Relaxation-Based Preprocessing Techniques for Markov Random Field Inference

Chen Wang
Cornell University
chenwang@cs.cornell.edu

Ramin Zabih
Cornell University
rdz@cs.cornell.edu

Abstract

Markov Random Fields (MRFs) are a widely used graphical model, but the inference problem is NP-hard. For first-order MRFs with binary labels, Dead End Elimination (DEE) [7] and QPBO [2, 14] can find the optimal labeling for some variables; the much harder case of larger label sets has been addressed by Kovtun [16, 17] and related methods [12, 23, 24, 25], which impose substantial computational overhead. We describe an efficient algorithm to correctly label a subset of the variables for arbitrary MRFs, with particularly good performance on binary MRFs. We propose a sufficient condition to check if a partial labeling is optimal, which is a generalization of DEE's purely local test. We give a hierarchy of relaxations that provide larger optimal partial labelings at the cost of additional computation. Empirical studies were conducted on several benchmarks, using expansion moves [4] for inference. Our algorithm runs in a few seconds, and improves the speed of MRF inference with expansion moves by a factor of 1.5 to 12.

1. Introduction

We address the inference problem for pairwise Markov Random Fields (MRFs) defined over n variables $x = (x_1, \ldots, x_n)$, where each x_i is labeled from a discrete label set $\mathcal{L}_i$. There is an energy function $E(x)$ that we wish to minimize given a set of parameters θ; θ characterizes the *unary costs* $\theta_i : \mathcal{L}_i \mapsto \mathbb{R}^+$ and the *pairwise costs* $\theta_{ij} : \mathcal{L}_i \times \mathcal{L}_j \mapsto \mathbb{R}^+$. The energy function is

$$E(x) = \sum_{i \in V} \theta_i(x_i) + \sum_{(i,j) \in E} \theta_{ij}(x_i, x_j) \qquad (1)$$

where $G = (V, E)$ is the graph representation of the MRF.

The MRF inference problem is to find $x^* = \arg\min_x E(x)$, which is equivalent to finding the MAP estimate. This is widely used in applications such as image segmentation, stereo, etc [11, 26]. Unfortunately the MRF inference problem is NP-hard even when $|\mathcal{L}| = 2$ (i.e. binary labels) [15].

Method	Handles $\lvert\mathcal{L}\rvert > 2$?	Bottleneck
Our method	Yes	None
DEE [7]	Sometimes	None
QPBO [2]	No	max-flow
Kovtun [16, 17]	Yes	max-flow
MQPBO [12]	Yes	max-flow
Swoboda [25]	Yes	LP, MRF inference
Shekhovtsov [23]	Yes	LP
Shekhovtsov [24]	Yes	LP, MRF inference

Table 1. Partial optimality algorithms. The bottleneck column indicates any subroutine with complexity significantly greater than linear time.

1.1. Optimal partial labelings

A popular approach to the inference problem is to try to find the optimal labeling for a subset of the variables [10, 12, 13, 22, 23, 24, 25, 28]. A partial labeling that holds in every global minimizer is said to be *persistent* [2]. An optimal labeling for a subset of the variables can be used to reduce the difficulty of the inference problem, or can be the basis for a variety of heuristics such as QPBO-I [21].

Techniques like QPBO [2, 14] find an optimal partial labeling by seeking an even stronger condition, namely a partial labeling which will not increase the energy if it is applied to any complete labeling. QPBO in particular is widely used in computer vision since it often finds the correct label for the vast majority of the variables.

Algorithms for finding optimal partial labelings are summarized in Figure 1 and discussed in Section 2. Except for Dead End Elimination (DEE) [7] they all impose significant computational costs, using max flow, linear programming or both. Our technique generalizes DEE, and has significantly better performance experimentally.

1.2. Contributions and outline

In this paper, we address the problem of finding an optimal partial labeling as efficiently as possible. We propose a condition to guarantee that a labeling is part of every global minimizer, and represent this condition as a system of linear inequalities. We establish a hierarchy of relaxations of the original system and derive a family of tractable sufficient conditions. We then propose an efficient algorithm to find a set of variables satisfying the corresponding conditions. Using the loosest condition in the relaxation hierarchy, we can find a globally optimal subset of variables by running a small number of iterations, each of which takes $O(|V| + |E|)$ time; this is very efficient compared to the $O(|V|^2|E|)$ running time of max-flow.[1] The hierarchy of relaxations allow us to trade off between the running time and the number of variables optimally labeled. Our methods perform particularly well on binary MRFs. We conduct experiments on a variety of vision applications and obtain promising experimental results. In particular, when integrated into expansion moves [4] as the MRF inference algorithm our technique labels a large number of variables with minimal overhead, thus producing a substantial speedup.

We review the literature in Section 2. The proposed algorithm is given in Section 3. Experimental results are illustrated in Section 4. Additional experimental results and all the detailed proofs are provided in the supplementary material.

2. Related Work

Empirical studies of MRF inference approaches can be found in [11, 26]. Since the problem is NP-hard in general these techniques find approximate solutions. Rapidly determining the optimal labels for a subset of the variables would obviously be of great utility. Existing methods are summarized in Table 1.

Optimal partial labelings are commonly used in conjunction with graph cuts, a technique that achieves strong performance on both binary and multilabel MRF inference [26]. Graph cuts handle binary MRFs by reduction to min-cut, which is then solved via max-flow (see [2, 8] for reviews). The most widely used graph cut methods for multi-label MRF inference are move-making techniques, which generate a new proposal at each iteration and reduce the multi-label problem into a series of binary subproblems (should each pixel stick with the old label or switch to the new label in the proposal) and then solved by max-flow/min-cut. Popular algorithms in this family include expansion moves [4] and their generalization to fusion moves [19].

QPBO is a generalization of the binary graph cut reduction that uses max-flow to find an optimal partial labeling [2, 14, 21]. The graph where max-flow is run is twice the size of the original MRF, with $2|V|$ nodes and $2|E| + 2|V|$ edges. When the energy function is submodular[2], then the partial labeling is complete (i.e., it labels every pixel and is a global minimizer). However, the computational expense of running max-flow is non-trivial, and our goal is to find substantially faster techniques. Note that the only methods with significantly better running time than max-flow are DEE and our technique.

Kovtun [16, 17] proposed an approach to handle multi-label MRFs by constructing a series of binary auxiliary problems and solve each of them via graph cuts. MQPBO [12] and generalized roof duality [28] generalized QPBO to multi-label cases. The computational costs for these methods are all at least as large as max-flow.

Recently, Swoboda et. al. [25] use standard MRF inference algorithms to iteratively update the persistent variable set. Shekhovtsov [23] formalized the problem to maximize the number of optimally labeled variables as an LP. They also proposed to combine these two approaches together which can take advantage of both of them [24]. The number of variables labeled by these approaches are significantly more than Kovtun's approach and MQPBO. However, the running time of these approaches is significantly longer, since these approaches involve solving complex programming (either via standard MRF inference solver or LP solver) iteratively.

Dead End Elimination (DEE) [7] is the only existing method with cheaper computational costs than max-flow. It checks a local sufficient condition which only involves a single vertex and its adjacent edges. We will show in Section 3.1 that this condition is a special case of the loosest condition of our approach, hence our approach will always label at least as many variables as DEE, with the same running time complexity. Experimental results confirm our approach can label substantially more variables than DEE.

An intuitive comparison of our technique with DEE is provided in Figure 1. The most striking difference is that DEE considers one individual variable at a time and potentially rules out one of its labels; for binary problems, this allows it to determine the globally optimal label for that variable. Our method can determine whether a particular label is optimal for a set of variables, and is not restricted to binary labels. Note that when an entire set of variables fails our sufficient condition for optimality, we shrink the set, as shown in the middle figure. The crucial step in our method is the second from the last one, highlighted with a gray background, where a group of 6 variables is given their optimal labels all at once.

[1] To be precise, $O(|V| + |E|)$ is the running time of our inner subroutine, which finds a globally optimal subset of variables that increases on each iteration. In practice this needs to be run a very limited number of times, as shown in the experimental results that run 5 iterations.

[2] For every pairwise cost, we have $\theta_{ij}(0,0) + \theta_{ij}(1,1) \leq \theta_{ij}(0,1) + \theta_{ij}(1,0)$.

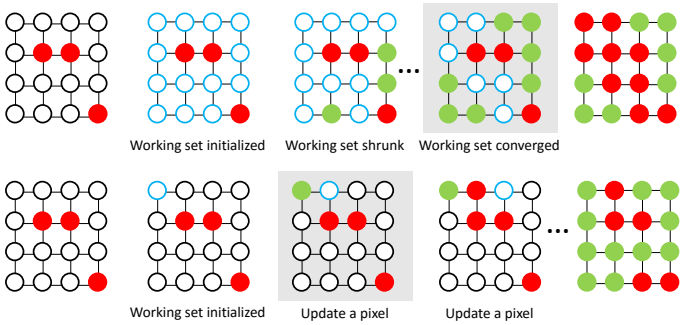

Figure 1. Our method (top row) versus DEE (bottom row), running on a binary-valued MRF with 16 variables. Optimally labeled variables are shown in red, while the working set is light blue. Green variables fail the sufficient test to be optimally labeled. The key step of each algorithm is highlighted with a grey background.

Methods that optimally label a subset of the variables can obviously be used to accelerate MRF inference algorithms such as expansion moves. For example, Radhakrishnan and Su [20] used DEE while Alahari et. al. [1] applied Kovtun's approach.

3. Hierarchical Relaxation of Persistency

Notation and preliminaries Recall that we have n variables $x_1, \ldots, x_n$ in (1) and each variable x_i takes its value from a discrete label set $\mathcal{L}_i$. We will use x to represent the vector $(x_1, \ldots, x_n)$ and x_S to represent a subvector of x with indices in S, where $S \subseteq V$. We will refer to x and x_S as a *full labeling* and *partial labeling* (w.r.t. S) respectively. Define $\mathcal{L}_S = \Pi_{i \in S} \mathcal{L}_i$ as the label space of x_S, which contains the special case where $\mathcal{L}_V$ is the label space of x. Given two partial labelings x_A and x_B where A and B are disjoint, the partial labeling $x_A \oplus x_B$ defined on $A \cup B$ as the composition of partial labelings x_A and x_B.[3] We will view overwriting a full labeling y with a partial labeling x_S as a special case of label composition, i.e., $x_S \oplus y_{V \setminus S}$.

The following definitions come from the literature on pseudo-boolean optimization (see, e.g. [2]).

Definition 1. *A partial labeling x_S is persistent if*

$$x_S = x^*_S, \quad \forall x^* \in \arg\min_x E(x). \tag{2}$$

Definition 2. *A partial labeling x_S is an autarky if*

$$\begin{aligned} &E(x_S \oplus z_{V \setminus S}) < E(y_S \oplus z_{V \setminus S}), \\ &\forall z_{V \setminus S} \in \mathcal{L}_{V \setminus S} \text{ and } \forall y_S \in \mathcal{L}_S \text{ such that } y_S \neq x_S. \end{aligned} \tag{3}$$

[3] Formally, $y = x_A \oplus x_B$ for disjoint A, B means that $y_i = (x_A)_i$ when $i \in A$ and $y_i = (x_B)_i$ when $i \in B$.

Persistency is the property that we seek, since it means that a partial labeling assigns the optimal value to its variables. Autarky is a stronger condition, which states that overwriting an arbitrary labeling with this partial labeling will reduce the energy. Autarky clearly implies persistency, and is tractable since it can be computed without knowing the global minimizer(s) x^*. We will therefore use autarky as a sufficient condition for persistency. Also note that for a set S, if there exists a persistent partial labeling x_S, it must be unique. So we may also say the set S is persistent without explicitly referring to its labeling.

We study two questions: 1) given the energy function $E(x)$ and partial labeling x_S, determine if x_S is persistent; 2) given the energy function $E(x)$, find a persistent partial labeling x_S as large as possible, where the size of partial labeling is defined by $|S|$. We will refer to these as the *persistency decision problem* and *persistency construction problem* respectively.

3.1. Persistency decision problem

Since there are exponentially many inequalities in (3), it's computationally intractable to examine them one by one. Moreover, the persistency decision problem is NP-complete [2] so that we cannot expect to check it exactly. To handle it, we will establish a hierarchical relaxation of the autarky inequality system (3), which gives us a family of sufficient conditions to check persistency.

Define $\Delta E(y_S \leftarrow x_S \mid z_{V \setminus S}) := E(y_S \oplus z_{V \setminus S}) - E(x_S \oplus z_{V \setminus S})$ to be the energy change when we substitute y_S for x_S. Let the index set $A := \{i \in S \mid y_i \neq x_i\}$ be the indices of variables we actually changed from x_S to y_S. Then we know that $\Delta E(y_A \leftarrow x_A \mid x_{S \setminus A}, z_{V \setminus S}) = \Delta E(y_S \leftarrow x_S \mid z_{V \setminus S})$ and the autarky property in (3) is equivalent to $\min_{z_{V \setminus S} \in \mathcal{L}_{V \setminus S}} \min_{y_i \neq x_i, i \in A} \Delta E(y_A \leftarrow x_A \mid x_{S \setminus A}, z_{V \setminus S}) > 0$ for all $A \subseteq S, A \neq \emptyset$. We expand the definition of energy using (1) and cancel out the unchanged terms. Then a further relaxation could be taken by pushing the min operators into the summation:

$$\begin{aligned} &\min_{z_{V \setminus S} \in \mathcal{L}_{V \setminus S}} \min_{y_i \neq x_i, i \in A} \Delta E(y_A \leftarrow x_A \mid x_{S \setminus A}, z_{V \setminus S}) \\ \geq &\sum_{i \in A} \min_{y_i \neq x_i} \big(\theta_i(y_i) - \theta_i(x_i)\big) \\ + &\sum_{ij \in (A, S \setminus A)} \min_{y_i \neq x_i} \big(\theta_{ij}(y_i, x_j) - \theta_{ij}(x_i, x_j)\big) \\ + &\sum_{ij \in (A, V \setminus S)} \min_{y_i \neq x_i, z_j \in \mathcal{L}_j} \big(\theta_{ij}(y_i, z_j) - \theta_{ij}(x_i, z_j)\big) \\ + &\sum_{ij \in (A, A)} \min_{y_i \neq x_i, y_j \neq x_j} \big(\theta_{ij}(y_i, y_j) - \theta_{ij}(x_i, x_j).\big) \end{aligned} \tag{4}$$

Here $ij \in (A, B)$ is short for $\{(i, j) \in E \mid i \in A, j \in B\}$.

In order to simplify the notation in (4), we define a short-

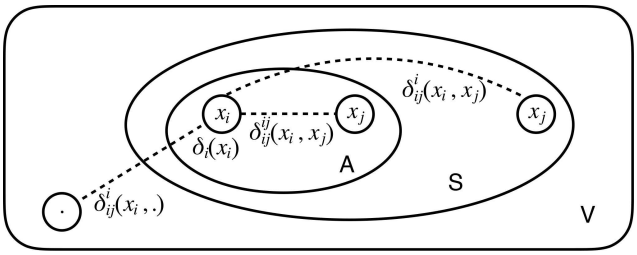

Figure 2. The terms in our relaxation (4). To prove the persistency of the partial labeling x_S, we show that every subset $A \subseteq S$ meets certain conditions. All the variables in A change from their values in x_S, the variables in $S \backslash A$ stay the same, and the ones in $V \backslash S$ are arbitrary.

hand for each term inside the summation as follows:

$$\begin{aligned} \delta_i(x_i) &:= \min_{y_i \neq x_i} \theta_i(y_i) - \theta_i(x_i) \\ \delta^i_{ij}(x_i, x_j) &:= \min_{y_i \neq x_i} (\theta_{ij}(y_i, x_j) - \theta_{ij}(x_i, x_j)) \\ \delta^i_{ij}(x_i, .) &:= \min_{y_i \neq x_i, z_j \in \mathcal{L}_j} (\theta_{ij}(y_i, z_j) - \theta_{ij}(x_i, z_j)) \\ \delta^{ij}_{ij}(x_i, x_j) &:= \min_{y_i \neq x_i, y_j \neq x_j} (\theta_{ij}(y_i, y_j) - \theta_{ij}(x_i, x_j)) \end{aligned} \tag{5}$$

The relationship between A, S and V and the notation we introduced in (5) is illustrated in Figure 2. Note that all the unary terms and pairwise terms inside $V \backslash A$ are unchanged. So $\delta_i(x_i)$, which is the minimum energy change (possibly negative) in unary cost θ_i inside A, and $\delta^{ij}_{ij}(x_i, x_j), \delta^i_{ij}(x_i, x_j), \delta^i_{ij}(x_i, .)$, which are the minimum energy change in pairwise cost θ_{ij} inside A and crossing the boundary of A respectively, are the only terms we need to focus on.[4]

Now we can summarize our analysis above to obtain our first relaxation of the sufficient conditions.

Theorem 3. *The partial labeling x_S is persistent if:*

$$\begin{aligned} &\sum_{i \in A} \delta_i(x_i) + \sum_{ij \in (A, S \backslash A)} \delta^i_{ij}(x_i, x_j) + \sum_{ij \in (A, V \backslash S)} \delta^i_{ij}(x_i, .) + \\ &\sum_{ij \in (A, A)} \delta^{ij}_{ij}(x_i, x_j) > 0, \qquad \forall A \subseteq S, A \neq \emptyset. \end{aligned} \tag{6}$$

Note that there are still $O(2^{|S|})$ inequalities in (6). To further relax it, we will focus on testing the persistency of an independent local minimum (ILM) labeling defined as follows.

Definition 4 (Independent local minimum labeling)**.** *A partial labeling x_S is called an independent local minimum if it minimizes each pairwise term θ_{ij} such that $i, j \in S$.*

[4] In general, our notation $\delta^B_A(x_A)$ represents the smallest energy change for one term θ_A with initial labeling x_A, minimizing over all different values of all the variables in B. Sometimes we need to consider variables not in B to have arbitrary values, which we represent with a period. For the unary term, the superscript in $\delta^i_i(x_i)$ is redundant, hence omitted.

Recall that, as shown in Figure 1, our method considers an input labeling and then shrinks it, while DEE considers a single pixel at a time. Our input must be an ILM labeling, and the larger the better. Fortunately we will show in Section 3.3 that we can easily construct large ILM labelings for the vast majority of MRFs used in vision, and can also guarantee an ILM labeling of at least size 2 for an arbitrary MRF. So even in the worst case we retain our advantage over DEE.

The direct benefit from the ILM labeling assumption is we have $\delta^i_{ij}(x_i, x_j) \geq 0$ and $\delta^{ij}_{ij}(x_i, x_j) \geq 0$ by definition. Now we can obtain a hierarchy of relaxations to (6) as follows.

Theorem 5 (k-condition for S)**.** *The ILM partial labeling x_S containing at least k variables is persistent if $\forall B \subseteq S, |B| = k \geq 1$, the following inequalities hold:*

$$\begin{aligned} &\sum_{i \in C} \delta_i(x_i) + \sum_{ij \in (C, B \backslash C)} \delta^i_{ij}(x_i, x_j) + \sum_{ij \in (C, V \backslash S)} \delta^i_{ij}(x_i, .) > 0, \\ &\qquad \forall C \subseteq B, C \neq \emptyset \end{aligned} \tag{7}$$

Proof. Here is a sketch to show (7) is a sufficient condition to derive (6). More details can be found in the supplementary material.

For any $A \subseteq S, 1 \leq |A| \leq k$, we can pick arbitrary $B \supseteq A, |B| = k$ and let $C := A$, then we have $\sum_{i \in A} \delta_i(x_i) + \sum_{ij \in (A, B \backslash A)} \delta^i_{ij}(x_i, x_j) + \sum_{ij \in (A, V \backslash S)} \delta^i_{ij}(x_i, .) > 0$ from (7). Combining with the fact that 1) $(A, B \backslash A) \subseteq (A, S \backslash A)$ and $\delta^i_{ij}(x_i, x_j) \geq 0$, 2) $\delta^{ij}_{ij}(x_i, x_j) \geq 0$, we get the desired inequality in (6).

For any $A \subseteq S, |A| > k$, we know $\sum_{i \in B} \delta_i(x_i) + \sum_{ij \in (B, V \backslash B)} \delta^i_{ij}(x_i, .) > 0$ from (7) (by choosing $C := B$). Now we pick all $B \subseteq A, |B| = k$ and sum up the previous inequality, we will have $\sum_{i \in A} \delta_i(x_i) + \sum_{ij \in (A, V \backslash A)} \delta^i_{ij}(x_i, .) > 0$. Combining with the fact that $\delta^{ij}_{ij}(x_i, x_j) \geq 0$ and $\delta^i_{ij}(x_i, x_j) \geq 0$, we have the desired inequality in (6). □

Our k-condition in (7) is a hierarchy of relaxations of (6) for different k's. There are $\binom{|S|}{k}(2^k - 1)$ inequalities in the k-condition for S, hence it's computationally efficient to check when k is small. Meanwhile, the larger k is, the more tightly (7) approximates (6). We thus obtain a tradeoff between the complexity and accuracy of the relaxation by varying k.

Now we will claim the sufficient condition to check persistency in DEE is a special case of our 1-condition (i.e., $k = 1$). Note that our 1-condition says the constant partial labeling x_S is persistent if

$$\delta_i(x_i) + \sum_{j \in S, (i,j) \in E} \delta^i_{ij}(x_i, x_j) + \sum_{j \notin S, (i,j) \in E} \delta^i_{ij}(x_i, .) > 0, \quad \forall i \in S, \tag{8}$$

while the Goldstein condition used in DEE says[5] that variable x_i is persistent if

$$\delta_i(x_i) + \sum_{(i,j)\in E} \delta_{ij}^i(x_i, .) > 0. \tag{9}$$

This is a special case of our 1-condition when $S = \{i\}$. Thus, our 1-condition generalized DEE's Goldstein condition from a single variable to an ILM partial labeling.

Approximating the k-condition Recall our k-condition consists of $\binom{|S|}{k}(2^k - 1)$ inequalities, so checking them one by one will become computationally intractable soon with the growth of k. Therefore, we propose an approximate way to check the k-condition that is very efficient in practice, based on the following lemma.

Lemma 6. *The ILM partial labeling x_S is persistent if we can partition S into disjoint subsets $S = \bigcup_t S_t$ and each S_t satisfies the corresponding $|S_t|$-condition.*

Proof. We will also provide a sketch to show this is a sufficient condition of (6) here and defer the details to the supplementary material.

For any non-empty $A \subseteq S$, we can define $A_t := A \cap S_t$. Then we know that $A = \bigcup_t A_t$ and all A_t are disjoint. Our goal is to show $\sum_{i\in A} \delta_i(x_i) + \sum_{ij\in(A,S\setminus A)} \delta_{ij}^i(x_i, x_j) + \sum_{ij\in(A,A)} \delta_{ij}^{ij}(x_i, x_j) + \sum_{ij\in(A,V\setminus S)} \delta_{ij}^i(x_i, .)$ is positive. By dropping non-negative terms and rearranging things, we can prove $\sum_{i\in A} \delta_i(x_i) + \sum_{ij\in(A,S\setminus A)} \delta_{ij}^i(x_i, x_j) + \sum_{ij\in(A,A)} \delta_{ij}^{ij}(x_i, x_j) + \sum_{ij\in(A,V\setminus S)} \delta_{ij}^i(x_i, .) \geq \sum_t (\sum_{i\in A_t} \delta_i(x_i) + \sum_{ij\in(A_t,S_t\setminus A_t)} \delta_{ij}^i(x_i, x_j) + \sum_{ij\in(A_t,V\setminus S_t)} \delta_{ij}^i(x_i, .))$ and we know the RHS is positive due to each S_t satisfying the $|S_t|$-condition. □

In practice, we can approximately test the k-condition for $k > 1$ by doing an incremental breadth-first search style greedy partition of $S = \bigcup_t S_t$ such that S_t are all disjoint, $|S_t| \leq k$, and S_t satisfies the $|S_t|$-condition, which is described in Algorithm 1. The idea is for each single variable i not satisfying the 1-condition, we will search for a subset B containing i that can satisfy the 2-condition, 3-condition, etc. The first found B will be added to our partition. Finally, we add all the left-over single variables satisfying the 1-condition into our partition and claim the remaining variables (i.e., U at the end of Algorithm 1) cannot be proved to be persistent. Note this approximation is still a sound condition guaranteed by Lemma 6, i.e., when $U = \emptyset$ at the end, we know that x_S is persistent.

[5]The original Goldstein condition is to claim one label cannot be persistent, which is equivalent to say its opposite is persistent for the binary case. For the multi-label case, we need to check $|\mathcal{L}_i| - 1$ labels cannot be persistent, so that the remaining one is persistent.

Input: ILM partial labeling x_S
$U \leftarrow S$; $t \leftarrow 0$;
for $i \in U$ *s.t.* $\{i\}$ *fails* 1*-condition test* **do**
  for $k' \leftarrow 2$ ***to*** k **do**
    Search for $B \subseteq U$ s.t. $i \in B$, $|B| = k'$, B satisfies k'-condition;
    if *find such a* B **then**
      $t \leftarrow t + 1$; $S_t \leftarrow B$; $U \leftarrow U \setminus B$;
      break;
    end
  end
end
for $i \in U$ *s.t.* $\{i\}$ *satisfies* 1*-condition* **do**
  $t \leftarrow t + 1$; $S_t \leftarrow \{i\}$; $U \leftarrow U \setminus \{i\}$;
end
if $U = \emptyset$ **then**
  return x_S is persistent;
else
  return U as the cause that x_S fails the test;
end
Algorithm 1: Approximate k-condition Test

Input: Energy function $E(x)$
$W \leftarrow \emptyset$; $x_W \leftarrow \emptyset$;
repeat
  Construct a set of ILM partial labeling $\mathcal{X}$ described in Section 3.3;
  for $x_S \in \mathcal{X}$ **do**
    repeat
      Test x_S using k-condition;
      if x_S *fails the test* **then**
        Find x_i causing violation;
        $S \leftarrow S \setminus \{i\}$;
      end
    until x_S *passes the test or* $S = \emptyset$;
    $x_W \leftarrow x_W \oplus x_S$; $W \leftarrow W \cup S$;
    $\mathcal{L}_i \leftarrow \{(x_S)_i\}, \forall i \in S$;
  end
until *converge or after* τ *iterations*;
return x_W as the persistent partial labeling of $E(x)$;
Algorithm 2: Persistency Construction

Theoretical connection to [23, 24, 25] The autarky property in (3) is a special case of the improving mapping described in [23]. Our sufficient conditions in (6) is a special case of the partial optimality criterions described in [25]. However, checking the sufficient conditions in [23, 24, 25] require a general MRF inference solver as a subroutine, which is computational expensive. We proposed a set of computational tractable sufficient conditions and approximation algorithm in (7), (8). Therefore, while the sufficient conditions in [23, 24, 25] are tighter, our conditions can be

checked more efficiently.

3.2. Persistency construction problem

In Section 3.1, we described a hierarchy of sufficient conditions and their sound approximations to check persistency of an ILM partial labeling. Now, we will use these conditions as a subroutine to construct a persistent partial labeling for a given energy $E(x)$.

The method is shown in Algorithm 2. Assume we can find a set of ILM partial labelings $\mathcal{X}$ as candidates (we defer a discussion of how to do this until Section 3.3). For each $x_S \in \mathcal{X}$, we adopt a shrinking scheme. We will apply the k-condition test[6] or its approximation to check the persistency of x_S. The test will either proves x_S is persistent or reports some B's violating (7); we will shrink S by removing $i \in B$ with the minimum $\delta_i(x_i) + \sum_{(i,j)\in E} \delta_{ij}^i(x_i, .)$ value for each violated B. If we apply our approximation to the k-condition, we will remove the remaining variables in U from S. We repeat this procedure until x_S satisfies the k-condition. Now we can composite all the persistent partial labelings we found together. It's easy to see the composition of persistent partial labelings is still persistent by definition, which proves that our algorithm is sound.

Finally, similar to the iterative idea in DEE, after determining x_S to be persistent, we can update $E(x)$ by fixing x_S without changing the minimizer. This in turn can potentially find additional persistent variables. We iteratively run the procedure described above until it converges or reaches the pre-defined stopping parameter τ.

Let P_{DEE} be the set of persistent variables DEE found, and P_{PR} the set of persistent variables our algorithm found at convergence. Our algorithm always finds at least as many persistent variables.

Theorem 7. *$P_{\text{DEE}} \subseteq P_{\text{PR}}$ for binary MRFs.*

Proof. We prove this by contradiction. Suppose $\exists x_i \in P_{\text{DEE}}, x_i \notin P_{\text{PR}}$, then x_i satisfies the 1-condition at convergence of PR. This contradicts our assumption that PR has converged, since we should have added x_i into P_{PR}. A detailed proof is provided in the supplementary material. □

Example We note that DEE is based on such a strong local condition that it may fail even in extremely simple cases. Consider a binary Potts MRF with two variables x_i, x_j such that $\theta_i(0) = \theta_j(0) = 0, \theta_i(1) = \theta_j(1) = a \geq 0, \theta_{ij}(0,0) = \theta_{ij}(1,1) = 0, \theta_{ij}(0,1) = \theta_{ij}(1,0) = b > a$. DEE cannot determine any of the variables to be persistent while our approach will easily find that $x_i = x_j = 0$ is a persistent partial labeling. Our experiments demonstrate that our approach indeed finds significantly more persistent variables than DEE.

[6]To be completely precise, for the corner case of a tiny labeling with $|S| < k$, we would test the $|S|$-condition instead of the k-condition.

3.3. ILM labeling construction

In this section, we will show how to construct a candidate set $\mathcal{X}$ of ILM partial labelings. Ideally, we want to start Algorithm 2 with as large a partial labeling as possible. We can show that for a wide family of energy functions used in typical vision problem, we can efficiently find the maximum ILM partial labeling, and even for an arbitrary MRF we can guarantee an ILM partial labeling of at least size 2. We consider three special cases that are widely used in vision: weakly associative energies, binary submodular, and binary non-submodular. Finally, we discuss the case of an arbitrary multi-label MRF.

Definition 8 (Weakly associative energy)**.** *$E(x)$ is called weakly associative if all of its pairwise costs satisfy $\theta_{ij}(x_i, x_j) \geq 0$ and $\theta_{ij}(x_i, x_j) = 0$ when $x_i = x_j$.*

Weakly associative: It's easy to see that any constant labeling (i.e., all the variables take the same value) is ILM. So for each label α, we can let $S := \{i \mid \alpha \in \mathcal{L}_i\}$ and $x_S := \vec{\alpha}$ then put it into $\mathcal{X}$.

Binary submodular: We use the reparameterization scheme introduced in [14] to equivalently transform the energy function into a weakly associative one. Therefore, we can put the maximum constant partial labeling with label 0 and 1 into our $\mathcal{X}$.

Binary non-submodular: Again, we use the reparameterization scheme introduced in [14]. Now, all the submodular terms will be transformed to be weakly associative. Meanwhile, all the supmodular terms will be transformed as $\theta_{ij}(0,0) = \theta_{ij}(1,1) = 0$ and $\theta_{ij}(0,1) = \theta_{ij}(1,0) = c < 0$ with $(0,1)$ and $(1,0)$ as the local minimizer. Therefore, in the ILM partial labeling, we want x_i and x_j to take the same value when θ_{ij} is submodular and to take different values otherwise. We use a greedy approach[7] to find a large enough ILM partial labeling. After we find the first ILM partial labeling, we can add it into $\mathcal{X}$ and iteratively run the greedy algorithm on the remaining variables in $V \backslash S$ to find more ILM partial labelings to be added into $\mathcal{X}$.

Arbitary multilabel: The Potts model and truncated L_p prior, the two most widely used multilabel pairwise terms in vision, are weakly associative. Therefore, we can just construct the maximum constant labeling for each label and add them into $\mathcal{X}$. For an arbitrary multi-label energy, it's hard to find the maximum ILM partial labeling. However, we can still use the greedy algorithm we used in the binary non-submodular case as a good heuristics in practice. This will return multiple ILM partial labelings with size at least 2, which is still better than checking persistency on a single variable as DEE does.

[7]Starting from $x_S = x_1 \in \mathcal{L}_1$, then for $i = 2, 3, \ldots, n$, when we can find $x_i \in \mathcal{L}_i$ such that compositing x_i into x_S is still ILM, then do so.

4. Experiments

Approaches We will focus on three partial optimality based preprocessing techniques in the experiment section, namely DEE [20], Kovtun's approach [16, 17] and our approach. They will be referred as DEE, KOVTUN and PR (Persistency Relaxation) respectively. We will use PR-k to refer our approach using k-conditions and PR-k-APX when we use its approximation. We experimented with MQPBO [12], but it was too slow to be competitive. LP-based approaches [23, 24, 25] are also not considered due to their computational overhead, which is documented in [24]. We will apply α-expansion for MRF inference [4], using the max-flow algorithm of [3]. We will refer to α-expansion algorithm without any preprocessing technique as α-EXP, which is the baseline against which we compare all other approaches.

The α-expansion algorithm, like most move-making techniques, reduces the multi-label MRF inference problem to a series of binary MRF inference problems. Therefore, we can either (1) apply partial optimality based preprocessing techniques to the multi-label MRF directly and use α-EXP to infer the remaining variables, or (2) in each iteration of α-EXP, apply the preprocessing technique to the induced binary MRF.[8] We will refer to approach (1) as mDEE and mPR, and to (2) as iDEE and iPR. As a small optimization, for iDEE and iPR we only determined which variables do not switch to the new label, except for on the first iteration through the label set. Note that KOVTUN can only be used in approach (1) since it degenerates to QPBO for the induced binary problem, which is equivalent to the max-flow problem α-EXP needs to solve in each iteration.

Dataset We conducted experiments on a variety of computer vision benchmarks for MRF inference, including brain-MRI [6], color segmentation [18], inpainting [5], Middlebury MRF dataset (including stereo, image inpainting and photomontage tasks) [26] and scene decomposition [9]. All these datasets are wrapped in OpenGM2 [11] and are available online. Table 2 briefly summarizes the scale of each dataset. Note that the first three datasets use the Potts model, so the binary subproblem is submodular, while the last two have non-submodular (non-weakly associative) subproblems.

Experimental Environment All the experiments were executed on a single machine with dual 3GHz Intel i7 Core and 16GB 1600 MHz DDR3 memory.

Measurement We will evaluate the different approaches in two respects. First, we report the improvement in overall running time for the preprocessing and the inference on the remaining undetermined variables. We use α-EXP as the baseline and report the speedup for other methods compared to α-EXP. The reported numbers are averaged over all instances for each dataset. Second, we report the size of the partial optimal labeling found by the various preprocessing methods. The reported numbers are first averaged for each iteration of α-EXP in each instance, then averaged over all instances in one dataset.

[8] We can also combine these two approaches, i.e., applying preprecessing for both the multi-label MRF and each induced binary MRF. Our experiments shows that it will only provide marginal improvement over only applying preprocessing to each induced binary MRF, so we don't report the results in this paper.

Table 2. Dataset Description

Dataset	$\lvert V\rvert$	$\lvert\mathcal{L}\rvert$	# Instances
Brain MRI	785,540–1,413,972	5	8
Color Seg	76,800–86,400	3–12	9
Inpainting	14400	4	2
Middlebury	21,838–514,080	5–256	7
Scene Decomp	150–208	8	715

Experimental results We summarize the experimental results on several benchmarks in Figure 3; the raw numbers behind this figure are provided in the supplementary material. Besides the baseline technique α-EXP we also show results from iDEE and KOVTUN. Our approaches obtain a 1.5x-12x speedup compared to α-EXP and label significantly more variables than all other methods. For some specific instances, the speedup can be up to 40x; our speedup numbers, of course, include the cost of pre-processing. Note that KOVTUN was too slow on the Middlebury dataset to be competitive, running at least 5x slower than the baseline algorithm α-EXP. This suggests that when we have a large label set, it's very hard to compute partial optimality on the multi-label MRF directly, which is a major limitation of KOVTUN. We can also see PR-based methods find significantly more persistent variables than all the baseline methods. Per instance analysis indicates that our methods are superior on almost all instances.

Figure 3 also illustrate the power of the relaxation hierarchy we proposed. For example, on Color-Seg-n4 dataset, iPR-1 finds 14% more partial persistent variables than iDEE, iPR-2-APX finds additional 10% more than iPR-1. The gap between iPR-3-APX and iPR-2-APX are less significant, but still exists. It indicates that PR-based approaches significantly outperform the baseline DEE. The further we utilize the hierarchy, the more variables we can label, although the marginal gain is diminishing.

We also have run experiments with solving the multi-label problem directly, described in the supplemental material. In the multi-label setting our mPR-based methods also outperforms mDEE. However, solving the induced binary problem via expansion moves generally seems to be a better approach.

Our algorithms have one parameter, the maximum number of iterations τ. Figure 4 and 5 illustrate running DEE

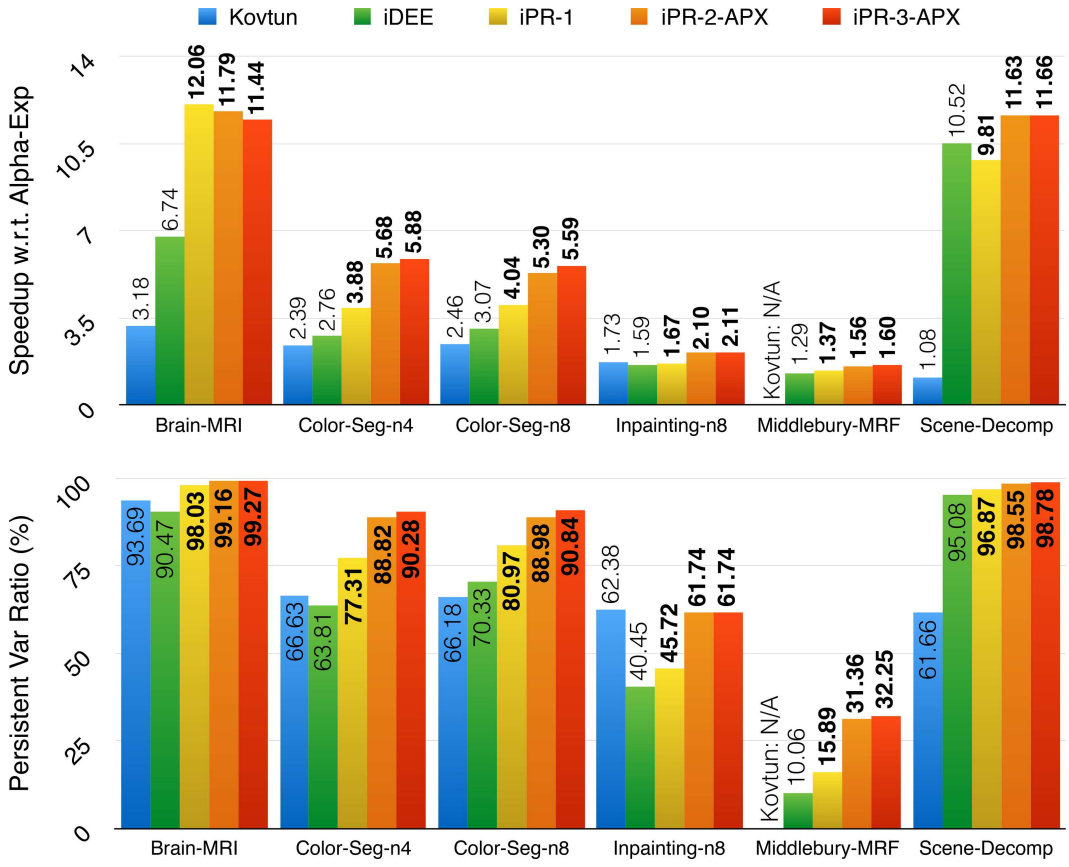

Figure 3. Performance of various methods in terms of speedup and percentage of persistent variables. Higher numbers indicate better performance. Our three methods are at right, with numbers on the chart in bold. KOVTUN was too slow on the Middlebury-MRF dataset.

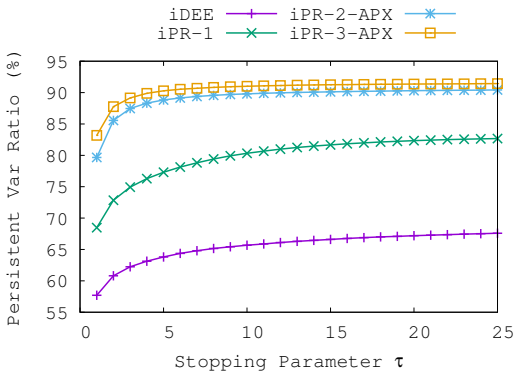

Figure 4. Persistent variables ratio vs. stopping parameter τ.

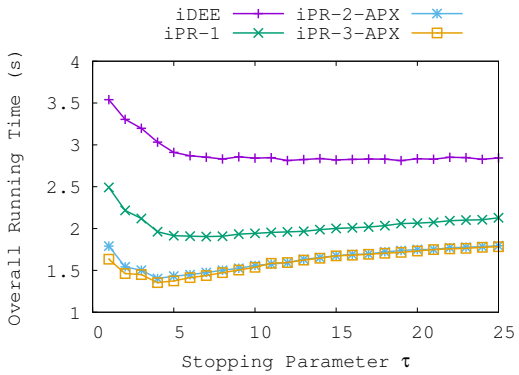

Figure 5. Overall running time v.s. stopping parameter τ.

and PR-based methods with different τ's on the Color-Seg-n4 dataset. Similar results are achieved on other datasets, which we report in the supplementary material. We can see the persistent var ratio converges very quickly with the growth of τ. In general, the overall running time decreases firstly and then increases due to it's a tradeoff between the speed and the quality of the preprocessing step. The proposed approach is not very sensitive to the choice of this stopping parameter, low total running time can be achieved in a very broad range. PR-based approaches significantly outperform DEE and other baseline methods no matter which τ we choose. Figure 3 was computed with $\tau = 5$, but other choices produce similar results.

Acknowledgements This research was supported by NSF grants IIS-1161860 and IIS-1447473. We thank Endre Boros and the anonymous reviewers for helpful comments.

References

[1] K. Alahari, P. Kohli, and P. H. Torr. Dynamic hybrid algorithms for MAP inference in discrete MRFs. *TPAMI*, 32(10):1846–1857, 2010. 3

[2] E. Boros and P. L. Hammer. Pseudo-boolean optimization. *Discrete Applied Mathematics*, 123(1):155–225, 2002. 1, 2, 3

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *TPAMI*, 26(9):1124–1137, 2004. 7

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23(11):1222–1239, 2001. 1, 2, 7

[5] A. Chambolle, D. Cremers, and T. Pock. A convex approach to minimal partitions. *SIAM Journal on Imaging Sciences*, 5(4):1113–1158, 2012. 7

[6] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, G. B. Pike, and A. C. Evans. Brainweb: Online interface to a 3D MRI simulated brain database. In *NeuroImage*, 1997. 7

[7] J. Desmet, M. D. Maeyer, B. Hazes, and I. Lasters. The dead-end elimination theorem and its use in protein side-chain positioning. *Nature*, 356(6369):539–542, 1992. 1, 2

[8] P. F. Felzenszwalb and R. Zabih. Dynamic programming and graph algorithms in computer vision. *TPAMI*, 33(4):721–740, 2011. 2

[9] S. Gould, R. Fulton, and D. Koller. Decomposing a scene into geometric and semantically consistent regions. In *ICCV*, pages 1–8, 2009. 7

[10] F. Kahl and P. Strandmark. Generalized roof duality. *Discrete applied mathematics*, 160(16):2419–2434, 2012. 1

[11] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, T. Kroeger, B. X. Kausler, J. Lellmann, B. Savchynskyy, N. Komodakis, and C. Rother. A comparative study of modern inference techniques for discrete energy minimization problems. *IJCV*, 2015. 1, 2, 7

[12] P. Kohli, A. Shekhovtsov, C. Rother, V. Kolmogorov, and P. Torr. On partial optimality in multi-label MRFs. In *ICML*, pages 480–487, 2008. 1, 2, 7

[13] V. Kolmogorov. Generalized roof duality and bisubmodular functions. In *NIPS*, pages 1144–1152, 2010. 1

[14] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. *TPAMI*, 29(7):1274–1279, 2007. Earlier version appears as technical report MSR-TR-2006-100. 1, 2, 6

[15] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *TPAMI*, 26(2):147–159, 2004. 1

[16] I. Kovtun. Partial optimal labeling search for a NP-hard subclass of (max,+) problems. In *Pattern Recognition*, pages 402–409. 2003. 1, 2, 7

[17] I. Kovtun. Image segmentation based on sufficient conditions of optimality in NP-complete classes of structural labelling problem. *Ukrainian. PhD thesis. IRTC ITS National Academy of Sciences, Ukraine*, 2004. 1, 2, 7

[18] J. Lellmann and C. Schnörr. Continuous multiclass labeling approaches and algorithms. *Journal on Imaging Sciences*, 4(4):1049–1096, 2011. 7

[19] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for Markov Random Field optimization. *TPAMI*, 32(8):1392–1405, 2010. 2

[20] M. L. Radhakrishnan and S. L. Su. Dead-end elimination as a heuristic for min-cut image segmentation. In *ICIP*, pages 2429–2432, 2006. 3, 7

[21] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary MRFs via extended roof duality. In *CVPR*, 2007. 1, 2

[22] B. Savchynskyy, J. H. Kappes, P. Swoboda, and C. Schnörr. Global MAP-optimality by shrinking the combinatorial search area with convex relaxation. In *NIPS*, pages 1950–1958, 2013. 1

[23] A. Shekhovtsov. Maximum persistency in energy minimization. In *CVPR*, pages 1162–1169, 2014. 1, 2, 5, 7

[24] A. Shekhovtsov, P. Swoboda, and B. Savchynskyy. Maximum persistency via iterative relaxed inference with graphical models. In *CVPR*, pages 521–529, 2015. 1, 2, 5, 7

[25] P. Swoboda, B. Savchynskyy, J. H. Kappes, and C. Schnorr. Partial optimality by pruning for MAP-inference with general graphical models. In *CVPR*, pages 1170–1177, 2014. 1, 2, 5, 7

[26] R. Szeliski, R. Zabih, D. Scharstein, O. Veksler, V. Kolmogorov, A. Agarwala, M. Tappen, and C. Rother. A comparative study of energy minimization methods for Markov Random Fields. *TPAMI*, 30(6):1068–1080, 2008. 1, 2, 7

[27] H. Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, pages 509–533, 1935.

[28] T. Windheuser, H. Ishikawa, and D. Cremers. Generalized roof duality for multi-label optimization: optimal lower bounds and persistency. In *ECCV*, pages 400–413. 2012. 1, 2