Efficiently Searching for Frustrated Cycles in MAP Inference (Supplementary Material)

David Sontag, Do Kook Choe, Yitao Li
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University

1 Derivation of Dual with Cycle Inequalities

In this section we show how to derive the *dual* of the LP relaxation which includes all of the k-ary cycle inequalities. We show this for pairwise MRFs, but it is straightforward to generalize to arbitrary factor graphs (see, e.g., [2]). Consider the primal LP,

$$\max_{\mu \geq 0} \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) + \sum_{ij} \sum_{x_i, x_j} \mu_{ij}(x_i, x_j)\theta_{ij}(x_i, x_j) \tag{1}$$

subject to:

$$\begin{aligned}
\sum_{x_j} \mu_{ij}(x_i, x_j) &= \mu_i(x_i), \ \ \forall ij \in E, x_i \\
\sum_{x_i} \mu_{ij}(x_i, x_j) &= \mu_j(x_j), \ \ \forall ij \in E, x_j \\
\sum_{ij \in C \setminus F} \sum_{\substack{x_i, x_j : \\ \pi_i(x_i) \neq \pi_j(x_j)}} \mu_{ij}(x_i, x_j) + \sum_{ij \in F} \sum_{\substack{x_i, x_j : \\ \pi_i(x_i) = \pi_j(x_j)}} \mu_{ij}(x_i, x_j) &\geq 1, \ \ \forall \text{cycles } C, F \subseteq C, |F| \text{ odd}, \pi \\
\sum_{x_i, x_j} \mu_{ij}(x_i, x_j) &= 1, \ \ \forall ij \in E \\
\sum_{x_i} \mu_i(x_i) &= 1, \ \ \forall i \in V
\end{aligned}$$

We first introduce the Lagrange multipliers $\delta_{ji}(x_i)$ an $\delta_{ij}(x_j)$ for the first two constraints. We also introduce Lagrange multipliers $\lambda_{C,F,\pi}$ for each k-ary cycle inequality. Leaving the last two equality constraints and the non-negativity constraints explicit, we obtain the following equivalent optimization problem:

$$\begin{aligned}
\min_{\lambda \geq 0, \delta} \max_{\mu \geq 0} \quad & \sum_i \sum_{x_i} \mu_i(x_i)\theta_i(x_i) + \sum_{ij \in E} \sum_{x_i, x_j} \mu_{ij}(x_i, x_j)\theta_{ij}(x_i, x_j) \\
& + \sum_{ij \in E} \sum_{x_i} \delta_{ji}(x_i)\Big(\mu_i(x_i) - \sum_{x_j} \mu_{ij}(x_i, x_j)\Big) \\
& + \sum_{ij \in E} \sum_{x_j} \delta_{ij}(x_j)\Big(\mu_j(x_j) - \sum_{x_i} \mu_{ij}(x_i, x_j)\Big) \\
& + \sum_{C,F,\pi} \lambda_{C,F,\pi}\Big(\sum_{ij \in C \setminus F} \sum_{\substack{x_i, x_j : \\ \pi_i(x_i) \neq \pi_j(x_j)}} \mu_{ij}(x_i, x_j) + \sum_{ij \in F} \sum_{\substack{x_i, x_j : \\ \pi_i(x_i) = \pi_j(x_j)}} \mu_{ij}(x_i, x_j) - 1\Big)
\end{aligned}$$

subject to:

$$\sum_{x_i,x_j} \mu_{ij}(x_i,x_j) = 1, \ \forall ij \in E \tag{2}$$

$$\sum_{x_i} \mu_i(x_i) = 1, \ \forall i \in V. \tag{3}$$

Re-arranging the objective, we get

$$\begin{aligned}
\min_{\lambda\geq 0,\delta} \max_{\mu\geq 0} \quad & \sum_i \sum_{x_i} \mu_i(x_i)\Big(\theta_i(x_i) + \sum_{j\in N(i)} \delta_{ji}(x_i)\Big) \\
& + \sum_{ij} \sum_{x_i,x_j} \mu_{ij}(x_i,x_j)\Big(\theta_{ij}(x_i,x_j) - \delta_{ji}(x_i) - \delta_{ij}(x_j) \\
& \qquad + \sum_{C,F,\pi \,:\, ij\in C\setminus F} \lambda_{C,F,\pi} 1[\pi_i(x_i) \neq \pi_j(x_j)] \\
& \qquad + \sum_{C,F,\pi \,:\, ij\in F} \lambda_{C,F,\pi} 1[\pi_i(x_i) = \pi_j(x_j)]\Big) \\
& - \sum_{C,F,\pi} \lambda_{C,F,\pi} \ .
\end{aligned}$$

Finally, we analytically solve the maximization with respect to $\mu \geq 0$ and the normalization constraints from Eq. 2 and Eq. 3 to obtain the dual objective

$$\begin{aligned}
J(\delta,\lambda) = & \sum_i \max_{x_i} \Big(\theta_i(x_i) + \sum_{j\in N(i)} \delta_{ji}(x_i)\Big) \\
& + \sum_{ij} \max_{x_i,x_j} \Big(\theta_{ij}(x_i,x_j) - \delta_{ji}(x_i) - \delta_{ij}(x_j) \\
& \qquad + \sum_{C,F,\pi \,:\, ij\in C\setminus F} \lambda_{C,F,\pi} 1[\pi_i(x_i) \neq \pi_j(x_j)] \\
& \qquad + \sum_{C,F,\pi \,:\, ij\in F} \lambda_{C,F,\pi} 1[\pi_i(x_i) = \pi_j(x_j)]\Big) \\
& - \sum_{C,F,\pi} \lambda_{C,F,\pi} \ .
\end{aligned} \tag{4}$$

2 Coordinate Descent on Cycle Inequality Dual Variables

We next show how to take a coordinate descent step with respect to one $\lambda_{C,F,\pi}$ variable. The corresponding update could be used together with the block coordinate-descent approaches for solving the dual together with cycle inequalities. In our paper, however, we use this as the basis for the bound criterion that we use to choose cycles.

The dual objective $J(\delta,\lambda)$ is a piecewise linear function of $\lambda_{C,F,\pi}$ (see Figure 1). As long as $\lambda_{C,F,\pi}$ does not play a role in the edge terms (we will make this precise in a moment), we can *increase* the value of $\lambda_{C,F,\pi}$, thereby decreasing $J(\delta,\lambda)$. On the other hand, if $\lambda_{C,F,\pi}$ is active in two or more edge terms (which outweighs the single $-\lambda_{C,F,\pi}$ term), *decreasing* its value will decrease $J(\delta,\lambda)$.

For $ij \in C$, consider the inside of the edge maximization terms, ignoring the terms that involve $\lambda_{C,F,\pi}$.

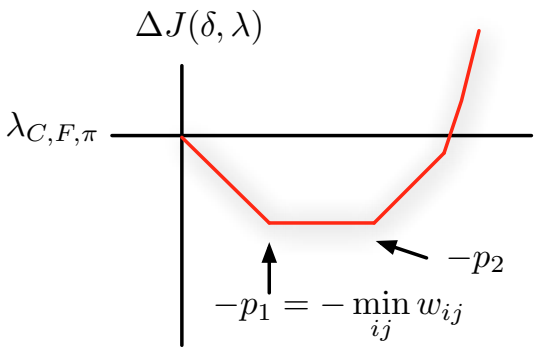

Figure 1: Illustration of decrease in dual objective (see Eq. 4) as a function of $\lambda_{C,F,\pi}$. p_1 refers to the minimal value in $\{w_{ij}\}$, while p_2 is the next smallest value. Clearly we must have $w_{ij} > 0$ for all edges, as otherwise the decrease would be zero.

Defining

$$b_{ij}^{-C,F,\pi}(x_i, x_j) = \theta_{ij}(x_i, x_j) - \delta_{ji}(x_i) - \delta_{ij}(x_j) + \sum_{\substack{(C',F',\pi') \neq (C,F,\pi) \\ : ij \in F'}} \lambda_{C',F',\pi'} 1[\pi'_i(x_i) = \pi'_j(x_j)] + \sum_{\substack{(C',F',\pi') \neq (C,F,\pi) \\ : ij \in C' \setminus F'}} \lambda_{C',F',\pi'} 1[\pi'_i(x_i) \neq \pi'_j(x_j)] \ ,$$

we can rewrite the relevant terms of the dual objective in Eq. 4 as

$$\begin{aligned} J(\lambda_{C,F,\pi}) \quad = \quad & \sum_{ij \in F} \max_{x_i, x_j} \left(b_{ij}^{-C,F,\pi}(x_i, x_j) + \lambda_{C,F,\pi} 1[\pi_i(x_i) = \pi_j(x_j)] \right) \\ & + \sum_{ij \in C \setminus F} \max_{x_i, x_j} \left(b_{ij}^{-C,F,\pi}(x_i, x_j) + \lambda_{C,F,\pi} 1[\pi_i(x_i) \neq \pi_j(x_j)] \right) \\ & - \lambda_{C,F,\pi} \ . \end{aligned}$$

If $ij \in F$, we call $\lambda_{C,F,\pi}$ *active* for edge ij when $\max_{x_i,x_j \,:\, \pi_i(x_i) = \pi_j(x_j)} \left(b_{ij}^{-C,F,\pi}(x_i, x_j) + \lambda_{C,F,\pi} \right) \geq \max_{x_i,x_j} b_{ij}^{-C,F,\pi}(x_i, x_j)$, in which case further increasing $\lambda_{C,F,\pi}$ results in a linear increase in the corresponding edge term of $J(\lambda_{C,F,\pi})$. Similarly, if $ij \notin F$, we call $\lambda_{C,F,\pi}$ active for edge ij when

$$\max_{x_i,x_j \,:\, \pi_i(x_i) \neq \pi_j(x_j)} \left(b_{ij}^{-C,F,\pi}(x_i, x_j) + \lambda_{C,F,\pi} \right) \geq \max_{x_i,x_j} b_{ij}^{-C,F,\pi}(x_i, x_j). \tag{5}$$

We define w_{ij} to be the largest that $\lambda_{C,F,\pi}$ can be before becoming active for edge ij:

$$w_{ij} = \begin{cases} \max_{x_i,x_j \,:\, \pi_i(x_i) \neq \pi_j(x_j)} b_{ij}^{-C,F,\pi}(x_i, x_j) - \max_{x_i,x_j \,:\, \pi_i(x_i) = \pi_j(x_j)} b_{ij}^{-C,F,\pi}(x_i, x_j) & \text{if } ij \in F, \\ \max_{x_i,x_j \,:\, \pi_i(x_i) = \pi_j(x_j)} b_{ij}^{-C,F,\pi}(x_i, x_j) - \max_{x_i,x_j \,:\, \pi_i(x_i) \neq \pi_j(x_j)} b_{ij}^{-C,F,\pi}(x_i, x_j) & \text{if } ij \notin F. \end{cases} \tag{6}$$

When $\min_{ij \in C} w_{ij} > 0$, the dual objective $J(\delta, \lambda)$ decreases as $\lambda_{C,F,\pi}$ increases, until $\lambda_{C,F,\pi} = \min_{ij \in C} w_{ij} = p_1$ (let ij^* denote the argmin). At this point, the function has zero slope, and remains constant until $\lambda_{C,F,\pi} = \min_{ij \neq ij^*} w_{ij} = p_2$. Thus, by setting $\lambda_{C,F,\pi} = \frac{p_1 + p_2}{2}$ we obtain the maximal decrease. When $p_2 \neq p_1$, there are a range of values for $\lambda_{C,F,\pi}$ that achieve the maximal decrease in the dual objective. The midpoint might be preferable because it leads to dual optimal solutions for which "decoding", or finding the corresponding primal optimal solution, is easier.

3 Proofs for Section 4

Theorem 3.1. *When $k = 2$ and the beliefs $b_{ij}(x_i, x_j)$ correspond to a dual optimal solution,* $\max_{C \in \mathcal{C}_{cycles}(G)} d(C) = \max_{C,F:|F| \text{ odd}} d(C, F)$.

Proof. First, defining $w_e(\mathbf{x}_e) = \max_{\hat{\mathbf{x}}_e} b_e(\hat{\mathbf{x}}_e) - b_e(\mathbf{x}_e)$, note that

$$\begin{aligned} d(C) &= \sum_{e \in C} \max_{\mathbf{x}_e} b_e(\mathbf{x}_e) - \max_{\mathbf{x}_C} \left[\sum_{e \in C} b_e(\mathbf{x}_e) \right] && (7) \\ &= \sum_{e \in C} \max_{\mathbf{x}_e} b_e(\mathbf{x}_e) + \min_{\mathbf{x}_C} \left[- \sum_{e \in C} b_e(\mathbf{x}_e) \right] && (8) \\ &= \min_{\mathbf{x}_C} \sum_{e \in C} \left[\left(\max_{\hat{\mathbf{x}}_e} b_e(\hat{\mathbf{x}}_e) \right) - b_e(\mathbf{x}_e) \right] = \min_{\mathbf{x}_C} \sum_{e \in C} w_e(\mathbf{x}_e) \ . && (9) \end{aligned}$$

Our proof proceeds as follows. In part (a) we show that for any cycle C where $d(C) > 0$, for all edges $ij \in C$, either $\arg\max_{x_i,x_j} b_{ij}(x_i, x_j) = \{(0,0),(1,1)\}$ or $\arg\max_{x_i,x_j} b_{ij}(x_i, x_j) = \{(1,0),(0,1)\}$.[1] Then, calling the edges with maximizing assignments equal to $\{(1,0),(0,1)\}$ "cut" and the edges with maximizing assignments equal to $\{(0,0),(1,1)\}$ "not cut", we show in part (b) that a cycle C has $d(C) > 0$ if and only if it has an *odd* number of cut edges. In part (c) we show that, when $d(C) > 0$, $d(C) = \min_{e \in C, \mathbf{x}_e \text{ s.t. } w_e(\mathbf{x}_e) > 0} w_e(\mathbf{x}_e)$.

Recall that, by part (a), $b_{ij}(0,0) = b_{ij}(1,1)$ for an edge that is not cut and $b_{ij}(0,1) = b_{ij}(1,0)$ for an edge that is cut. Let the cost of "cutting" an edge ij refer to the smallest value t such that either of $b_{ij}(0,1) + t$ or $b_{ij}(1,0) + t$ is equal in value to $b_{ij}(0,0)$ and $b_{ij}(1,1)$ Similarly, let the cost of "un-cutting" an edge ij refer to the smallest value t such that either of $b_{ij}(0,0) + t$ or $b_{ij}(1,1) + t$ is equal in value to $b_{ij}(0,1)$ and $b_{ij}(1,0)$. It follows from part (c) that, when $d(C) > 0$, $d(C)$ is the minimal cost, over all edges in C, of cutting an edge that is not cut, or un-cutting an edge that is cut. Thus, letting F' be the set of cut edges in C, when $d(C) > 0$ we have

$$\min_{e \in C, \mathbf{x}_e \text{ s.t. } w_e(\mathbf{x}_e) > 0} w_e(\mathbf{x}_e) = \max_{F \subseteq C : |F| \text{ odd}} \ \min_{ij \in C} \ w_{ij} \ , \tag{10}$$

where w_{ij} was defined in Eq. 6. The equality is because $\min_{e \in C, \mathbf{x}_e \text{ s.t. } w_e(\mathbf{x}_e) > 0} w_e(\mathbf{x}_e) = \min_{ij \in C} \ w_{ij}$ for $F = F'$ (and, by part (b), $|F'|$ is odd), and $\min_{ij \in C} \ w_{ij} < 0$ for $F \neq F'$ (whereas the left hand side of Eq. 10 is positive). By part (b), when $d(C) = 0$ we have that $\min_{ij \in C} \ w_{ij} < 0$ for *all* $|F|$ odd, so $\max_{F \subseteq C : |F| \text{ odd}} d(C, F) = 0$. We conclude that $d(C) = \max_{F \subseteq C : |F| \text{ odd}} d(C, F)$, which shows the claim.

(a) We first show that either $\{(0,0),(1,1)\} \subseteq \arg\max_{x_i,x_j} b_{ij}(x_i, x_j)$ or $\{(1,0),(0,1)\} \subseteq \arg\max_{x_i,x_j} b_{ij}(x_i, x_j)$, or both (these are not mutually exclusive). By the definition of $w_e(\mathbf{x}_e)$, every edge ij has at least one assignment x_i, x_j such that $w_{ij}(x_i, x_j) = 0$. Suppose for contradiction that there is an edge ij such that $\{(0,0),(1,1)\} \not\subseteq \arg\max_{x_i,x_j} b_{ij}(x_i, x_j)$ and $\{(1,0),(0,1)\} \not\subseteq \arg\max_{x_i,x_j} b_{ij}(x_i, x_j)$. We will show just the following case, with the others following by similar arguments:

$$\begin{aligned} (1,1) &\notin \arg\max_{x_i,x_j} b_{ij}(x_i, x_j) \ , && (11) \\ (1,0) &\notin \arg\max_{x_i,x_j} b_{ij}(x_i, x_j) \ , \text{ and} && (12) \\ (0,0) &\in \arg\max_{x_i,x_j} b_{ij}(x_i, x_j) \ . && (13) \end{aligned}$$

Let μ be any primal optimal solution corresponding to b. By complementary slackness, we have that $\mu_{ij}(0,0) > 0$, which implies that $\mu_i(0) > 0$. Let j and k denote the two neighbors of node i in the cycle. By complimentary slackness and the pairwise consistency constraints, for any x_i, if there exists x_k

[1] We use the notation $\arg\max_{x_i,x_j} b_{ij}(x_i, x_j)$ to refer to the *set* of assignments $\hat{x}_i, \hat{x}_j$ such that $b_{ij}(\hat{x}_i, \hat{x}_j) = \max_{x_i,x_j} b_{ij}(x_i, x_j)$.

such that $w_{ki}(x_k, x_i) = 0$, then there exists x_j such that $w_{ij}(x_i, x_j) = 0$. Using this property, we can construct an assignment $\mathbf{x}_C$ for the variables of the cycle such that $\sum_{e \in C} w_e(\mathbf{x}_e) = 0$ by starting with $x_i = 0$ and consecutively setting each neighbor's assignment (starting with k, and continuing in the same direction along the cycle). Importantly, we must return to $x_i = 0$ because we have assumed that $w_{ij}(1,0) > 0$ and $w_{ij}(1,1) > 0$. We have thus contradicted our assumption that $d(C) > 0$.

To show the equality, suppose for contradiction that there is an edge ij such that $\{(0,0),(1,1),(0,1)\} \subseteq \arg\max_{x_i, x_j} b_{ij}(x_i, x_j)$. Then, we can construct an assignment using the same technique, starting at $x_i = 0$ and going in the direction of k, which shows that $d(C) = 0$, again contradicting our assumption.

(b) Since $w_e(\mathbf{x}_e) \geq 0$, $\sum_{e \in C} w_e(\mathbf{x}_e) = 0$ if and only if every edge assignment $\mathbf{x}_e$ is consistent with our definition of "cut" and "not cut". However, any assignment to the variables in a cycle must correspond to an even number of cut edges. Thus, $d(C) = 0$ if and only if the cycle has an even number of cut edges.

(c) If $d(C) > 0$, then there are an odd number of cut edges. Cutting an uncut edge or un-cutting a cut edge would make the cycle have an even number of cut edges, and so there would be an assignment $\mathbf{x}_C$ such that $w_e(\mathbf{x}_e) = 0$ for all edges other than the modified edge, using the construction from part (a). Thus, for all edges e' and edge assignments $\mathbf{x}_{e'}$ such that $w_{e'}(\mathbf{x}_{e'}) > 0$, $\min_{\mathbf{x}_{C \setminus e'}} \sum_{e \in c} w_e(\mathbf{x}_e) = w_{e'}(\mathbf{x}_{e'})$. Since $w_e(\mathbf{x}_e) \geq 0$ always, if $d(C) > 0$ then for all assignments $\mathbf{x}_c$, there must be some e' and $\mathbf{x}_{e'}$ such that $w_{e'}(\mathbf{x}_{e'}) > 0$. However, we just showed that there exists a completion $\mathbf{x}_{C' \setminus e'}$ such that $w_e(\mathbf{x}_e) = 0$ for all edges $e \neq e'$. Thus, when $d(C) > 0$, the value of $d(C)$ is equal to $\min_{e \in C, \mathbf{x}_e \text{ s.t. } w_e(\mathbf{x}_e) > 0} w_e(\mathbf{x}_e)$. □

Theorem 3.2. *The optimization problem* $\max_{C \in \mathcal{C}_{cycles}(G)} d(C)$ *is NP-hard for* $k = 2$ *and beliefs* $b_{ij}(x_i, x_j)$ *arising from a non-optimal dual feasible point.*

Proof. Our reduction is from the Hamiltonian cycle problem, which is known to be NP-hard. The Hamiltonian cycle problem is: Given a graph $G = (V, E)$, decide whether there exists a cycle in G that visits all vertices exactly once.

We show how to efficiently construct a Markov random field $G' = (V', E')$ with $x_i \in \{0, 1\}$ and beliefs $b_{ij}(x_i, x_j)$ for $ij \in E'$ such that there is a 1-1 mapping between cycles $C \in G$ and $C' \in G'$, and evaluating $d(C')$ for $C' \in G'$ gives the length of the corresponding cycle in G. As a result, we have that $\max_{C' \in \mathcal{C}_{\text{cycles}}(G')} d(C')$ gives the length of the longest cycle in G. Thus, if we could solve this optimization problem, then, simply by checking whether the solution is $|V|$, we answer the Hamiltonian cycle problem.

Let $V' = V \cup \{x_{ij}, \forall ij \in E\}$, where we introduce a new variable x_{ij} for every edge in E. The edges are $E' = \{(i, x_{ij}), (x_{ij}, j), \forall ij \in E\}$, where we replace every edge in G with a length-2 path in G'. For each $ij \in E$, denoting $k = x_{ij}$, we let the beliefs be:

$b_{ik}(x_i, x_{ij}) =$

	$x_{ij} = 0$	$x_{ij} = 1$
$x_i = 0$	$\lvert V \rvert$	0
$x_i = 1$	0	0

$b_{kj}(x_{ij}, x_j) =$

	$x_j = 0$	$x_j = 1$
$x_{ij} = 0$	0	0
$x_{ij} = 1$	0	1

Then, we have that:

$w_{ik}(x_i, x_{ij}) =$

	$x_{ij} = 0$	$x_{ij} = 1$
$x_i = 0$	0	$\lvert V \rvert$
$x_i = 1$	$\lvert V \rvert$	$\lvert V \rvert$

$w_{kj}(x_{ij}, x_j) =$

	$x_j = 0$	$x_j = 1$
$x_{ij} = 0$	1	1
$x_{ij} = 1$	1	0

As a result of our construction, every cycle $C \in G$ on nodes $i, j, k, \ldots, m$ corresponds 1-1 with the cycle $C' \in G'$ on nodes $i, x_{ij}, j, x_{jk}, \ldots, m, x_{mi}$. It can be verified that for all cycles $C' \in G'$, $d(C') = \min_{\mathbf{x}_{C'}} \sum_{e \in C'} w_e(\mathbf{x}_e) = |C'|/2 = |C|$, where C is the cycle corresponding to C' (the minimum is attained by the all zeros assignment). We thus have that $\max_{C' \in \mathcal{C}_{\text{cycles}}(G')} d(C') = \max_{C \in G} |C|$. □

Theorem 3.3. *The optimization problem* $\max_{C \in \mathcal{C}_{cycles}(G)} d(C)$ *is NP-hard for* $k \geq 3$ *even for beliefs* $b_{ij}(x_i, x_j)$ *corresponding to a dual optimal solution of the pairwise relaxation.*

Proof. As in the proof of Theorem 3.2, we reduce from the Hamiltonian cycle problem, for an input graph G. First, we show that the Hamiltonian cycle problem is NP-hard even when restricted to graphs with an

odd number of nodes, by reducing from the general case. Suppose we are given a graph with an even number of nodes and we want to decide whether it has a Hamiltonian cycle. We repeat the following, once for each edge ij: construct a new graph which is identical to the original except that we introduce a new node n and replace the edge ij with the edges in and nj. We then check whether any of the new graphs (all of which now have an odd number of vertices) have a Hamiltonian cycle. If the answer is "yes", we have found a Hamiltonian cycle for the original graph. Otherwise, the original graph does not have a Hamiltonian cycle.

Assume for the rest of the proof that G has an odd number of vertices. We show how to efficiently construct a Markov random field $G' = (V', E')$ with $x_i \in \{0, 1, 2\}$ and beliefs $b_{ij}(x_i, x_j)$ for $ij \in E'$ such that there is a 1-1 mapping between cycles $C \in G$ and $C' \in G'$, and evaluating $d(C')$ for $C' \in G'$ gives the length of the corresponding cycle in G. As a result, we have that $\max_{C' \in \mathcal{C}_{\text{cycles}}(G')} d(C')$ gives the length of the longest cycle in G. Thus, if we could solve this optimization problem, then, simply by checking whether the solution is $|V|$, we answer the Hamiltonian cycle problem.

Let $V' = V \cup \{x_{ij}, \forall ij \in E\}$, where we introduce a new variable x_{ij} for every edge in E, also with 3 states. The edges are $E' = \{(i, x_{ij}), (x_{ij}, j), \forall ij \in E\}$, where we replace every edge in G with a length-2 path in G'. For each $ij \in E$, denoting $k = x_{ij}$, we let the beliefs be:

$b_{ik}(x_i, x_{ij})$ =

	$x_{ij}=0$	$x_{ij}=1$	$x_{ij}=2$
$x_i=0$	$\|V\|$	0	0
$x_i=1$	0	$\|V\|$	0
$x_i=2$	0	0	$\|V\|-.5$

(14)

$b_{kj}(x_{ij}, x_j)$ =

	$x_j=0$	$x_j=1$	$x_j=2$
$x_{ij}=0$	0	$\|V\|$	0
$x_{ij}=1$	$\|V\|$	0	0
$x_{ij}=2$	0	0	$\|V\|-.5$

As a result of our construction, every cycle $C \in G$ on nodes $i, j, k, \ldots, m$ corresponds 1-1 with the cycle $C' \in G'$ on nodes $i, x_{ij}, j, x_{jk}, \ldots, m, x_{mi}$. Every cycle $C \in G$ where $|C|$ is even corresponds to a cycle $C' \in G'$ such that $\min_{\mathbf{x}_{C'}} \sum_{e \in C'} w_e(\mathbf{x}_e) = 0$ (the minimum is attained by the assignment $0011 \ldots 0011$). On the other hand, every cycle $C \in G$ where $|C|$ is *odd* corresponds to a cycle $C' \in G'$ such that $\min_{\mathbf{x}_{C'}} \sum_{e \in C'} w_e(\mathbf{x}_e) = .5|C'| = |C|$ (the minimum is attained by the assignment of 2 to every node). Thus, G (which has an odd number of nodes) has a Hamiltonian cycle if and only if $\max_{C' \in \mathcal{C}_{\text{cycles}}(G')} d(C') = |V|$.

What remains is to show that the beliefs $b_{ij}(x_i, x_j)$ that we constructed are dual optimal for some potentials $\theta(\mathbf{x})$. We do this by illustrating a primal and dual feasible point for which the primal objective is equal to the dual objective. Let $\theta_{ij}(x_i, x_j) = b_{ij}(x_i, x_j)$ and $\delta_{ij}(x_j) = 0$ for all edges $ij \in E$ and assignments x_j. Clearly $\delta_{ij}(x_j)$ is dual feasible, and it gives an objective value of $|E||V|$. Consider the following primal point μ:

$\mu_{ik}(x_i, x_{ij})$ =

	$x_{ij}=0$	$x_{ij}=1$	$x_{ij}=2$
$x_i=0$	.5	0	0
$x_i=1$	0	.5	0
$x_i=2$	0	0	0

(15)

$\mu_{kj}(x_{ij}, x_j)$ =

	$x_j=0$	$x_j=1$	$x_j=2$
$x_{ij}=0$	0	.5	0
$x_{ij}=1$	.5	0	0
$x_{ij}=2$	0	0	0

The point $\boldsymbol{\mu}$ satisfies the pairwise consistency constraints (the single node marginals are $\mu_i(x_i) = .5$ for $x_i \in \{0, 1\}$, and 0 otherwise), and has objective value $|E||V|$. Note that μ and δ also satisfy the complementary slackness conditions (as they must, since they are a primal-dual optimal pair). □

4 Bound Criterion in Sparse Graphs

One potential approach for using the bound criterion from [3] on a sparse graphical model (which may not have *any* short cycles) is to first triangulate the graph before running the algorithm. Of course, for a sparse graphical model with large treewidth, this could result in a very large number of new edges being added, which would significantly increase the running time for minimizing the dual. In this section we show that, even if we had done this, the bound criterion can be non-informative in these settings, and thus triangulation may not even be helpful.

Consider a binary-valued MRF on four variables X_1, X_2, X_3, X_4 which has edges in the form of a square: $E = \{(1,2),(2,3),(3,4),(1,4)\}$. We now define the edge potentials. For $(i,j) \in \{(1,2),(2,3),(3,4)\}$, let $\theta_{ij}(x_i, x_j) = 1$ if $x_i \neq x_j$, and 0 if $x_i = x_j$. We do the opposite for edge $(1,4)$, letting $\theta_{1,4}(x_1, x_4) = 1$ if $x_1 = x_4$, and 0 if $x_1 \neq x_4$.

All of the MAP assignments have value 3. For example, one MAP assignment is $(X_1, X_2, X_3, X_4) = (1,0,1,0)$. The pairwise LP relaxation, on the other hand, has value 4, obtained by $\mu_i(x_i) = 0.5$ for $i \in \{1,2,3,4\}$ and $x_i \in \{0,1\}$, $\mu_{ij}(0,1) = \mu_{ij}(1,0) = 0.5$ for $(i,j) \in \{(1,2),(2,3),(3,4)\}$, and $\mu_{1,4}(0,0) = \mu_{1,4}(1,1) = 0.5$. One way to triangulate the graph is to add the edge $(1,3)$. However, as can be seen by setting the edge marginal for $(1,3)$ to $\mu_{1,3}(1,1) = \mu_{1,3}(0,0) = 0.5$ (note that this satisfies the pairwise consistency constraints), the pairwise LP relaxation still has value 4.

Now consider adding a triplet consistency constraint for the cluster $c = \{1,2,3\}$. Solving the new LP relaxation, we find that it again has value 4. The solution is the same as earlier, with the new triplet marginal taking value $\mu_{1,2,3}(1,0,1) = \mu_{1,2,3}(0,1,0) = 0.5$ (note that this is consistent with the edge marginals already given, as it should be). Let's see what this corresponds to in the dual. Suppose that we solve the dual of the pairwise LP relaxation to optimality. By LP duality, the dual objective has value 4. Also by LP duality, we know that the optimal dual objective *after* adding the triplet cluster must also be 4. Recall that the bound criterion $d(c)$ corresponds to the amount that the dual objective will decrease after one block coordinate descent step involving the new cluster. Since the dual objective is lower bounded by 4 (its value at optimality), we conclude that $d(c)$ must be zero for the triplet cluster $c = \{1,2,3\}$.

The same can be shown for the triplet cluster $c = \{1,3,4\}$. A generalization of the argument shows that, for a MRF that consists of a single cycle of length larger than 3, and for any dual optimal solution of the pairwise LP relaxation (after triangulation), $d(c) = 0$ for all of the triplets in the triangulation. In contrast, if we had evaluated $d(c)$ for c corresponding to the *whole cycle* we would see that it is non-zero.

5 Algorithm FindPartition

In this section, we give an algorithm to find partitions of each variable's states, giving a tractable approach to go beyond the k-projection graph in the case of non-binary Markov networks. The algorithm FindPartition is given in Figure 2 and illustrated in Figure 3. First, for each variable and each state, it adds to the projection graph the corresponding one-versus-all partition – that is, we start with the k-projection graph. Then, for each each edge $ij \in E$, it finds a partition for variable i and for variable j that maximizes Eq. 10 for the single edge belief $b_{ij}(x_i, x_j)$, and adds both partitions to the projection graph (if they do not already exist). To be efficient, the algorithm makes use of the Union-Find data structure [1].

For an edge ij, FindPartition finds a partition $\pi_i^q(x_i)$ and $\pi_j^r(x_j)$ that maximizes $\max_{x_i,x_j:\pi_i^q(x_i)=\pi_j^r(x_j)} b_{ij}(x_i, x_j)$. It can be shown that these partitions at the same time minimize $\max_{x_i,x_j:\pi_i^q(x_i)\neq\pi_j^r(x_j)} b_{ij}(x_i, x_j)$ and thus it suffices to consider just the maximization problem.

First, note that either $\max_{x_i,x_j:\pi_i^q(x_i)=\pi_j^r(x_j)} b_{ij}(x_i, x_j)$ or $\max_{x_i,x_j:\pi_i^q(x_i)\neq\pi_j^r(x_j)} b_{ij}(x_i, x_j)$ will be equal to $\max_{x_i,x_j} b_{ij}(x_i, x_j)$ regardless of what the partitions are, since the argmax states x_i, x_j necessarily have either $\pi_i^q(x_i) = \pi_j^r(x_j)$ or $\pi_i^q(x_i) \neq \pi_j^r(x_j)$. Thus, the optimal first choice is to assign the highest weight states to the same partition. This argument is then repeated. With sorted edges in decreasing $b_{ij}(x_i, x_j)$ order, Union-Find sequentially adds each pair of states x_i, x_j to the same partition. When Union-Find cannot add a pair of states to the same partition (this happens when doing so would create one partition with all states of one of the nodes, which would be invalid), the optimal partition is found. The remaining

FindPartition ({beliefs $b_{ij}(x_i, x_j)$})

Initialize G_π to have one-versus-all partitions for every variable and state.
for each edge ij, where $x_i \in \{1, \ldots, m\}$, $x_j \in \{1, \ldots, n\}$:
    Initialize M to be the collection of all possible states of node i and node j, namely :
        $M \leftarrow \{\{(i, x_i)\} : x_i \in \{1, \ldots, m\}\} \cup \{\{(j, x_j)\} : x_j \in \{1, \ldots, n\}\}$
    Define list L as $\langle((i, u_1), (j, v_1), b_{ij}(u_1, v_1)), \ldots, ((i, u_{mn}), (j, v_{mn}), b_{ij}(u_{mn}, v_{mn})\rangle$ given by sorting
        $\{(x_i, x_j, b_{ij}(x_i, x_j)) : x_i \in \{1, \ldots, m\}, x_j \in \{1, \ldots, n\}\}$ by $b_{ij}(x_i, x_j)$ in descending order
    for $k = 1$ to mn :
        find (as in union-find) $m_1, m_2 \in M$ such that $(i, u_k) \in m_1$, $(j, v_k) \in m_2$
        if $m_1 \neq m_2$ then :
            if replacing m_1, m_2 with $m_1 \cup m_2$ in M causes all states of i or j to be in 1 partition then :
                $A \leftarrow m_1, B \leftarrow \Big((\{i\} \times \{1, \ldots, m\}) \cup (\{j\} \times \{1, \ldots, n\})\Big) \setminus m_1$
                break
            remove m_1, m_2 from M and add $m_1 \cup m_2$ to M

    Add partitions defined by A, B to G_π.
Return G_π.

Figure 2: Pseudocode of FindPartition algorithm.

states are assigned randomly (in practice we canonically assign them to the largest partition, to allow for us to re-use identical partitions that are found using different edges). When FindPartition is finished, each variable i will have at most $k_i + N_i$ partitions where k_i is its cardinality and N_i is its number of neighbors.

References

[1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

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

[3] David Sontag, Talya Meltzer, Amir Globerson, Yair Weiss, and Tommi Jaakkola. Tightening LP relaxations for MAP using message-passing. In *UAI*, pages 503–510. AUAI Press, 2008.

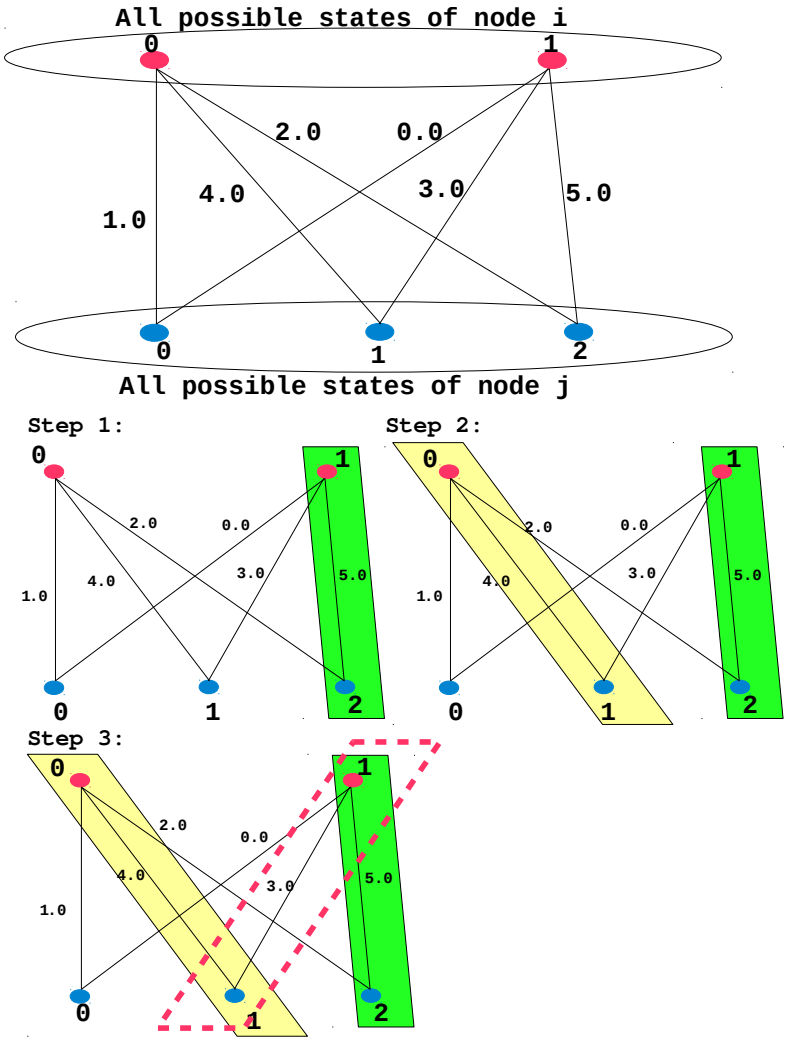

Figure 3: Illustration of the algorithm FindPartition for optimally choosing partitions with respect to a single edge. In this example, node i has 2 states, node j has 3 states, and the $b_{ij}(x_i, x_j)$ values are $0.0, 1.0, \ldots, 5.0$. The algorithm first merges $\{(i,1)\}$ with $\{(j,2)\}$ (as shown in green), and then $\{(i,0)\}$ with $\{(j,1)\}$ (as shown in yellow), before considering merging $\{(i,1)\}$ with $\{(j,1)\}$ (as shown with the red dashes). Because merging $\{(i,1)\}$ with $\{(j,1)\}$ would cause an invalid partitioning of states on node i, and in this case $m_2 = \{(i,0),(j,1)\}$, $m_1 = \{(i,1),(j,2)\}$ (yellow and green, respectively), the algorithm terminates with 2 partitions $\{(i,0),(j,1)\}$ and its complement $\{(i,1),(j,0),(j,2)\}$.