Graph cuts always find a global optimum for Potts models (with a catch)

Hunter Lang [1] **David Sontag** [1] **Aravindan Vijayaraghavan** [2]

Abstract

We prove that the α-expansion algorithm for MAP inference *always* returns a globally optimal assignment for Markov Random Fields with Potts pairwise potentials, *with a catch*: the returned assignment is only guaranteed to be optimal for an instance within a small perturbation of the original problem instance. In other words, all local minima with respect to expansion moves are global minima to slightly perturbed versions of the problem. On "real-world" instances, MAP assignments of small perturbations of the problem should be very similar to the MAP assignment(s) of the original problem instance. We design an algorithm that can certify whether this is the case in practice. On several MAP inference problem instances from computer vision, this algorithm certifies that MAP solutions to *all* of these perturbations are very close to solutions of the original instance. These results taken together give a cohesive explanation for the good performance of "graph cuts" algorithms in practice. Every local expansion minimum is a global minimum in a small perturbation of the problem, and all of these global minima are close to the original solution.

1. Introduction

Markov random fields are widely used for structured prediction in computer vision tasks such as image segmentation and stereopsis (Geman & Geman, 1984), including in the modern "deep" era (e.g., Zheng et al., 2015). Making predictions involves performing MAP inference. However, in general, exactly solving the MAP inference problem is NP-hard (Wainwright & Jordan, 2008) and one must resort to approximate inference.

"Graph cuts" algorithms for approximate MAP inference in pairwise Markov random fields have been very influential in computer vision. These algorithms are popular because they are simple and efficient, and they return very high-quality solutions in practice (Szeliski et al., 2008; Kappes et al., 2015). The α-*expansion* method of Boykov et al. (2001) starts with an arbitrary initial labeling (an assignment of labels to variables), then iteratively makes "expansion moves" to improve the current labeling. At each step, the optimal expansion move of the current labeling can be computed very efficiently by solving a binary minimum cut problem (hence the name "graph cuts"). The algorithm converges when no expansion moves can improve the labeling any further.

Algorithm 1 summarizes the high-level algorithm steps. The α-expansion algorithm is only guaranteed to return a *local minimum* with respect to the moves made by the algorithm. Figure 1 shows a globally optimal (MAP) labeling, which took over four hours to obtain with an integer linear programming (ILP) solver, and the local minimum returned by α-expansion in less than ten seconds. Although α-expansion is only guaranteed to find a local minimum, the two assignments agree on over 99% of the vertices.

Despite this good practical performance, the sharpest worst-case theoretical guarantee for α-expansion is that it obtains a 2-approximation to the objective value of the MAP labeling (Boykov et al., 2001). A 2-factor objective approximation often translates to a very weak guarantee for recovering the exact solution: Lang et al. (2019) show that MAP inference problems from computer vision admit 2-approximate labelings that agree with the optimal assignment on fewer than 1% of variables. Compare this to Figure 1, where the expansion solution agrees with the exact solution on over 99% of the nodes. Additionally, objective gap bounds obtained from primal-dual variants of α-expansion are sometimes very close to one in practice (Komodakis et al., 2007). Those bounds (which depend on the algorithm's initialization) show that graph-cuts algorithms often vastly outperform their theoretical guarantees. So a large gap exists between the worst-case guarantee of 2-approximation and the practical performance (in both objective value and recovery of the true solution) of the α-expansion algorithm. *Why are the local minima with respect to expansion moves so good in practice?*

In this work, we prove a surprising structural result that

[1]MIT CSAIL, Cambridge MA, USA [2]Northwestern University, Evanston IL, USA. Correspondence to: Hunter Lang <hjl@mit.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

Algorithm 1 α-expansion algorithm

Initialize a labeling $x : V \to [k]$.
`improved` $\leftarrow$ `True`.
while `improved` **do**
  `improved` $\leftarrow$ `False`
  for $\alpha \in [k]$ **do**
    $x^\alpha \leftarrow$ optimal α-expansion of x.
    if $obj(x^\alpha) < obj(x)$ **then**
      $x \leftarrow x^\alpha$.
      `improved` $\leftarrow$ `True`.
    end if
  end for
end while
return x.

Figure 1. Left: image of `venus` scene. Center: exact MAP depth labeling, found using ILP solver; Left: a local minimum w.r.t. expansion moves. The two labelings agree on over 99% of nodes.

characterizes the local energy minima with respect to (w.r.t.) expansion moves. Informally, we prove that for a widely used model (the *Potts model*) all local expansion minima are actually *global* minima of slightly perturbed instances of the input problem. This result implies that the α-expansion algorithm *always* returns a MAP assignment—the "catch" is that the assignment is *not* guaranteed to be optimal for the input instance, but rather for some closely-related instance. In other words, we prove that when we run α-expansion on an instance I, it always outputs a MAP solution to an instance I' that is a small perturbation of I.

Our result implies that real-world instances should have no "spurious" local minima with respect to expansion moves. This is because in practice, MAP solutions to all small perturbations of the problem instance should be very close to the MAP solution of the original instance. We design an efficient algorithm to check whether this is truly the case. On real-world instances of MAP inference from computer vision, our algorithm certifies that all solutions to these perturbations are very close to the solution of the original instance. Our results thus give a theoretical explanation for the excellent empirical performance of α-expansion and related graph cut methods like FastPD (Komodakis et al., 2007). These algorithms naturally take advantage of the fact that solutions to all small perturbations tend to be close to the original solution in practice.

2. Preliminaries

Before we discuss related work, we formally introduce the inference problem considered in this paper and fix notation. Fix a constant k and a graph $G = (V, E)$ with $|V| = n$, $|E| = m$. A *labeling* of G is a map $g : V \to [k]$. The (pairwise) MAP inference problem on G can be written:

$$\underset{g:V\to[k]}{\text{minimize}} \sum_{u\in V} \theta_u(g(u)) + \sum_{(u,v)\in E} \theta_{uv}(g(u), g(v)).$$

In this *energy minimization* format, $\theta_u(i)$ is the *node cost* of assigning label $i \in \{1, \dots, k\}$ to node u, and $\theta_{uv}(i, j)$ is the *edge cost* or *pairwise energy* of simultaneously assigning label i to u and j to v. We assume without loss of generality that $\theta_u(i) \geq 0$ for all (u, i). Consider image segmentation: the nodes $u \in V$ correspond to image pixels, and the edges $(u, v) \in E$ connect nearby pixels. The node costs $\theta_u(i)$ can be set as the negative score of pixel u for segment i, and the pairwise terms $\theta_{uv}(i, j)$ can be set to encourage nearby pixels to belong to the same segment.

We can identify each labeling $g : V \to L$ with a point $x^g \in \{0, 1\}^{nk+mk^2}$ defined by the following indicator functions:

$$x^g_u(i) = \begin{cases} 1 & g(u) = i \\ 0 & \text{otherwise.} \end{cases}$$

$$x^g_{uv}(i, j) = \begin{cases} 1 & g(u) = i,\ g(v) = j \\ 0 & \text{otherwise.} \end{cases}$$

The MAP inference problem can then be written as:

$$\min_{g:V\to[k]} \sum_{u\in V} \sum_{i\in[k]} \theta_u(i) x^g_u(i) + \sum_{uv\in E} \sum_{i,j} \theta_{uv}(i, j) x^g_{uv}(i, j).$$

The *marginal polytope* $M(G)$ is defined as the convex hull of all x^g:

$$M(G) \triangleq \text{conv}\left(\{x^g |\ g : V \to [k]\}\right).$$

We can denote the coordinates of an arbitrary point in $x \in M(G)$ as $x_u(i)$, $x_{uv}(i, j)$, with $u \in V$, $uv \in E$, and i and j in $[k]$. Collecting the objective coefficients $\theta_u(i)$ and $\theta_{uv}(i, j)$ in the vector $\theta = (\theta_u : u;\ \theta_{uv} : uv) \in \mathbb{R}^{nk+mk^2}$, we can rewrite MAP inference as a linear optimization over $M(G)$:

$$\underset{x\in M(G)}{\text{minimize}} \langle \theta, x \rangle,$$

since the vertices of this polytope are precisely the points x^g corresponding to labelings of G. $M(G)$ typically lacks an efficient description, and optimizing a linear function over it is NP-hard in general (Wainwright & Jordan, 2008). The minimization problem above can be represented as the

integer linear program (ILP):

$$\begin{aligned} \underset{x}{\text{minimize}}\ & \langle \theta, x \rangle && (1)\\ \text{subj. to: } & \sum_i x_{uv}(i,j) = x_v(j) && \forall (u,v) \in E,\ j \in [k]\\ & \sum_j x_{uv}(i,j) = x_u(i) && \forall (u,v) \in E,\ i \in [k]\\ & \sum_i x_u(i) = 1 && \forall u \in V\\ & x_{uv}(i,j) \in \{0,1\} && \forall (u,v),\ (i,j)\\ & x_u(i) \in \{0,1\} && \forall u,\ i. \end{aligned}$$

The feasible points of this ILP are precisely the vertices of $M(G)$. A common approximate approach to MAP inference is to solve the following linear programming (LP) relaxation:

$$\underset{x \in L(G)}{\text{minimize}}\ \langle \theta, x \rangle, \tag{2}$$

where $L(G)$ is the *local polytope* defined by relaxing all the integrality constraints above from $\{0,1\}$ to $[0,1]$. In general, $M(G)$ is a strict subset of $L(G)$. The first two sets of constraints are called *marginalization constraints*, and ensure the edge variables $x_{uv}(i,j)$ locally match the "marginals" $x_u(i)$ and $x_v(j)$. The third set consists of *normalization* constraints that ensure the x_u variables sum to 1. Note that there are some redundant constraints in this formulation. In the LP relaxation, the variables $x_u(i)$, $x_{uv}(i,j)$ correspond to potentially *fractional* labelings of G, since we only have that $\sum_i x_u(i) = 1$.

We refer to (2) as the *local LP relaxation* of the MAP inference problem. This relaxation has been widely studied (Sontag, 2010; Wainwright & Jordan, 2008), including in the context of stability and the ferromagnetic Potts model (Kleinberg & Tardos, 2002; Lang et al., 2018; 2019; 2021). Many algorithms for approximate MAP inference can be related to this relaxation (e.g., MPLP (Globerson & Jaakkola, 2008) performs coordinate ascent in its dual), but we only use it here as a tool in our analysis. We say that (2) is *tight* on an instance of the MAP inference problem if there exists a vertex of $M(G)$ that is a solution to (2). This optimal vertex must correspond to an exact MAP labeling.

In this work, we focus on the *ferromagnetic Potts model*, where the pairwise terms $\theta_{uv}(i,j) = w_{uv}\mathbb{I}[i \neq j]$, with $w_{uv} \geq 0$. That is, the cost of an edge only depends on whether the labels of its endpoints match. While seemingly simple, this model is popular in practice (for example, it accounts for several of the instances studied in Kappes et al. (2015)). We still use $\theta_{uv}(i,j)$ in what follows for notational convenience, but in the rest of our results, we assume $\theta_{uv}(i,j)$ takes this form. MAP inference in this model is also called *uniform metric labeling* (Kleinberg & Tardos, 2002), and it is NP-hard for variable $k \geq 3$, even when G is planar (Dahlhaus et al., 1992).

We often use the same symbol x to refer to both a vertex of $M(G)$, referencing the values $x_u(i)$, $x_{uv}(i,j)$, and to a labeling of G, referencing $x(u)$. This is justified because these two objects are in one-to-one correspondence. For example, we write the objective value of a labeling $x : V \to [k]$ as $\langle \theta, x \rangle$. We also define the (normalized) *Hamming distance* between two labelings x and x' as:

$$\frac{1}{n}\sum_u \mathbb{I}[x(u) \neq x'(u)] = \frac{1}{2n}\sum_u \sum_i |x_u(i) - x'_u(i)|.$$

Finally, for a fixed graph G and a fixed k, we identify an instance of the MAP inference problem with its objective vector $\theta = (\theta_u : u, \theta_{uv} : uv)$.

2.1. Expansion

Let $x : V \to [k]$ be a labeling of G. For any label $\alpha \in [k]$, we say that x' is an α-expansion of x if the following hold for all $u \in V$:

$$\begin{aligned} x(u) = \alpha &\implies x'(u) = \alpha,\\ x'(u) \neq \alpha &\implies x'(u) = x(u). \end{aligned}$$

That is, x' may not shrink the region of nodes labeled α—that region can only expand—and if x' changes any label, the new label must be α. The optimal α-expansion move of x can be found very efficiently by solving a minimum cut problem in an auxiliary graph $G^x(\alpha)$ (Boykov et al., 2001). Algorithm 1 starts with an arbitrary labeling $x : V \to [k]$, then iteratively improves it by making expansion moves. The algorithm converges when there are no expansion moves that decrease the objective $\langle \theta, x \rangle$. We say a labeling x is a local minimum w.r.t. expansion moves if no expansion move of x strictly decreases the objective.

The approximation guarantee for α-expansion states that the objective of any local minimum is at most the objective of the MAP solution x^* plus the edge cost paid by x^*.

Theorem 1 ((Boykov et al., 2001) Theorem 6.1)**.** *Let x be a local minimum w.r.t. expansion moves. Then*

$$\langle \theta, x \rangle \leq \langle \theta, x^* \rangle + \sum_{uv} \theta_{uv}(x^*(u), x^*(v))$$

In particular, $\langle \theta, x \rangle \leq 2\langle \theta, x^* \rangle$.

3. Related work

3.1. Perturbation stability

Lang et al. (2018) define $(1,2)$-stable instances of uniform metric labeling as those instances whose MAP solution does not change when any subset of edges $S \subset E$ can have the

weights w_{uv} multiplied by an edge-dependent $\gamma_{uv} \in [1, 2]$. They prove that α-expansion recovers the exact MAP solution on $(1, 2)$-stable instances. As is typically the case in work on perturbation stability, few guarantees are given for instances that do not satisfy the stability definition. Unfortunately, the real-world instances that motivated Lang et al. (2018)'s work are not stable—the requirement of stability that the solution doesn't change *at all* turns out to be too strict to be practical (Lang et al., 2019). Our results are much more general, since they apply to *any* instance (stable or not). To go beyond stability, Lang et al. (2019) showed that an LP relaxation has approximate recovery guarantees when "blocks" (sub-instances) of the instance are perturbation stable, and Lang et al. (2021) showed that perturbation-stable instances are still approximately solvable after being corrupted by noise. Neither of these works gives a guarantee for graph cuts.

3.1.1. Checking stability

Lang et al. (2019) designed algorithms for checking stability and sub-instance stability for uniform metric labeling that are based on solving (a series of) integer linear programs. Surprisingly, we show that our algorithm, which bounds the performance of all possible α-expansion minima, is *computationally efficient* once an exact MAP solution x^* is known.

3.2. Primal-dual graph cut algorithms

Komodakis et al. (2007) showed how to interpret α-expansion as a primal-dual algorithm for solving the energy minimization problem. This view enables the algorithm to compute certificates of (sub)optimality at essentially no extra cost, so bounds on the gap between the objective of the labeling returned by expansion and the optimal objective can be efficiently obtained in practice. Unlike our results, these bounds are initialization-dependent, and they only bound the objective value (they do *not* bound the difference from the global minimum itself). Our structural result can be taken as an *explanation* for (i) why these objective bounds tend to be close to 1 regardless of the initialization and (ii) why the returned labelings have small Hamming distance to the optimal labeling: all local minima w.r.t. expansion moves are global minima for some instance within a small perturbation of the input. On practical instances, solutions to small perturbations tend to have near-optimal objective in the original instance and are close in Hamming distance to the original solution.

3.3. Partial optimality results for α-expansion

A node/label pair (u, i) is a *partially optimal assignment* (henceforth, a *partopt*) if $x^*(u) = i$ for the MAP solution x^*. Several works have developed fast algorithms for finding provable partopts, i.e. identifying parts of the MAP assignment (e.g., Kovtun, 2003; Shekhovtsov, 2013; Swoboda et al., 2016; Shekhovtsov et al., 2017). Shekhovtsov & Hlavac (2011) showed that if Kovtun's procedure outputs a partopt (u, i), then any expansion minimum x^α has $x^\alpha(u) = i$. Like our result, this gives a guarantee for α-expansion that is independent of the algorithm's initialization: expansion always recovers $x^*(u)$ when Kovtun's procedure finds the optimal label at a vertex u. However, this result does not explain *when* Kovtun's procedure finds a large number of partopts. In contrast, our results only rely on a structural property of the *instance* itself (that the solutions to perturbations are close to the solutions of the original). Moreover, our algorithm in Section 5 for bounding α-expansion's Hamming error is meant to illustrate the tightness of our structural result, not to give a fast method for finding provably partially optimal assignments.

3.4. Certified algorithms

Our results are very related to the study of *certified* algorithms (Makarychev & Makarychev, 2020; Angelidakis et al., 2019). Informally, a *certified* algorithm is one that returns a global (exact) solution to a perturbation of the input problem. We prove that α-expansion is a certified algorithm for uniform metric labeling. Our algorithm in Section 5 for upper bounding α-expansion's error could be used to upper bound the error of other certified algorithms. The fact (proven here) that a popular algorithm with a long track record of empirical success is a certified algorithm suggests that this model could be useful for understanding the empirical performance of algorithms on hard problems. Exact solutions to small perturbations of the input can be efficiently obtained despite hardness of the original problem, and these exact solutions are often very close to those of the original problem in practice.

4. Expansion always finds a global optimum

In this section, we give our main theoretical results. Theorem 2 states that every labeling x that is a local minimum w.r.t. expansion moves is a global minimum (an exact MAP solution) in a perturbed version of the input problem instance. Theorem 3 then gives a precise characterization of a perturbation in which x is optimal. The simple structure of these perturbations is useful in the development of our algorithm in Section 5. We defer both proofs to Appendix A.

Theorem 2. *Let the labeling x be a local minimum with respect to expansion moves for the instance with objective θ. Let $\mathcal{I}(\theta)$ be the set of θ' that for some $\gamma \in [1, 2]^{|E|}$ satisfy:*

$$\begin{aligned} \theta'_u &= \theta_u && \forall u \in V \\ \theta'_{uv} &= \gamma_{uv}\theta_{uv} && \forall (u, v) \in E \end{aligned} \tag{3}$$

Then there exists $\theta' \in \mathcal{I}(\theta)$ for which x is a MAP solution.

The definition of $\mathcal{I}(\theta)$ requires that each $\theta' \in \mathcal{I}(\theta)$ has exactly the same node costs as θ, and that the pairwise potentials $\theta'_{uv} = \gamma_{uv}\theta_{uv}$ for an edge-dependent constant $\gamma_{uv} \in [1, 2]$. Theorem 2 says that for each local minimum x to the input instance θ, there exists at least one instance $\theta' \in \mathcal{I}(\theta)$ for which x is a global minimum. The next theorem gives a closed form for one such θ' in terms of x.

Theorem 3. *Given an instance θ with edge weights w_{uv} and an expansion minimum x for θ, define perturbed weights w^x_{uv}:*

$$w^x_{uv} = \begin{cases} w_{uv} & x(u) \neq x(v) \\ 2w_{uv} & x(u) = x(v), \end{cases} \tag{4}$$

and let

$$\theta^x_{uv}(i, j) = w^x_{uv}\mathbb{I}[i \neq j] \tag{5}$$

be the pairwise Potts energies corresponding to the weights w^x. Then x is a global minimum in the instance with objective vector $\theta^x = (\theta_u : u;\ \theta^x_{uv} : uv)$. This is the Potts model instance with the same node costs $\theta_u(i)$ as the original instance, but new pairwise energies $\theta^x_{uv}(i, j)$ defined using the perturbed weights w^x. Note that $\theta^x \in \mathcal{I}(\theta)$. Additionally, the LP relaxation (2) is tight on this perturbed instance.

Theorem 2 strictly and significantly generalizes Theorem 2 of Lang et al. (2018), since $\mathcal{I}(\theta)$ is the set of $(1, 2)$-perturbations of the input instance θ. The analysis is similar to that in Lang et al. (2018), but reinterpreted through the certified algorithm lens of Makarychev & Makarychev (2020). The guarantee of Theorem 3 that the LP relaxation (2) is tight in the perturbed instance is crucial to our algorithm in Section 5. Theorems 2 and 3 also apply to any iterative algorithm whose set of iterative moves contain the set of expansion moves, such as FastPD.

5. How "bad" are solutions to perturbations?

Theorem 2 guarantees that when α-expansion is run on an instance θ, it always returns a MAP solution to some instance $\theta' \in \mathcal{I}(\theta)$. To evaluate how informative this guarantee is, we need a method to find the "worst" solution out of all the solutions to instances in $\mathcal{I}(\theta)$. That is, let

$$\mathcal{S}(\theta) = \{x : x \text{ a MAP solution for some } \theta' \in \mathcal{I}(\theta)\}. \tag{6}$$

Theorem 2 implies all local expansion minima x have $x \in \mathcal{S}(\theta)$. This structural condition is informative for an instance θ if every solution in $\mathcal{S}(\theta)$ is close to the MAP solution x^* of θ. In that case, our result *explains* why α-expansion always performs well. Our hypothesis is that real-world instances should have this property, but we need a method to verify this hypothesis empirically.

In this section, we design an efficient algorithm for upper-bounding the value of any concave function $f(x)$ over $\mathcal{S}(\theta)$. For example, $f(x)$ could measure the Hamming distance to x^* or the objective gap of x in the original instance θ. Note that in these cases, the algorithm must be given x^* to compute $f(x)$, but this need not be true for general f. For example, in a *learning* scenario, $f(x)$ could measure Hamming distance to the known ground-truth assignment. Because all local expansion minima are contained in $S(\theta)$ by Theorem 2, bounds for these quantities give initialization-independent bounds on expansion's performance.

Formally, we want to solve

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f(x) \\ \text{subject to} \quad & x \in \mathcal{S}(\theta). \end{aligned} \tag{7}$$

Here $f(x)$ is a concave function that measures the "badness" of x. For example, if x^* is a MAP solution to the original instance, we could take $f(x) = \langle\theta, x\rangle / \langle\theta, x^*\rangle$, the objective gap of x. Similarly, we can let $f(x)$ be the Hamming distance between x and x^*, which can be expressed as:

$$f(x) = \frac{1}{2n}\left(\sum_{u\in V}\sum_{i\neq x^*(u)} x_u(i) - \sum_{u\in V} x_u(x^*(u)) + n\right),$$

where we are taking x^* as a labeling and x as a point of $M(G)$. This is an affine function of x. Let η be the optimal value of (7). Then by Theorem 2, all local expansion minima x satisfy $f(x) \leq \eta$. Solving (7) thus gives an upper bound on the error of all expansion minima. For simplicity, and because we use the two functions above in our empirical results, we assume in what follows that $f(x)$ is affine. We can then replace maximization of $f(x)$ with maximization of $\langle f, x\rangle$ for some vector f. However, our algorithm works for any concave function.

In the rest of this section, we design an efficient algorithm for upper-bounding the optimal value of (7) by deriving a sequence of equivalent problems, then performing a convex relaxation. In several of our experiments in Section 6, we find that the bound obtained by our algorithm is nearly tight.

Theorem 4. *For affine functions f, (7) can be exactly represented by an integer linear program (ILP). Additionally, an upper bound on the optimal value of (7) can be obtained efficiently using a linear program.*

Proof. As written, (7) is difficult to optimize because it searches over the set $\mathcal{S}(\theta)$ of MAP solutions to instances $\theta' \in \mathcal{I}(\theta)$. This is not a convex set. First, the following lemma gives a simpler characterization of $\mathcal{S}(\theta)$.

Lemma 1.

$$\mathcal{S}(\theta) = \{x : x \textit{ a MAP solution to the instance } \theta^x \textit{ defined by (4) and (5)}\}$$

Proof. We show in Appendix A that if x is optimal for any $\theta' \in \mathcal{I}(\theta)$, x is optimal for θ^x. This immediately gives the result. □

So we can rewrite (7) as:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & f(x) \\ \text{subject to} \quad & x \text{ a vertex of } M(G), \\ & x \text{ optimal in the instance} \\ & \text{with objective } \theta^x. \end{aligned} \tag{8}$$

The first constraint ensures that x is a valid labeling, and the second constraint ensures (by Lemma 1) that $x \in \mathcal{S}(\theta)$. Now we focus on simplifying the optimality constraint. Let x be an optimal labeling in the instance with objective θ^x. By Theorem 3, for all LP-feasible points $y \in L(G)$, we have that $\langle \theta^x, x \rangle \leq \langle \theta^x, y \rangle$. That is, the local LP relaxation is tight on this instance—even though x is an integer solution, its objective value $\langle \theta^x, x \rangle$ is as good as that of any fractional solution. This allows us to rewrite the optimality constraint using the following valid constraint:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & \langle f, x \rangle \\ \text{subject to} \quad & x \text{ a vertex of } M(G) \\ & \langle \theta^x, x \rangle \leq \min_{y \in L(G)} \langle \theta^x, y \rangle, \end{aligned}$$

Note that if the local LP relaxation were not tight on the instance with objective θ^x (as guaranteed by Theorem 3), this constraint would be invalid. Even with this simplification, the dependence of θ^x on x makes it unclear whether the new constraint is convex. Because x is a vertex of $M(G)$, it only takes values in $\{0, 1\}$, so we can rewrite w^x_{uv} from (4) as a linear function of x:

$$w^x_{uv} = w_{uv} + w_{uv} \left(1 - \sum_{i \neq j} x_{uv}(i, j) \right).$$

This is because $\sum_{i \neq j} x_{uv}(i, j)$ is 0 if $x(u) = x(v)$ and 1 otherwise. Then, because w^x_{uv} is a linear function of x, $\langle \theta^x, y \rangle$ is a linear function of x for each $y \in L(G)$. Additionally, observe that because $x \in M(G)$, (5) implies $\langle \theta^x, x \rangle = \langle \theta, x \rangle$: the perturbed objective of x is equal to its original objective (note, however, $y \neq x$ may have $\langle \theta^x, y \rangle \neq \langle \theta, y \rangle$). Using these simplifications, we can solve the following equivalent problem:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & \langle f, x \rangle \\ \text{subject to} \quad & x \text{ a vertex of } M(G) \\ & \langle \theta, x \rangle \leq \min_{y \in L(G)} \langle \theta^x, y \rangle. \end{aligned}$$

Because we have shown how to re-write $\langle \theta^x, y \rangle$ as a linear function of x and removed θ^x from the left-hand-side, the second constraint is convex. However, two barriers remain to solving this problem efficiently: (i) the optimality constraint $\langle \theta, x \rangle \leq \min_{y \in L(G)} \langle \theta^x, y \rangle$ is not in a convenient form, and (ii) the first constraint is not convex. We address (i) first.

For ease of notation, define A and b so that the local polytope $L(G) = \{x | Ax = b,\ x \geq 0\}$. Because strong duality holds for the local LP relaxation in the instance with objective θ^x, we know that

$$\min_{y : Ay = b, y \geq 0} \langle \theta^x, y \rangle = \max_{\nu : A^T \nu \leq \theta^x} \langle b, \nu \rangle.$$

Indeed, if there exists any feasible y, ν pair for which $\langle \theta^x, y \rangle = \langle b, \nu \rangle$, y and ν are optimal primal and dual solutions, respectively. We want to enforce the constraint that x is primal-optimal in the instance with objective θ^x, which is the case if and only if there exists a dual-feasible ν with $\langle \theta^x, x \rangle = \langle \theta, x \rangle = \langle b, \nu \rangle$. So we can rewrite the problem as

$$\begin{aligned} \underset{x, \nu}{\text{maximize}} \quad & \langle f, x \rangle \\ \text{subject to} \quad & x \text{ a vertex of } M(G) \\ & \langle \theta, x \rangle = \langle b, \nu \rangle \\ & A^T \nu \leq \theta^x. \end{aligned}$$

Because θ^x is a linear function of x, the latter two constraints are linear in x and ν. Together with linearizing θ^x and noting $\langle \theta^x, x \rangle = \langle \theta, x \rangle$, this primal-dual trick allowed us to encode the second constraint of (8) as two sets of linear constraints. This trick heavily relies on the guarantee from Theorem 3 that the local LP is tight on the instance with objective θ^x.

The only remaining issue is the first constraint, that x is a vertex of $M(G)$. We saw in Section 2 how to encode the vertices of $M(G)$ using linear and integrality constraints, so we can rewrite the above problem as the ILP:

$$\begin{aligned} \underset{x, \nu}{\text{maximize}} \quad & \langle f, x \rangle \\ \text{subject to} \quad & x \in L(G) \\ & x_u(i) \in \{0, 1\} \\ & x_{uv}(i, j) \in \{0, 1\} \\ & \langle \theta, x \rangle = \langle b, \nu \rangle \\ & A^T \nu \leq \theta^x. \end{aligned} \tag{9}$$

Unfortunately, this ILP is too large for off-the-shelf ILP solvers to handle in practice. Instead, we relax this exact formulation to obtain upper bounds.

In particular, we iteratively solve the following optimization problem:

$$\begin{aligned} \underset{x, \nu}{\text{maximize}} \quad & \langle f, x \rangle \\ \text{subject to} \quad & x \in K_t \\ & \langle \theta, x \rangle = \langle b, \nu \rangle \\ & A^T \nu \leq \theta^x, \end{aligned} \tag{10}$$

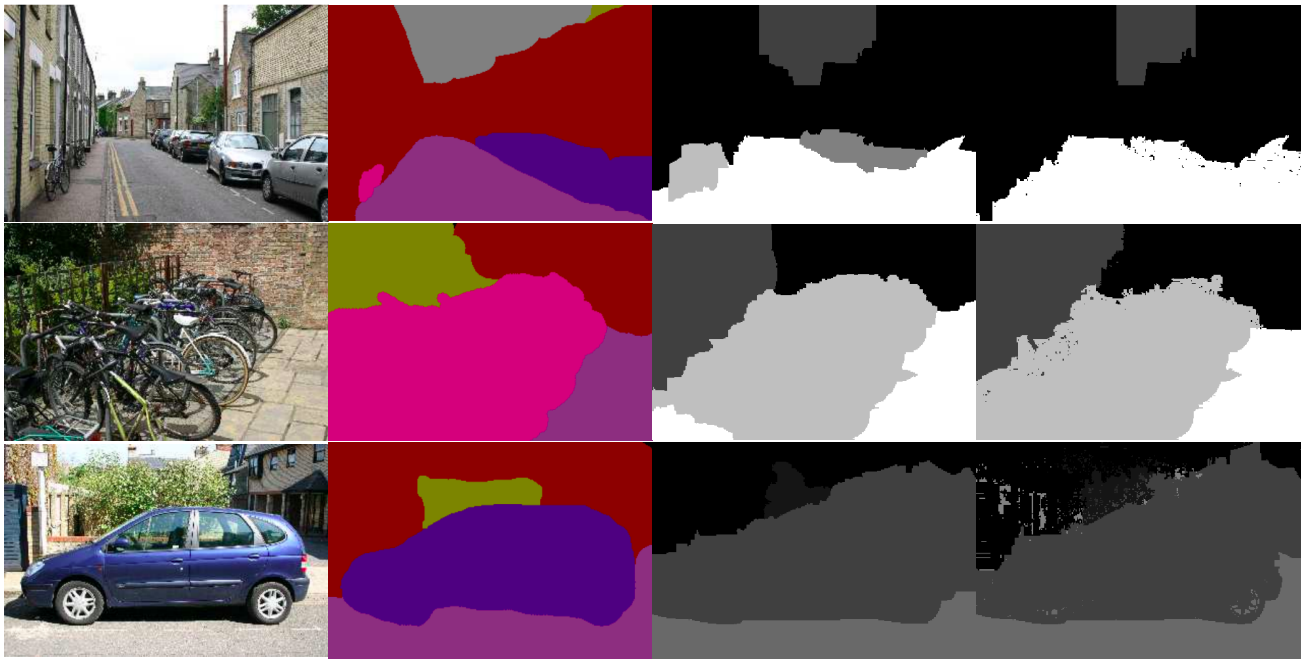

Figure 2. Left column: original image; Center-left: ground-truth segmentation map; Center-right: exact MAP solution x^* in the observed instance; Right: a local expansion minimum that nearly achieves our upper bound on the Hamming error. Rows: `road`, `bikes`, `car`. On these instances, our theoretical result guarantees that the Hamming error of *any* local expansion minimum is at most 17%, 14%, and 8%, respectively. The local expansion minima in the rightmost column have Hamming error of 11%, 8%, and 7% of the nodes, respectively. Our theoretical result implies that these local minima are almost the "worst possible" w.r.t. Hamming error. These "bad" expansion minima were found by initializing the α-expansion algorithm with (a rounded version of) the labeling x output by (10).

where $M(G) \subset K_t$ for all t, and $K_t \subset K_{t-1}$. We start with $K_0 = L(G)$, then use the "cycle constraints" from Sontag & Jaakkola (2008) to go from K_t to K_{t+1}. Violated cycle constraints can be found efficiently by computing shortest paths in an auxiliary graph that depends on the solution x^t to this program. Even if we could efficiently represent the constraint that $x \in M(G)$, this approach would still be a relaxation of the ILP formulation, because the optimal x may not be attained at a vertex of $M(G)$. However, this relaxation is nearly tight on several of our empirical examples. The exact ILP formulation (9) and its relaxation (10) give both claims of Theorem 4. □

There is also a simpler approach to upper-bounding the optimal value of (7) for affine f based on Theorem 1, the original approximation guarantee for α-expansion. That result guarantees that any expansion minimum satisfies $\langle \theta, x \rangle \leq \langle \theta, x^* \rangle + \sum_{uv} \theta_{uv}(x^*(u), x^*(v))$. Therefore, we can upper bound (7) with the ILP:

$$\begin{aligned} \underset{x}{\text{maximize}} \quad & \langle f, x \rangle \\ \text{subject to} \quad & x \text{ a vertex of } M(G) \\ & \langle \theta, x \rangle \leq \langle \theta, x^* \rangle + \sum_{uv} \theta_{uv}(x^*(u), x^*(v)). \end{aligned} \tag{11}$$

Like (9), this is an ILP. We refer to this as the *naive* bound, since it comes directly from the approximation guarantee for α-expansion. In the next section, we compare (10) to (11) on real-world instances, and find that our bound (10) is much tighter. Intuitively, our bound carefully tries to enforce that the optimization variable x is an *optimal point* in some instance, whereas the naive bound may allow for feasible points x that are not optimal in any instance.

6. Numerical results

In this section, we run (10) on several real-world MAP inference instances to evaluate the tightness of bounds derived from our structural condition (Theorem 2). Theorem 2 guarantees that all local expansion minima x for instance θ are contained in $\mathcal{S}(\theta)$, the set of exact solutions to certain perturbations of the input problem θ. If we upper bound the Hamming distance to x^* and the objective gap $\langle \theta, x \rangle / \langle \theta, x^* \rangle$ over $\mathcal{S}(\theta)$, we obtain upper bounds on the Hamming recovery and objective gap that apply to *all* solutions that can possibly be returned by α-expansion. These "problem-dependent worst-case" bounds hold for every possible initial labeling and every possible update order in Algorithm 1.

Broadly, we find that the real-world examples we study are not pathological: global optima to perturbed instances tend to be quite close to global optima of the original instance. Together with Theorem 2, this implies that these instances have no spurious local minima w.r.t. expansion moves.

Table 1. Results of our bound on six real instances.

Instance	Obj. bd. (ours)	Obj. bd. (naive)	Ham. err. bd. (ours)	Ham. bd. (naive)
`tsukuba`	1.213	1.228	0.290	0.821
`venus`	1.199	1.268	0.375	0.703
`plastic`	1.073	1.095	0.373	0.779
`road`	1.031	1.036	0.171 (0.114)	0.256
`bikes`	1.027	1.030	0.146 (0.082)	0.229
`car`	1.019	1.047	0.081 (0.074)	0.225

Table 1. Results on six MAP inference instances from computer vision: 3 stereo vision (top) and 3 object segmentation (bottom). Our bounds on the objective gap and Hamming error are obtained by iteratively running (10). The "naive" bounds are obtained by using (11). Our procedure results in slightly tighter objective gap bounds and much tighter Hamming error bounds. For the object segmentation instances, lower bounds on the Hamming error of local expansion minima are shown in parentheses. That is, there exist local expansion minima with the Hamming error displayed in parentheses. These minima are shown in Figure 2, and were found by running α-expansion initialized with the output of (10). Our Hamming error bound implies that these are almost the "worst possible" expansion minima w.r.t. Hamming error. For example, on the `car` instance, our bound guarantees that *any* local expansion minimum agrees with the MAP solution on at least 91.9% of the vertices, and we have found a local minimum that agrees with the MAP solution on just 92.6% of the vertices.

We study two types of instances: first, a stereo vision problem, where the weights w and costs $\theta_u(i)$ are set "by hand" according to the model from Tappen & Freeman (2003). Given two images taken from slightly offset locations, the goal is to estimate the depth of every pixel in one of the images. This can be done by estimating, for each pixel, the disparity between the two images, since the depth is inversely proportional to the disparity. In the Tappen & Freeman (2003) model, the node costs are set using the sampling-invariant technique from Birchfield & Tomasi (1998), and the weights w_{uv} are set as:

$$w_{uv} = \begin{cases} P \times s & |I(u) - I(v)| < T \\ s & \text{otherwise,} \end{cases}$$

where P, T, and s are the parameters of the model, and $I(u)$ is the intensity of pixel u in one of the input images to the stereo problem. These edge weights charge more for separating pixels with similar intensities, since nearby pixels with similar intensities are likely to be at the same depth. We also study object segmentation instances, where the weights w and costs θ_u are learned from data. In this problem, the goal is to assign a label to each pixel that represents the object to which that pixel belongs. For these instances, we use the models from Alahari et al. (2010). We include the full details of both models in Appendix C.

Table 1 shows the results of running several rounds of (10) on six of these instances. For each instance, we compare against the naive objective bound $\langle \theta, x^* \rangle + \sum_{uv} \theta_{uv}(x^*(u), x^*(v))$ obtained from the original proof of α-expansion's approximation guarantee, and against the naive Hamming bound obtained by solving (11). We used Gurobi (Gurobi Optimization, 2020) to run the iterations of (10) and to solve the ILP (11). We added cycle inequalities using the k-projection graph (Sontag & Jaakkola, 2008), adding several violated inequalities per iteration. We ran between 10 and 20 iterations of (10) for each experiment. Tightening using the cycle inequalities was beneficial in practice. For example, it improved our Hamming error bound on the `tsukuba` instance from 0.38 to 0.29.

Compared to (11), our procedure results in slightly tighter objective bounds and much tighter Hamming bounds on these instances. For example, on the `car` instance, our bound certifies that *all* local minima w.r.t. expansion moves must agree with the MAP solution x^* on at least 91.9% of the nodes. Moreover, there exists an expansion minimum for this instance that agrees on only 92.6% of the vertices, which nearly matches our bound. This "worst-case" expansion minimum is shown in Figure 2.

7. Conclusion

We have shown that graph cuts algorithms, such as α-expansion and FastPD, take advantage of special structure in real-world problem instances with Potts potentials. Our empirical results show that the solutions (the global energy minima) to small perturbations of the input are often very close to the solutions of the original instance. Our theoretical result states that all local minima w.r.t. expansion moves are global minima in such perturbations. Taken together, these two results imply that there are no spurious local minima w.r.t. expansion moves in practice. This gives a new theoretical explanation for the good performance of graph cuts algorithms in the wild. Moreover, our structural result could have practical consequences for learning Markov random fields. To ensure α-expansion performs well on an instance, one could add a regularization term during learning that encourages the solutions to small perturbations $\mathcal{I}(\theta)$ of the instance to be close to the solution of the original.

Acknowledgments

The authors thank Chandler Squires for his helpful feedback on drafts of this paper and an anonymous reviewer for pointing us to Shekhovtsov & Hlavac (2011). This work was supported by NSF AitF awards CCF-1637585 and CCF-1723344.

References

Alahari, K., Kohli, P., and Torr, P. H. Dynamic hybrid algorithms for map inference in discrete mrfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32 (10):1846–1857, 2010.

Angelidakis, H., Makarychev, K., and Makarychev, Y. Algorithms for stable and perturbation-resilient problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 438–451, 2017.

Angelidakis, H., Awasthi, P., Blum, A., Chatziafratis, V., and Dan, C. Bilu-linial stability, certified algorithms and the independent set problem. In *27th Annual European Symposium on Algorithms, ESA 2019*, pp. 7. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2019.

Archer, A., Fakcharoenphol, J., Harrelson, C., Krauthgamer, R., Talwar, K., and Tardos, É. Approximate classification via earthmover metrics. In *Proceedings of the fifteenth annual ACM-SIAM symposium on Discrete algorithms*, pp. 1079–1087. Society for Industrial and Applied Mathematics, 2004.

Birchfield, S. and Tomasi, C. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, 1998.

Boykov, Y., Veksler, O., and Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on pattern analysis and machine intelligence*, 23(11): 1222–1239, 2001.

Dahlhaus, E., Johnson, D. S., Papadimitriou, C. H., Seymour, P. D., and Yannakakis, M. The complexity of multiway cuts. In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pp. 241–251, 1992.

Geman, S. and Geman, D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6):721–741, 1984.

Globerson, A. and Jaakkola, T. S. Fixing max-product: Convergent message passing algorithms for map lp-relaxations. In *Advances in neural information processing systems*, pp. 553–560, 2008.

Gurobi Optimization, L. Gurobi optimizer reference manual, 2020. URL http://www.gurobi.com.

Kappes, J. H., Andres, B., Hamprecht, F. A., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B. X., Kröger, T., Lellmann, J., et al. A comparative study of modern inference techniques for structured discrete energy minimization problems. *International Journal of Computer Vision*, 115(2):155–184, 2015.

Kleinberg, J. and Tardos, E. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002.

Komodakis, N., Tziritas, G., and Paragios, N. Fast, approximately optimal solutions for single and dynamic mrfs. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2007.

Kovtun, I. Partial optimal labeling search for a np-hard subclass of (max,+) problems. In *Joint Pattern Recognition Symposium*, pp. 402–409. Springer, 2003.

Lang, H., Sontag, D., and Vijayaraghavan, A. Optimality of approximate inference algorithms on stable instances. In *International Conference on Artificial Intelligence and Statistics*, pp. 1157–1166, 2018.

Lang, H., Sontag, D., and Vijayaraghavan, A. Block stability for map inference. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 216–225, 2019.

Lang, H., Reddy, A., Sontag, D., and Vijayaraghavan, A. Beyond perturbation stability: Lp recovery guarantees for map inference on noisy stable instances. In *The 24th International Conference on Artificial Intelligence and Statistics*. PMLR, 2021.

Makarychev, K. and Makarychev, Y. Certified algorithms: Worst-case analysis and beyond. In *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., and Westling, P. High-resolution stereo datasets with subpixel-accurate ground truth. In *German conference on pattern recognition*, pp. 31–42. Springer, 2014.

Shekhovtsov, A. Exact and partial energy minimization in computer vision. 2013.

Shekhovtsov, A. and Hlavac, V. On partial optimality by auxiliary submodular problems. *Control Systems and Computers*, (2), 2011.

Shekhovtsov, A., Swoboda, P., and Savchynskyy, B. Maximum persistency via iterative relaxed inference in graphical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1668–1682, 2017.

Shotton, J., Winn, J., Rother, C., and Criminisi, A. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision*, pp. 1–15. Springer, 2006.

Sontag, D. and Jaakkola, T. S. New outer bounds on the marginal polytope. In *Advances in Neural Information Processing Systems*, pp. 1393–1400, 2008.

Sontag, D. A. *Approximate inference in graphical models using LP relaxations*. PhD thesis, Massachusetts Institute of Technology, 2010.

Swoboda, P., Shekhovtsov, A., Kappes, J. H., Schnörr, C., and Savchynskyy, B. Partial optimality by pruning for map-inference with general graphical models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(7), 2016.

Szeliski, R., Zabih, R., Scharstein, D., Veksler, O., Kolmogorov, V., Agarwala, A., Tappen, M., and Rother, C. A comparative study of energy minimization methods for markov random fields with smoothness-based priors. *IEEE transactions on pattern analysis and machine intelligence*, 30(6):1068–1080, 2008.

Tappen and Freeman. Comparison of graph cuts with belief propagation for stereo, using identical mrf parameters. In *Proceedings Ninth IEEE International Conference on Computer Vision*, pp. 900–906 vol.2, 2003.

Wainwright, M. J. and Jordan, M. I. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P. H. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE international conference on computer vision*, pp. 1529–1537, 2015.

Graph cuts always find a global optimum for Potts models (with a catch): supplementary material

A. Proof of Theorem 3

In this section, we give the full proof of Theorem 3, restated here. Theorem 2 is then a straightforward corollary of Theorem 3 (Theorem 3 is essentially the constructive version of Theorem 2).

Theorem. *Consider an input instance θ with Potts pairwise potentials and weights w, and let the labeling x be a local minimum for θ with respect to expansion moves. Define perturbed weights $w^x : E \to \mathbb{R}_+$ as*

$$w^x_{uv} = \begin{cases} w_{uv} & x(u) \neq x(v) \\ 2w_{uv} & x(u) = x(v), \end{cases} \tag{12}$$

and let

$$\theta^x_{uv}(i,j) = w^x_{uv}\mathbb{I}[i \neq j] \tag{13}$$

be the pairwise Potts energies corresponding to the weights w^x. Then x is a global minimum in the instance with objective vector $\theta^x = (\theta_u : u;\ \theta^x_{uv} : uv)$. This is the Potts model instance with the same node costs $\theta_u(i)$ as the original instance, but new pairwise energies $\theta^x_{uv}(i,j)$ defined using the perturbed weights w^x. Additionally, the LP relaxation (2) is tight on this perturbed instance.

Proof. Let x be any labeling of G. We show that there exists an expansion x^α of x such that for some $\epsilon > 0$:

$$\langle \theta, x - x^\alpha \rangle \geq \epsilon \left(\langle \theta^x, x \rangle - \min_{y \in L(G)} \langle \theta^x, y \rangle \right). \tag{14}$$

This implies that as long as $\langle \theta^x, x \rangle - \min_{y \in L(G)} \langle \theta^x, y \rangle$ is positive, there exists an expansion move with strictly better objective than x. The right-hand-side of (14) is always nonnegative, because $x \in L(G)$. Therefore, x can only be a local minimum w.r.t. expansion moves if $\langle \theta^x, x \rangle = \min_{y \in L(G)} \langle \theta^x, y \rangle$. Every labeling of G corresponds to a point in $L(G)$, since $M(G) \subset L(G)$, so if $\langle \theta^x, x \rangle = \min_{y \in L(G)} \langle \theta^x, y \rangle$, x must be an optimal labeling in the instance with objective θ^x. This equality also implies that a vertex of $M(G)$ attains the optimal objective value for (2), which is the definition of "tightness" on an instance. So (14) dgives both claims of the theorem.

Let $y' \in \arg\min_{y \in L(G)} \langle \theta^x, y \rangle$ be an LP solution to the perturbed instance. To show (14), we design a rounding algorithm R that takes y' and x as input and outputs an expansion move x^α of x. We show that R satisfies

$$\mathbb{E}[\langle \theta, x - R(x, y') \rangle] \geq \epsilon(\langle \theta^x, x - y' \rangle), \tag{15}$$

which proves (14) because it implies there exists some x^α in the support of $R(x, y')$ that attains (14).

Algorithm 2 $R(x, y')$

1: Fix $0 < \epsilon < 1/k$.
2: Set $x' = \epsilon y' + (1-\epsilon)x$.
3: Choose $i \in \{1, \dots, k\}$ uniformly at random.
4: Choose $r \in (0, 1/k)$ uniformly at random.
5: Initialize labeling $x^\alpha : V \to [k]$.
6: **for** each $u \in V$ **do**
7: **if** $x'_u(i) > r$ **then**
8: Set $x^\alpha(u) = i$.
9: **else**
10: Set $x^\alpha(u) = x(u)$
11: **end if**
12: **end for**
13: **Return** x^α

Lemma 2 (Rounding guarantees)**.** *The labeling x^α output by Algorithm 2 is an expansion of x, and it satisfies the following guarantees:*

$$\begin{aligned} \mathbb{P}[x^\alpha(u) = i] &= x'_u(i) && \forall\, u \in V, i \in [k] \\ \mathbb{P}[x^\alpha(u) \neq x^\alpha(v)] &\leq 2d(u,v) && \forall\, (u,v) \in E : x(u) = x(v) \\ \mathbb{P}[x^\alpha(u) = x^\alpha(v)] &= (1 - d(u,v)) && \forall\, (u,v) \in E : x(u) \neq x(v), \end{aligned}$$

where $d(u,v) = \frac{1}{2}\sum_i |x'_u(i) - x'_v(i)|$.

Proof of Lemma 2 (rounding guarantees). The output x^α is clearly an i-expansion of x for the i chosen in line 3.

For the guarantees, first, fix $u \in V$ and a label $i \neq x(u)$. We output $x^\alpha(u) = i$ precisely when i is chosen in line 3, and $0 < r < x'_u(i)$, which occurs with probability $\frac{1}{k}\frac{x'_u(i)}{1/k} = x'_u(i)$ (we used here that $x'_u(i) \leq \epsilon < 1/k$ for all $i \neq x(u)$). Now we output $x^\alpha(u) = x(u)$ with probability $1 - \sum_{j \neq x(u)} \mathbb{P}[x^\alpha(u) = j] = 1 - \sum_{j \neq x(u)} x'_u(j) = x'_u(x(u))$, since $\sum_i x'_u(i) = 1$. This proves the first guarantee.

For the second, consider an edge (u,v) not cut by x, so $x(u) = x(v)$. Then (u,v) is cut by x^α when some $i \neq x(u)$ is chosen and r falls between $x'_u(i)$ and $x'_v(i)$. This occurs with probability

$$\frac{1}{k}\sum_{i \neq x(u)} \frac{\max(x'_u(i), x'_v(i)) - \min(x'_u(i), x'_v(i))}{1/k} = \sum_{i \neq x(u)} |x'_u(i) - x'_v(i)| \leq 2d(u,v).$$

Finally, consider an edge (u,v) cut by x, so that $x(u) \neq x(v)$. Here $x^\alpha(u) = x^\alpha(v)$ if some i, r are chosen with $r < \min(x'_u(i), x'_v(i))$. We have $r < \min(x'_u(i), x'_v(i))$ with probability $\frac{\min(x'_u(i), x'_v(i))}{1/k}$. Note that this is still valid if $i = x(u)$ or $i = x(v)$, since only one of those equalities can hold. So we get

$$\mathbb{P}[x^\alpha(u) = x^\alpha(v)] = \frac{1}{k}\sum_i \frac{\min(x'_u(i), x'_v(i))}{1/k} = \frac{1}{2}\left(\sum_i x'_u(i) + x'_v(i) - |x'_u(i) - x'_v(i)|\right) = 1 - d(u,v),$$

where we used again that $\sum_i x'_u(i) = 1$. □

Algorithm 2 is very similar to the rounding algorithm from Lang et al. (2018), essentially just using different constants to give a simplified analysis. The algorithm used in Lang et al. (2018) was itself a simple modification of the ϵ-close rounding from Angelidakis et al. (2017).

With these guarantees in hand, we can now prove (15). Let $x^\alpha = R(x, y')$. Let $E^x = \{(u,v) \in E : x(u) \neq x(v)\}$ be the

set of edges cut by x. Recall that $\theta_{uv}(i,j) = w_{uv}\mathbb{I}[i \neq j]$. Then we have:

$$\mathbb{E}[\langle \theta, x - x^\alpha \rangle] = \sum_u \theta_u(x(u))\mathbb{P}[x^\alpha(u) \neq x(u)] - \sum_u \sum_{i \neq x(u)} \theta_u(i)\mathbb{P}[x^\alpha(u) = i] + \sum_{uv \in E^x} w_{uv}\mathbb{P}[x^\alpha(u) = x^\alpha(v)] - \sum_{uv \in E \setminus E^x} w_{uv}\mathbb{P}[x^\alpha(u) \neq x^\alpha(v)].$$

Applying Lemma 2, we obtain:

$$\begin{aligned}\mathbb{E}[\langle \theta, x - x^\alpha \rangle] &\geq \sum_u \theta_u(x(u))(1 - x'_u(x(u))) - \sum_u \sum_{i \neq x(u)} \theta_u(i)x'_u(i) + \sum_{uv \in E^x} w_{uv}(1 - d(u,v)) - \sum_{uv \in E \setminus E^x} 2w_{uv}d(u,v) \\ &= \sum_u \theta_u(x(u)) + \sum_{uv \in E^x} w^x_{uv} - \sum_u \sum_i \theta_u(i)x'_u(i) - \sum_{uv \in E} w^x_{uv}d(u,v). \end{aligned} \tag{16}$$

Here we are using the formula for w^x_{uv} given by (12): $w^x_{uv} = w_{uv}$ if (u,v) is in E^x, and $2w_{uv}$ otherwise.

Because x is a vertex of $M(G)$, the node variables $x_u(i)$ are either 0 or 1. Then there is only one setting of $x_{uv}(i,j)$ that satisfies the marginalization constraints. So the edge cost paid by x on each edge is proportional to $\frac{1}{2}\sum_{uv} |x_u(i) - x_v(i)|$, since this is 1 if x labels u and v differently and 0 otherwise. Therefore,

$$\sum_{uv} \sum_{i,j} \theta^x_{uv}(i,j)x_{uv}(i,j) = \sum_{uv} \frac{w^x_{uv}}{2} \sum_i |x_u(i) - x_v(i)|.$$

The following proposition says we can also rewrite the edge cost paid by the LP solution y' in this way.

Proposition. *In uniform metric labeling, there is a closed form for the optimal edge cost that only involves the node variables. That is, fix arbitrary node variables $z_u(i)$ and $z_v(j)$. Then the value of*

$$\begin{aligned} \min_{z_{uv}} \quad & \sum_{i,j} \mathbb{I}[i \neq j] z_{uv}(i,j) \\ \textit{subject to} \quad & \sum_j z_{uv}(i,j) = z_u(i) \\ & \sum_i z_{uv}(i,j) = z_v(j) \\ & z_{uv}(i,j) \geq 0 \end{aligned}$$

is equal to $\frac{1}{2}\sum_i |z_u(i) - z_v(i)|$ (Archer et al., 2004; Lang et al., 2018). This fact is used to prove that the local LP relaxation is equivalent to the metric *LP relaxation for uniform metric labeling (Archer et al., 2004; Lang et al., 2018).*

Because y' is an optimal solution to (2) for objective θ^x, y' pays the minimum edge cost consistent with its node variables, since otherwise it cannot be optimal. Then the above proposition implies that:

$$\sum_{uv} \sum_{i,j} \theta^x_{uv}(i,j)y'_{uv}(i,j) = \sum_{uv} \frac{w^x_{uv}}{2} \sum_i |y'_u(i) - y'_v(i)|.$$

Since $x'_{uv}(i,j) = \epsilon y'_{uv}(i,j) + (1-\epsilon)x_{uv}(i,j)$, and $d(u,v)$ is convex,

$$d(u,v) = \frac{1}{2}\sum_i |x'_u(i) - x'_v(i)| \leq \frac{\epsilon}{2}\sum_i |y'_u(i) - y'_v(i)| + \frac{1-\epsilon}{2}\sum_i |x_u(i) - x_v(i)|$$

Using this, the definition of x', and the closed forms for the edge cost of y' and x, we can simplify (16) to:

$$\begin{aligned}\mathbb{E}[\langle\theta^x, x - x^\alpha\rangle] &\geq \langle\theta^x, x\rangle - \Bigg[(1-\epsilon)\sum_u \theta_u(x(u)) + \epsilon\sum_u\sum_i \theta_u(i)y'_u(i) + (1-\epsilon)\sum_{uv}\frac{w^x_{uv}}{2}\sum_i |x_u(i) - x_v(i)| \\ &\qquad + \epsilon\sum_{uv}\frac{w^x_{uv}}{2}\sum_i |y_u(i) - y_v(i)|\Bigg] \\ &= \langle\theta^x, x\rangle - [(1-\epsilon)\langle\theta^x, x\rangle + \epsilon\langle\theta^x, y'\rangle] \\ &= \epsilon\langle\theta^x, x - y'\rangle,\end{aligned}$$

which is what we wanted to show. This analysis implies that for any expansion minimum x, (i) x is a MAP solution to the instance θ^x and (ii) the local LP relaxation (2) is tight on the instance θ^x. Point (ii) is crucial for the correctness of our algorithm in Section 5. However, in the next section we give a simpler proof of (i) that does not use the local LP relaxation. □

A.1. Combinatorial proof of Theorem 3 part (i)

Here, we give a simpler proof for the first claim of Theorem 3, that a solution x returned by α-expansion is the optimal labeling in the instance with objective θ^x. However, the extra guarantee of Theorem 3, that the local LP relaxation is tight on the instance with objective θ^x, was crucial to the correctness of our algorithm in Section 5.

Theorem. *Consider an input instance θ with Potts pairwise potentials and weights w, and let the labeling x be a local minimum for θ with respect to expansion moves. Define perturbed weights $w^x : E \to \mathbb{R}_+$ as*

$$w^x_{uv} = \begin{cases} w_{uv} & x(u) \neq x(v) \\ 2w_{uv} & x(u) = x(v), \end{cases} \tag{17}$$

and let

$$\theta^x_{uv}(i,j) = w^x_{uv}\mathbb{I}[i \neq j] \tag{18}$$

be the pairwise Potts energies corresponding to the weights w^x. Then x is a global minimum in the instance with objective vector $\theta^x = (\theta_u : u;\ \theta^x_{uv} : uv)$. This is the Potts model instance with the same node costs $\theta_u(i)$ as the original instance, but new pairwise energies $\theta^x_{uv}(i,j)$ defined using the perturbed weights w^x.

Proof. We'll show that if some assignment y obtains $\langle\theta^x, y\rangle < \langle\theta^x, x\rangle$, there exists an expansion move x^α of x with $\langle\theta, x^\alpha\rangle < \langle\theta, x\rangle$. Consequently, when x is optimal with respect to expansion moves, it is also the global optimal assignment in the instance with objective θ^x. Assume such a y exists and define $V^\alpha = \{u \in V | y(u) = \alpha\}$. This is the set of points labeled α by y. The sets $(V^1, \ldots, V^k)$ form a partition of V. For each $\alpha \in [k]$, define the expansion x^α of x towards y as:

$$x^\alpha(u) = \begin{cases} \alpha & u \in V^\alpha \\ x(u) & \text{otherwise.} \end{cases}$$

We will show:

$$\sum_\alpha (\langle\theta, x\rangle - \langle\theta, x^\alpha\rangle) \geq \langle\theta^x, x\rangle - \langle\theta^x, y\rangle \tag{19}$$

This immediately gives the result: if $\langle\theta^x, y\rangle < \langle\theta^x, x\rangle$, then at least one term in the sum on the left-hand-side must be positive, and this corresponds to an expansion x^α of x with better objective in the original instance.

Consider a single term $\langle\theta, x\rangle - \langle\theta, x^\alpha\rangle$ on the left-hand-side of (19). The difference in node cost terms is precisely $\sum_{u\in V^\alpha}\theta_u(x(u)) - \theta_u(x^\alpha(u))$, since on all $v \in V \setminus V^\alpha$, $x^\alpha(v) = x(v)$. This is equal to $\sum_{u\in V^\alpha}\theta_u(x(u)) - \theta_u(y(u))$, so the sum over α gives the difference in node cost between x and y:

$$\sum_\alpha\sum_{u\in V^\alpha}\theta_u(x(u)) - \theta_u(x^\alpha(u)) = \sum_{u\in V}\theta_u(x(u)) - \theta_u(y(u)). \tag{20}$$

For any assignment z, let $E_z \subset E$ be the set of edges (u, v) separated by z. Then we can write the difference in edge costs between x and x^α, with the original weights w_{uv}, as

$$\sum_{uv \in E_x \setminus E_{x^\alpha}} w_{uv} - \sum_{uv \in E_{x^\alpha} \setminus E_x} w_{uv},$$

and the edge cost difference between x and y with weights w^x as:

$$\sum_{uv \in E_x \setminus E_y} w_{uv} - \sum_{uv \in E_y \setminus E_x} 2w_{uv},$$

where we used the definition of w^x_{uv}. Then what remains is to show:

$$\sum_\alpha \left(\sum_{uv \in E_x \setminus E_{x^\alpha}} w_{uv} - \sum_{uv \in E_{x^\alpha} \setminus E_x} w_{uv} \right) \geq \sum_{uv \in E_x \setminus E_y} w_{uv} - \sum_{uv \in E_y \setminus E_x} 2w_{uv}.$$

Define B^α to be the set of edges with exactly one endpoint in V^α i.e., $B^\alpha = \{(u, v) \in E : |\{u, v\} \cap V^\alpha| = 1\}$. For all $(u, v) \in B^\alpha$, $y(u) \neq y(v)$, and either $y(u) = \alpha$ or $y(v) = \alpha$.

Let $(u, v) \in E_x \setminus E_y$. Because $y(u) = y(v)$, the edge (u, v) appears in *exactly one* of the $E_x \setminus E_{x^\alpha}$. That is, $y(u) = y(v) = \alpha$, so x^α does not cut (u, v), and x^β cuts (u, v) for all $\beta \neq \alpha$. This implies

$$\sum_\alpha \sum_{uv \in E_x \setminus E_{x^\alpha}} w_{uv} \geq \sum_{uv \in E_x \setminus E_y} w_{uv} \tag{21}$$

If x^α separates an edge (u, v) that is not separated by x, exactly one endpoint of (u, v) is in V^α, since otherwise both endpoints would have been assigned label α. Thus $E_{x^\alpha} \setminus E_x \subset B_\alpha \setminus E_x$. This implies

$$\sum_\alpha \sum_{uv \in E_{x^\alpha} \setminus E_x} w_{uv} = \sum_\alpha \sum_{uv \in B^\alpha \setminus E_x} w_{uv} = 2 \sum_{uv \in E_y \setminus E_x} w_{uv}, \tag{22}$$

where the last equality is because each edge in E_y appears in two B^α. Combining (21) and (22), we obtain:

$$\sum_\alpha \left(\sum_{uv \in E_x \setminus E_{x^\alpha}} w_{uv} - \sum_{uv \in E_{x^\alpha} \setminus E_x} w_{uv} \right) \geq \sum_{uv \in E_x \setminus E_y} w_{uv} - \sum_{uv \in E_y \setminus E_x} 2w_{uv}, \tag{23}$$

which is what we wanted. Combining (20) and (23), we obtain (19). □

A.2. Proof of Lemma 1

Proof of lemma 1. Recall that $\mathcal{S}(\theta)$ is defined as the set of x for which there exists $\theta' \in \mathcal{I}(\theta)$ such that x is a MAP solution to the instance θ'. We want to show that $\mathcal{S}(\theta)$ can also be written as:

$$\mathcal{S}(\theta) = \{x : x \text{ a MAP solution to the instance } \theta^x \text{ defined by (4) and (5)}\}$$

To do show, we simply show that if x is a MAP solution for some $\theta' \in \mathcal{I}(\theta)$, then x is also the MAP solution to the instance θ^x. This is effectively because θ^x is the "best possible" perturbation for x that is contained in $\mathcal{I}(\theta)$. Fix $\theta' \in \mathcal{I}(\theta)$ for which x is a MAP solution. Then for all labelings $y \neq x$, $\langle \theta', y \rangle \geq \langle \theta', x \rangle$. In particular,

$$\sum_u \theta'_u(y(u)) + \sum_{uv} \theta'_{uv}(y(u), y(v)) \geq \sum_u \theta'_u(x(u)) + \sum_{uv} \theta'_{uv}(x(u), x(v)).$$

Because we assume throughout that $\theta_{uv}(i, j) = w_{uv}\mathbb{I}[i \neq j]$ (i.e., that the input instance is a Potts model), the definition of $\mathcal{I}(\theta)$ (equation 3) implies that every instance in $\mathcal{I}(\theta)$ is a Potts model. So let w' be the weights of the instance θ'. Additionally, recall that the definition of $\mathcal{I}(\theta)$ implies that $\theta'_u(i) = \theta_u(i)$ for all (u, i). Then the inequality above becomes:

$$\sum_u \theta_u(y(u)) - \sum_u \theta_u(x(u)) + \sum_{\substack{uv : y(u) \neq y(v) \\ x(u) = x(v)}} w'_{uv} - \sum_{\substack{uv : x(u) \neq x(v) \\ y(u) = y(v)}} w'_{uv} \geq 0$$

The definition of $\mathcal{I}(\theta)$ requires that for all (u, v), $w_{uv} \leq w'_{uv} \leq 2w_{uv}$. Together with the previous inequality, this implies

$$\sum_u \theta_u(y(u)) - \sum_u \theta_u(x(u)) + \sum_{\substack{uv: y(u)\neq y(v) \\ x(u)=x(v)}} 2w_{uv} - \sum_{\substack{uv: x(u)\neq x(v) \\ y(u)=y(v)}} w_{uv} \geq 0.$$

By definition of the perturbed weights w^x_{uv} (12), we have

$$\sum_u \theta_u(y(u)) - \sum_u \theta_u(x(u)) + \sum_{\substack{uv: y(u)\neq y(v) \\ x(u)=x(v)}} w^x_{uv} - \sum_{\substack{uv: x(u)\neq x(v) \\ y(u)=y(v)}} w^x_{uv} \geq 0,$$

which is equivalent to:

$$\langle \theta^x, y \rangle \geq \langle \theta^x, x \rangle.$$

Because y was arbitrary, this implies x is a MAP solution to the instance θ^x. □

B. Comparing (7) and (11)

In this section, we expound on the relationship between (7) and (11), the bound obtained directly from α-expansion's objective approximation guarantee. In particular, we show that any x that is feasible for (7) is also feasible for (11). While we solve the relaxation (10) of (7) in practice, this gives some intuition for why (10) gives much tighter bounds than (11).

We have two ways of characterizing the set of labelings x that are local optima w.r.t. expansion moves. The first, guaranteed by Boykov et al. (2001), is that all such x satisfy

$$\langle \theta, x \rangle \leq \langle \theta, x^* \rangle + \sum_{uv \in E} w_{uv} \mathbb{I}[x^*(u) \neq x^*(v)], \tag{24}$$

where x^* is a MAP solution. That is, the "extra" objective paid by x is at most the edge cost paid by a MAP solution. The second, guaranteed by Theorem 2, is that x is the MAP solution in the instance with objective θ^x (i.e., $x \in \mathcal{S}(\theta)$). We now show that any labeling x that is a MAP solution in the instance with objective θ^x also satisfies (24), but the converse is not true. This implies that the feasible region of (7) is strictly smaller than that of (11).

Proposition. *Let x be a labeling that is optimal in the instance with objective θ^x, and let x^* be a MAP solution to the original instance, with objective θ. Then:*

$$\langle \theta, x \rangle \leq \langle \theta, x^* \rangle + \sum_{uv \in E} w_{uv} \mathbb{I}[x^*(u) \neq x^*(v)],$$

Proof. Because x is optimal for θ^x, we have $\langle \theta^x, x \rangle \leq \langle \theta^x, x^* \rangle$. Recall from the definitions of w^x_{uv} and $\theta^x_{uv}(i, j)$ ((12) and (13)) that $\langle \theta^x, x \rangle = \langle \theta, x \rangle$. We also have that

$$\begin{aligned}\langle \theta^x, x^* \rangle = \sum_u \theta_u(x^*(u)) + \sum_{uv} w^x_{uv} \mathbb{I}[x^*(u) \neq x^*(v)] &\leq \sum_u \theta_u(x^*(u)) + 2\sum_{uv} w_{uv} \mathbb{I}[x^*(u) \neq x^*(v)] \\ &= \langle \theta, x^* \rangle + \sum_{uv} w_{uv} \mathbb{I}[x^*(u) \neq x^*(v)].\end{aligned}$$

Here we used that $w^x_{uv} \leq 2w_{uv}$ for all $(u, v) \in E$. Therefore, $\langle \theta, x \rangle \leq \langle \theta, x^* \rangle + \sum_{uv \in E} w_{uv} \mathbb{I}[x^*(u) \neq x^*(v)]$. □

Conversely, not all x satisfying (24) are optimal in the instance with objective θ^x. We now construct a simple example.

Example where (7) is much tighter than (11). Let $k = 4$ and consider a graph $G = (V, E)$ with two nodes s and t, and one edge (s, t). Let $w_{st} = 1$. For the node costs, Set $\theta_s(0) = 0$, $\theta_s(1) = \epsilon$, and $\theta_s(2) = \theta_s(3) = \infty$. Set $\theta_t(0) = \theta_t(1) = \infty$, $\theta_t(2) = \epsilon$, $\theta_t(3) = 0$. The MAP solution x^* clearly labels s with label 0 and t with label 3, for an objective of 1. Now consider the solution x that labels s with label 1 and t with label 2, for an objective of $1 + 2\epsilon$. For this x, because x cuts the only edge, $\theta^x = \theta$ (see (12)). Therefore, x is not optimal in the instance with objective θ^x, so it is not feasible for (7). However,

$$1 + 2\epsilon \leq \langle \theta, x^* \rangle + \sum_{uv \in E} w_{uv} \mathbb{I}[x^*(u) \neq x^*(v)] = 2$$

for $\epsilon < 1/2$. Therefore, x is feasible for (11). The Hamming distance between x and x^* is 1.0—x agrees with x^* on 0 nodes—so if we run (11) to bound the Hamming error on this instance, we obtain a bound of 1.0. On the other hand, (7) returns a Hamming distance bound of 0 for this instance, correctly indicating that α-expansion always recovers x^* regardless of the initialization. This is because any optimal x must cut (s, t), since otherwise it has infinite objective, and any x that cuts (s, t) has $\theta^x = \theta$, and x^* is the only optimal labeling for objective θ. Hence x^* is the only feasible point of (7) for this instance.

C. Model details

In this section, we give more details on the models used for our experiments in Section 6. These models are similar to the ones studied in Lang et al. (2019). There are two types of models: object segmentation and stereo vision.

C.1. Object segmentation

We use the object segmentation models from Shotton et al. (2006), which were also studied by Alahari et al. (2010) in the context of graph cut methods. These models are available as part of the OpenGM 2 benchmark (Kappes et al., 2015)[1]. In these models, G is a grid with one vertex per pixel and has edges connecting adjacent pixels. The node costs $\theta_u(i)$ are set based on a learned function of shape, color, and location features. Similarly, the edge weights are set using *contrast-sensitive* features:

$$w_{uv} = \eta_1 \exp\left(-\frac{||I(u) - I(v)||_2^2}{2\sum_{p,q} ||I(p) - I(q)||_2^2}\right) + \eta_2,$$

where $\eta = (\eta_1, \eta_2)$, $\eta \geq 0$ are learned parameters, and $I(u)$ is the vector of RGB values for pixel u in the image. Shotton et al. (2006) learn the parameters for the node and edge potentials using a shared boosting method. Each object segmentation instance has 68,160 nodes (the images are `213 x 320`) and either $k = 5$ or $k = 8$ labels. As noted in Kappes et al. (2015), the MRFs used in practice increasingly use potential functions that are learned from data, rather than set by hand. In our experimental results, we found that both the objective gap and Hamming distance bounds were very good for these instances (in comparison to the stereo examples, which have "hand-set" potentials). Do the learning dynamics automatically encourage solutions to perturbed instances to be close to solutions of the original instance? Understanding the relationship between learning and this "stability" property is an interesting direction for future work.

C.2. Stereo Vision

In these models, the weights w_{uv} and costs $\theta_u(i)$ are set "by hand" according to the model from Tappen & Freeman (2003). Given two images taken from slightly offset locations, the goal is to estimate the depth of every pixel in one of the images. This can be done by estimating, for each pixel, the disparity between the two images, since the depth is inversely proportional to the disparity. In the Tappen & Freeman (2003) model, the node costs are set using the sampling-invariant technique from Birchfield & Tomasi (1998). These costs are similar to

$$\theta_u(i) = (I_L(u) - I_R(u - i))^2,$$

where I_L and I_R are the pixel intensities in the left and right images. If node u corresponds to pixel location (h, w), we use $u - i$ to represent the pixel in location $(h, w - i)$. So this cost function measures how likely it is that the pixel at location u in the left image corresponds to the pixel at location $u - i$ in the right image. The Birchfield-Tomasi matching costs are set using a correction to this expression that accounts for image sampling. In the Tappen and Freeman model, the weights w_{uv} are set as:

$$w_{uv} = \begin{cases} P \times s & |I(u) - I(v)| < T \\ s & \text{otherwise,} \end{cases}$$

where P, T, and s are the parameters of the model, and $I(u)$ is the intensity of pixel u in one of the input images to the stereo problem (in our experiments, we use I_L, the left image). These edge weights charge more for separating pixels with similar intensities, since nearby pixels with similar intensities are likely to correspond to the same object, and therefore be at the same depth. In our experiments, we follow Tappen & Freeman (2003) and set $s = 50$, $P = 2$, $T = 4$. In our experiments,

[1] All OpenGM 2 benchmark models are accessible at `http://hciweb2.iwr.uni-heidelberg.de/opengm/index.php?l0=benchmark`

we used images from the Middlebury stereo dataset (see, e.g., Scharstein et al., 2014). We used a downscaled version of the `tsukuba` image that was `120 x 150`, and had $k = 7$. Our `venus` model used the full-size image, which is `383 x 434`, and has $k = 5$. For `plastic`, we again used a downscaled, `111 x 127` image with $k = 5$. The large size of the `venus` image, in particular, shows that our verification algorithm is tractable to run even on fairly large problems.