connected grid-graphs, and we search over all 3-cliques.

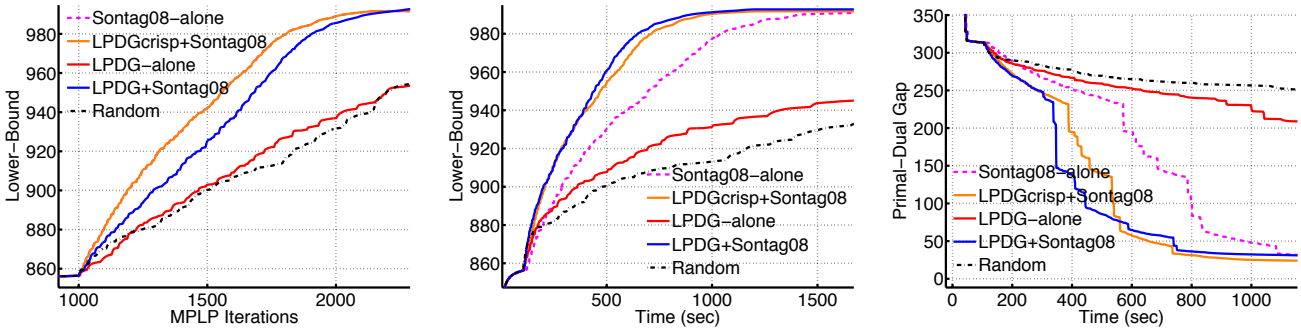
In all experiments, we keep adding clusters till either the integrality gap has fallen below a threshold (10^{-4}), if all dictionary members have been added (typical for smaller experiments) or if the increase in dual value due to adding clusters falls below a threshold of 10^{-4} (typical for larger experiments). If we stop due to the integrality gap threshold, we can say that the MAP solution has been found.

We evaluate the quality of the five cluster-scores in three ways: 1) dual-value vs MPLP iterations, 2) dual-value vs time, and 3) primal-dual gap vs time. Dual value is the objective function of the dual LP (Fig. 1), which provides a lower-bound on the MAP value. A larger value means that the relaxation is tighter. At each MPLP iteration, integral primal labellings are generated by independently minimizing node reparameterized energy vectors, *i.e.* $x^p_i = \operatorname{argmin}_{x_i} \tilde{\theta}_i(x_i)$, breaking ties arbitrarily. The energy of this primal labelling provides an upper-bound on the MAP value.



(a) Dual vs. MPLP iterations. (b) Dual Value vs. Time. (c) Primal-Dual Gap vs. Time.

Figure 3: Synthetic K_{30}: Randomly adding triplets performs better (higher dual-value and lower primal-dual gap) than Sontag08-alone and LPDGcrisp+Sontag08. LPDG is non-zero for most cycles and thus LPDGcrisp+Sontag08 performs the same as Sontag08.



(a) Dual vs. MPLP iterations. (b) Dual Value vs. Time. (c) Primal-Dual Gap vs. Time.

Figure 4: Synthetic 20x20 grid: LPDG-alone and Random perform poorly, while LPDG+Sontag08 performs the best (with highest dual value and lowest primal-dual gap at any given time). Interestingly, LPDGcrisp+Sontag08 performs significantly better than Sontag08-alone so LPDG does have some information in it.

At each MPLP iteration, primal-dual gap is the difference between the best upper-bound so far and lower-bound at this iteration. A primal-dual gap of zero means that we have obtained an optimal solution to the problem and guarantee of optimality. A non-zero primal-dual gap of g provides a per-instance guarantee that the current primal solution is at most g worse in energy than the MAP solution.

Scoring functions come into play only after the edge-consistent LP is solved, so our plots show behaviour immediately after this point. We point out that LPDGcrisp+Sontag08 and Sontag08-alone compute exactly the same scores and differ only in the time taken to compute these scores. Hence they will be *exactly* the same curve in dual-value vs iteration plots.

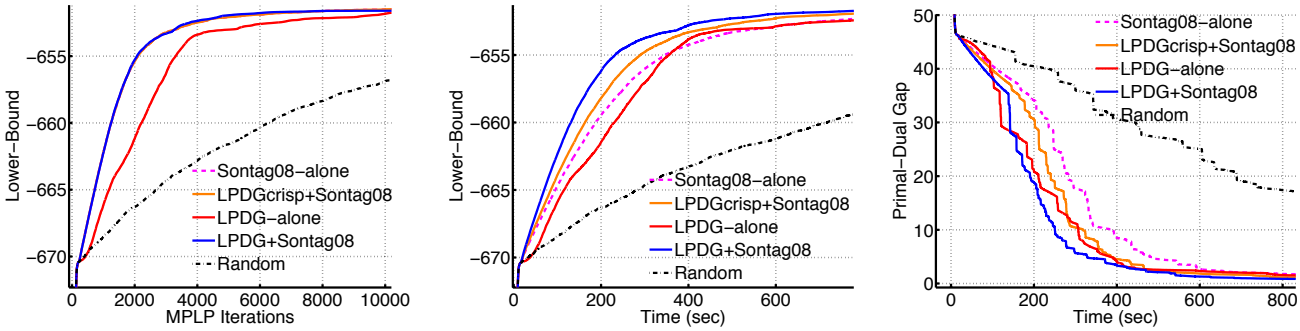
All experiments are performed on a 64-bit 8-Core Mac Pro with 8GB RAM and the timing reported is cpu-time (via the c++ clock() function).

Our results will demonstrate the effectiveness of LPDG as both a scoring function for cliques by itself and as an efficient candidate generation heuristic to compute expensive w^s scores on only a subset of cliques. We will show that the hybrid method LPDG+Sontag08 leads to the tightest relaxation (highest value of dual and lowest value of primal-dual gap) and provides significant speed-ups over all other methods. Moreover, LPDG-based methods will allow clique search in previously unsolvable cases, where the dictionary size is prohibitively large.

4.1 Synthetic Problems

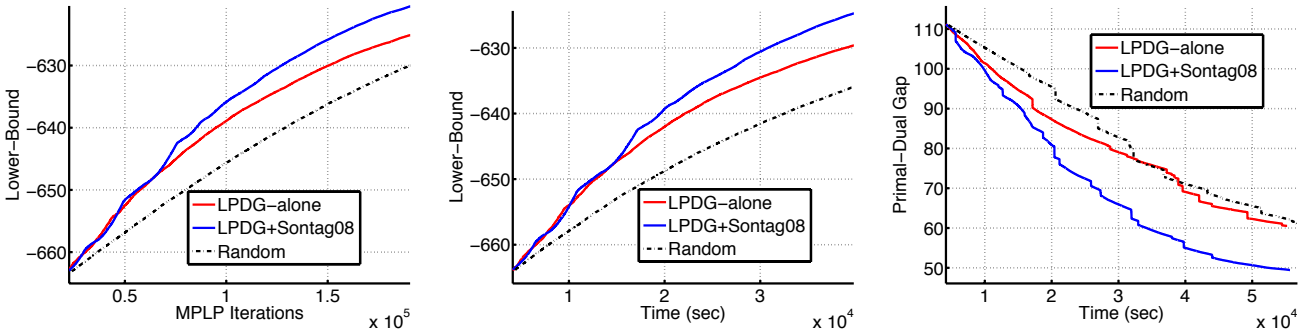
For synthetic problem we experimented with two graph structures: a complete graph on 30 nodes, K_{30}; and a 20x20 grid.

K_{30} **:** Each variable in the model took 20 states. We create synthetic energy functions by sampling from Gaussians, $\theta_i(x_i) \sim \mathcal{N}(0,1)$ and $\theta_{ij}(x_i,x_j) \sim \mathcal{N}(0,4)$. Fig. 3 shows dual-value vs MPLP iterations, dual-value vs time and primal-dual gap vs time. We can see that the edge-consistent LP is solved in ∼200 iteration and less than 10 sec, but it leaves a primal-dual gap of around 700. There are a total of $\binom{30}{3} = 4060$ triplets in this model, and we added triplets in batches of 20. For LPDG+Sontag08, we first find 60 candidates with the highest LPDG scores, and then add the 20 (of the 60) with the highest w^s scores. For this model, the times taken by various methods for each round triplet-search were 3-4 secs for Sontag08-alone and LPDGcrisp+Sontag08, 5-10 millisecs for LPDG-alone,



(a) Dual vs. MPLP iterations. (b) Dual Value vs. Time. (c) Primal-Dual Gap vs. Time.

Figure 5: Image Deconvolution: 3x3 blur kernel. We observe that LPDG+Sontag08 performs the best, but even LPDG-alone performs very well, and leads to lower primal-dual gap than Sontag08-alone.



(a) Dual vs. MPLP iterations. (b) Dual Value vs. Time. (c) Primal-Dual Gap vs. Time.

Figure 6: Image Deconvolution: 5x5 blur kernel. This model contains $\sim 450,000$ triplets and the dictionary-enumeration implementation of Sontag et al. (2008b) runs out of memory. LPDG-based methods allow for greedy search methods, which perform significantly better than randomly adding cycles.

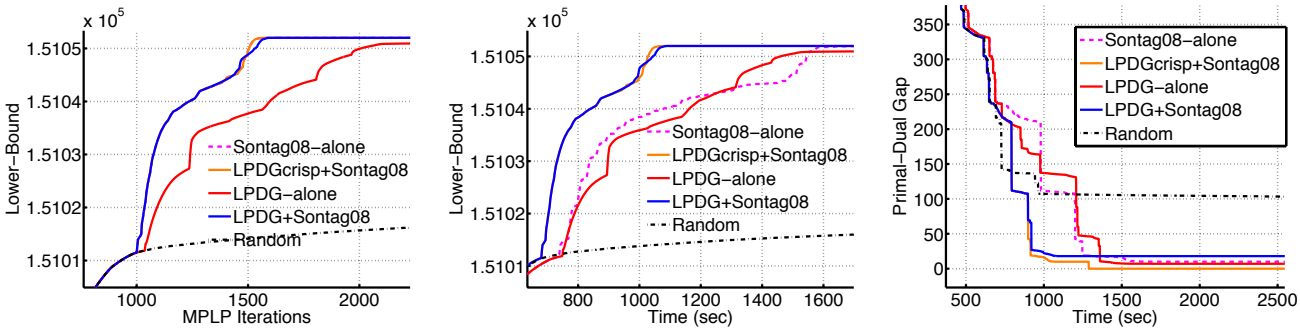
50-80 millisecs for LPDG+Sontag08 and 5-10 millisecs for Random. We can see that computing LPDG scores is really cheap – almost as fast as sampling scores randomly. As a comparison, the time taken for one round of message-passing was 0.01 sec before adding any triplets, and 1-2 secs after adding all 4060 cycles (the message-passing time depends on the number of triplets in J). Thus, in the time taken by Sontag08 to search for the best cycles, LPDG-based methods can add cycles and run between 3 and 345 extra rounds of message passing (depending on how many triplets are already in the relaxation). This is exactly why we see LPDG-based methods producing higher dual values and lower primal-dual gaps at corresponding time budgets. Interestingly, in this case, we also observe that randomly adding triplets performs better than Sontag08-alone and LPDGcrisp+Sontag08. This is not surprising given that the θs are randomly generated, thus a priori, no clique has any reason to be significantly more helpful than any other. However, this will not be true for real applications where the energies have a lot of structure to them and typically a few very helpful cliques. The pattern that we do see in all our experiments is that clique-search times dominate message-passing times (at least till a large number of cliques are added to the relaxation), and this makes LPDG-based methods very important.

20x20 grid: Each variable in this model took 50 states. The node energies were again sampled from standard Gaussians. With edge-energies we tried to simulate the contrast-sensitive Potts model, commonly used in vision problems, *i.e.*, $\theta_{ij}(x_i, x_j) = \lambda_{ij} \cdot \delta(x_i - x_j)$, where $\delta(\cdot)$ is the dirac-delta function which is 1 if the input argument is 0, and 0 otherwise. λ_{ij} is sampled from a uniform distribution, $\lambda_{ij} \sim [-1\ 1]$. Fig. 4 shows the results. 4-cycles were added in batches of 1, and 10 candidates were generated by LPDG for LPDG+Sontag08. We can see that as before the edge-consistent LP is solved quickly, and most of the time is spent in finding 4-cycles to add to the relaxation. Random performs the worst and LPDG+Sontag08 performs the best, both in terms of having the highest dual-value and lowest primal-dual gap at any time.

4.2 Image De-convolution

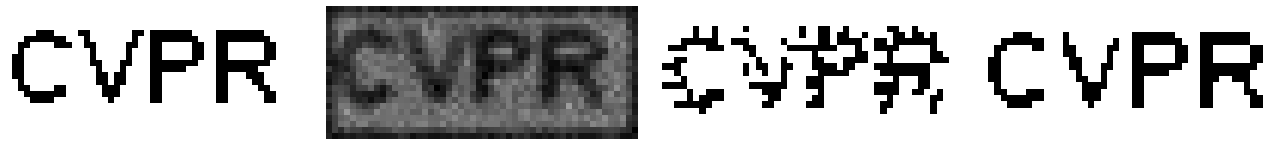
Given a blurry and noisy (binary) input image the task in image de-convolution is to reconstruct the ground truth. Raj and Zabih (2005) formulated generalized image deconvolution as a MAP-inference problem and Rother et al. (2007) gave models for binary images. We use the "CVPR" model of Rother et al. (2007).[1] Fig. 8

[1]Unfortunately, they did not create an "AISTATS" model.



(a) Dual vs. MPLP iterations. (b) Dual Value vs. Time. (c) Primal-Dual Gap vs. Time.

Figure 7: Stereo Vision: Tsukuba. Even though Schlesinger's LP is not tight, it is very close to the MAP. Out of $\sim 17,500$ cycles LPDG scores for all but $\sim$50 cycles were completely zero ($\leq 10^{-12}$). As a result, randomly adding cycles performs very poorly and both LPDGcrisp+Sontag08 and LPDG+Sontag08 give significant improvement over random.



(a) Original Image. (b) Blurry Noisy Image. (c) Schlesinger's LP solution. (d) MAP from Triplet LP.

Figure 8: Image Deconvolution: 3x3 blur kernel. We can see that edge-consistent LP solution is of poor quality, while the MAP state almost completely recovers the original image. Image courtesy Rother et al. (2007)

shows the blurry (20x50) image. Given an $n \times n$ blur-kernel, the MRF connectivity is $(2n-1) \times (2n-1) - 1$. For a 3x3 blur-kernel, this model contains $\sim$45,000 triplets, and for a 5x5 blur-kernel $\sim$450,000 triplets.

Fig. 5 shows the results for the 3x3 blur-kernel. Again, we find that adding random triplets performs the poorest and LPDG+Sontag08 performs the best.

The 5x5 blur-kernel model is prohibitively large for dictionary-enumeration methods and we found that the implementation of Sontag et al. (2008b) ran out of memory while computing scores for half a million triplets. We implemented a greedy search algorithm that constructs a heavy cluster by incrementally adding parts (nodes, edges) with the highest LPDG scores. We also use LPDG to propose top 100 candidates and only computed expensive w^s scores on those. Fig. 6 shows the results.

4.3 Stereo Vision

Stereo vision involving predicting disparity labels for each pixel given a pair of stereo images. Graphical models for this problem have been proposed by Tappen and Freeman (2003) and extensively studied by Meltzer et al. (2005). For our experiments, we use the Tsukuba MRF of Sontag et al. (2008b). The graph is 4-connected grid-graph (116x154 pixels) with each variable having 16 labels.

Fig. 7 shows the results. This model contains $(116-1) \times (154-1) \sim 17,500$ 4-cycles and we found that after solving the edge-consistent LP, LPDG scores for all but $\sim$50 cycles were completely zero ($\leq 10^{-12}$). This tell us that the edge-consistent LP is a very tight approximation to the MAP, and the cycle-selection problem is indeed very important because most of the cycles in this model are now completely useless. As an immediate result, randomly adding cycles performs very poorly and both LPDGcrisp+Sontag08 and LPDG+Sontag08 give significant improvement over Sontag08-alone. To give an idea of times, each message-passing iteration takes 0.65-0.90 sec, while each round of cycle-search takes 14-15 sec for Sontag08-alone, 0.1 sec for LPDGcrisp+Sontag08, 0.07 sec for LPDG-alone, 0.08 sec for LPDG+Sontag08 and 0.04 sec for Random.

5 Conclusions

In summary, we presented a cluster-scoring function for obtaining cluster-based LPs relaxation to MAP inference in MRFs. Our cluster-scoring function, which we call Local Primal-Dual Gap, is a generalization of the complementary slackness condition in the primal dual programs of LP relaxation of MAP-MRF. Intuitively LPDG quantifies by how much the complementary slackness condition is violated, and attributes violations to individual potentials. This combined with the fact that LPDG can be computed very cheaply from the primal and dual variables, makes LPDG a powerful clique pruning heuristic. We showed that combining LPDG with the expensive scoring function of Sontag et al. (2008b) works best in practice; in one case allowing us to solve a previously insolvable problem.

References

D. Bertsimas and J. N. Tsitsiklis. *Introduction to Linear Optimization*. 1997.

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, 2004.

A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. In *NIPS*, 2007.

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

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

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

A. Koster, C. P. M. van Hoesel, and A. W. J. Kolen. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters*, 23:89–97, 1998.

T. Meltzer, C. Yanover, and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *ICCV*, pages 428–435, 2005.

A. Raj and R. Zabih. A graph cut algorithm for generalized image deconvolution. In *ICCV*, volume 2, pages 1048–1054 Vol. 2, oct. 2005.

C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, June 2007.

M. I. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions (in Russian). *Kibernetika*, 4:113–130, 1976.

S. E. Shimony. Finding maps for belief networks is NP-hard. *Artificial Intelligence*, 68(2):399–410, August 1994.

D. Sontag. *Approximate Inference in Graphical Models using LP Relaxations*. PhD thesis, Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2010.

D. Sontag and T. Jaakkola. Tree block coordinate descent for MAP in graphical models. In *AISTATS*, volume 8, pages 544–551. JMLR: W&CP, 2009.

D. Sontag, A. Globerson, and T. Jaakkola. Clusters and coarse partitions in LP relaxations. In *NIPS*, 2008a.

D. Sontag, T. Meltzer, A. Globerson, T. Jaakkola, and Y. Weiss. Tightening LP relaxations for MAP using message passing. In *UAI*, 2008b.

M. Tappen and W. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *ICCV*, pages 900–906 vol.2, oct. 2003.

M. Wainwright, T. Jaakkola, and A. Willsky. Map estimation via agreement on (hyper)trees: Message-passing and linear-programming approaches. *IEEE Trans. Inf. Th.*, 51(11):3697–3717, 2005.

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

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

T. Werner. High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft constraint optimisation (MAP-MRF). In *CVPR*, 2008.

C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation – an empirical study. *J. Mach. Learn. Res.*, 7:1887–1907, 2006.

D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3(1):103–128, 2007.