



HAL
open science

Rounding-based Moves for Semi-Metric Labeling

M. Pawan Kumar, Puneet K. Dokania

► **To cite this version:**

M. Pawan Kumar, Puneet K. Dokania. Rounding-based Moves for Semi-Metric Labeling . Journal of Machine Learning Research, 2016, 17, pp.1 - 42. hal-01577719

HAL Id: hal-01577719

<https://inria.hal.science/hal-01577719>

Submitted on 27 Aug 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Rounding-based Moves for Semi-Metric Labeling

M. Pawan Kumar

*Department of Engineering Science
University of Oxford
Parks Road OX1 3PJ
United Kingdom*

PAWAN@ROBOTS.OX.AC.UK

Puneet K. Dokania

*Center for Visual Computing
CentraleSupélec
92295 Châtenay-Malabry
France*

PUNEET.KUMAR@ECP.FR

Editor: Jeff Bilmes

Abstract

Semi-metric labeling is a special case of energy minimization for pairwise Markov random fields. The energy function consists of arbitrary unary potentials, and pairwise potentials that are proportional to a given semi-metric distance function over the label set. Popular methods for solving semi-metric labeling include (i) move-making algorithms, which iteratively solve a minimum *st*-cut problem; and (ii) the linear programming (LP) relaxation based approach. In order to convert the fractional solution of the LP relaxation to an integer solution, several randomized rounding procedures have been developed in the literature. We consider a large class of parallel rounding procedures, and design move-making algorithms that closely mimic them. We prove that the multiplicative bound of a move-making algorithm exactly matches the approximation factor of the corresponding rounding procedure for any arbitrary distance function. Our analysis includes all known results for move-making algorithms as special cases.

Keywords: semi-metric labeling, move-making algorithms, linear programming relaxation, multiplicative bounds

1. Introduction

A Markov random field (MRF) is a graph whose vertices are random variables, and whose edges specify a neighborhood over the random variables. Each random variable can be assigned a value from a set of labels, resulting in a labeling of the MRF. The putative labelings of an MRF are quantitatively distinguished from each other by an energy function, which is the sum of potential functions that depend on the cliques of the graph. An important optimization problem associated with the MRF framework is energy minimization, that is, finding a labeling with the minimum energy.

Semi-metric labeling is a special case of energy minimization, which models several useful low-level vision tasks (Boykov et al., 1998, 1999; Szeliski et al., 2008). It is characterized by a finite, discrete label set and a semi-metric distance function over the labels. The energy function in semi-

metric labeling consists of arbitrary unary potentials and pairwise potentials that are proportional to the distance between the labels assigned to them. The problem is known to be NP-hard (Veksler, 1999). Two popular approaches for semi-metric labeling are: (i) move-making algorithms (Boykov et al., 1999; Gupta and Tardos, 2000; Kumar and Koller, 2009; Kumar and Torr, 2008; Veksler, 2007), which iteratively improve the labeling by solving a minimum *st*-cut problem; and (ii) linear programming (LP) relaxation (Chekuri et al., 2001; Koster et al., 1998; Schlesinger, 1976; Wainwright et al., 2005), which is obtained by dropping the integral constraints in the corresponding integer programming formulation. Move-making algorithms are very efficient due to the availability of fast minimum *st*-cut solvers (Boykov and Kolmogorov, 2004) and are very popular in the computer vision community. In contrast, the LP relaxation is significantly slower, despite the development of several specialized solvers (Globerson and Jaakkola, 2007; Hazan and Shashua, 2008; Kolmogorov, 2006; Komodakis et al., 2007; Ravikumar et al., 2008; Tarlow et al., 2011; Wainwright et al., 2005; Weiss et al., 2007; Werner, 2007, 2010). However, when used in conjunction with randomized rounding algorithms, the LP relaxation provides the best known polynomial-time theoretical guarantees for semi-metric labeling (Archer et al., 2004; Chekuri et al., 2001; Kleinberg and Tardos, 1999).

At first sight, the difference between move-making algorithms and the LP relaxation appears to be the standard accuracy vs. speed trade-off. However, we prove that, for any arbitrary semi-metric distance function, it is possible to design move-making algorithms that match the theoretical guarantees of a large class of randomized rounding procedures, which we call parallel rounding. Our proofs are constructive, which allows us to test the rounding-based move-making algorithms empirically. Our experimental results confirm that rounding-based moves provide similar accuracy to the LP relaxation while being significantly faster.

2. Related Work

As our work is concerned with only semi-metric labeling, we will focus on the related work for this special case of energy minimization. For a more general survey, we refer the reader to (Wang et al., 2013), and for a thorough empirical comparison of the various algorithms we refer the reader to (Kappes et al., 2015; Szeliski et al., 2008).

The earliest known theoretical guarantee of a move-making algorithm was provided by Veksler (1999), who presented an analysis of the popular expansion move-making algorithm. The analysis showed that the expansion algorithm and the parallel rounding procedure of Kleinberg and Tardos (1999) provide matching guarantees for the uniform distance function. Komodakis et al. (2008) also proposed a move-making algorithm based on the primal-dual scheme of optimizing the LP relaxation. The resulting algorithm is similar to the expansion algorithm in terms of its theoretical guarantees, but provides a principled formulation for improving its efficiency in dynamic settings. Despite the existing analysis of expansion-like algorithms, parallel rounding procedures (Chekuri et al., 2001; Kleinberg and Tardos, 1999) are known to perform better for several distance functions of interest such as the truncated linear distance, the truncated quadratic distance, and the hierarchically well-separated tree (HST) metric.

Kumar and Torr (2008) proposed the range expansion move-making algorithm that matches the guarantees of the rounding procedure of Chekuri et al. (2001) for the truncated linear and quadratic distance. Kumar and Koller (2009) proposed a hierarchical move-making algorithm that matches the guarantees of the rounding procedure of Kleinberg and Tardos (1999) for the HST metric. However, their analysis is restricted to special cases of semi-metric labeling, whereas our work does not make

any assumptions on the form of the semi-metric distance function. We note that, recently, a more general algorithm has also been proposed for non-convex priors (Ajanthan et al., 2014). However, this generalized algorithm does not provide any strong theoretical guarantees on the quality of the solution, which is the key focus of our work.

While our analysis focuses on parallel rounding, it is also worth noting that Archer et al. (2004) have proposed a serial rounding procedure for metric labeling (which is a special case of semi-metric labeling where the distance function also satisfies the triangular inequality property). The theoretical guarantee of serial rounding depends on the *tree decomposibility* of a distance function. Given an MRF with n random variables, the tree decomposibility for any metric distance function is guaranteed to be less than or equal to $O(\log n)$. This worst-case theoretical guarantee of $O(\log n)$ over all metric distance functions can be matched by a mixture-of-tree algorithm that can be derived from the work of Andrew et al. (2011). Specifically, Andrew et al. (2011) describe an approximate algorithm for capacitated metric labeling that builds on the cut-based decomposition method of Racke (2008). Ignoring the capacity constraints, which are not present in the standard metric labeling problem, results in an approximation of the original MRF using an $O(m \log n)$ mixture of tree-structured MRFs. Here, m is the number of edges in the MRF, and n is the number of random variables. Solving the metric labeling problem over each tree-structured MRF optimally using belief propagation (Pearl, 1998), and choosing the best labeling in terms of the original problem, provides the $O(\log n)$ guarantee. We refer the reader to (Andrew et al., 2011) for details.

Note that while the mixture-of-tree algorithm provides matching worst-case guarantees to serial rounding, it does not provide a matching guarantee for a given distance function. In other words, the guarantee of the mixture-of-tree algorithm does not depend on the tree decomposibility of the distance function. The existence or the impossibility of a matching combinatorial algorithm for serial rounding remains an open question.

3. Our Contributions

Our paper can be thought of as a generalization of (Kumar and Koller, 2009; Kumar and Torr, 2008; Veksler, 1999). Indeed, the move-making algorithms proposed in this paper are simple extensions of the expansion, the range expansion and the hierarchical move-making algorithms. The novelty of our work lies in our analysis, which improves on the state of the art in two significant aspects. First, we do not place any assumptions on the form of the distance function except that it is a semi-metric distance. This is in contrast to previous works that only focus on special cases such as the uniform metric, the truncated convex models or the HST metric. Second, we show that the matching guarantees provided by our move-making algorithms and the parallel rounding procedures are *tight*, that is, they cannot be improved through a more careful analysis.

A preliminary version of this article has appeared as (Kumar, 2014). There are two significant differences between the preliminary version and the current paper. First, we provide detailed proofs of all the theorems, which were omitted from the preliminary version due to a lack of space. Second, we provide an experimental comparison of the rounding-based move-making algorithms with the state of the art methods for semi-metric labeling, including those that are directly related to the LP relaxation such as TRW-S (Kolmogorov, 2006), and those that are indirectly related to the LP relaxation such as (Andrew et al., 2011).

Finally, we note that since the publication of the preliminary version, we have also extended rounding-based move-making algorithms to special cases of high-order potentials. This includes the

interval move-making algorithm for truncated max-of-convex potentials (Pansari and Kumar, 2015) and the hierarchical move-making algorithm for parsimonious labeling (Dokania and Kumar, 2015).

4. Preliminaries

4.1 Semi-metric Labeling

The problem of semi-metric labeling is defined over an undirected graph $\mathbf{G} = (\mathbf{X}, \mathbf{E})$. The vertices $\mathbf{X} = \{X_1, X_2, \dots, X_n\}$ are random variables, and the edges \mathbf{E} specify a neighborhood relationship over the random variables. Each random variable can be assigned a value from the label set $\mathbf{L} = \{l_1, l_2, \dots, l_h\}$. We assume that we are also provided with a semi-metric distance function $d : \mathbf{L} \times \mathbf{L} \rightarrow \mathbb{R}^+$ over the labels. Recall that a semi-metric distance function satisfies the following properties: $d(l_i, l_j) \geq 0$ for all $l_i, l_j \in \mathbf{L}$, and $d(l_i, l_j) = 0$ if and only if $i = j$. Furthermore, a distance function is said to be metric if, in addition to the above condition, it also satisfies the triangular inequality, that is, $d(l_i, l_j) + d(l_j, l_k) \geq d(l_i, l_k)$ for all $l_i, l_j, l_k \in \mathbf{L}$.

We refer to an assignment of values to all the random variables as a labeling. In other words, a labeling is a vector $\mathbf{x} \in \mathbf{L}^n$, which specifies the label x_a assigned to each random variable X_a . The h^n different labelings are quantitatively distinguished from each other by an energy function $Q(\mathbf{x})$, which is defined as follows:

$$Q(\mathbf{x}) = \sum_{X_a \in \mathbf{X}} \theta_a(x_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} d(x_a, x_b).$$

Here, the unary potentials $\theta_a(\cdot)$ are arbitrary, and the edge weights w_{ab} are non-negative. Semi-metric labeling requires us to find a labeling with the minimum energy. It is known to be NP-hard.

4.2 Multiplicative Bound

As semi-metric labeling plays a central role in low-level vision, several approximate algorithms have been proposed in the literature. A common theoretical measure of accuracy for an approximate algorithm is the multiplicative bound. In this work, we are interested in the multiplicative bound of an algorithm with respect to a distance function. Formally, given a distance function d , the multiplicative bound of an algorithm is said to be B if the following condition is satisfied for all possible values of unary potentials $\theta_a(\cdot)$ and non-negative edge weights w_{ab} :

$$\sum_{X_a \in \mathbf{X}} \theta_a(\hat{x}_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} d(\hat{x}_a, \hat{x}_b) \leq \sum_{X_a \in \mathbf{X}} \theta_a(x_a^*) + B \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} d(x_a^*, x_b^*). \quad (1)$$

Here, $\hat{\mathbf{x}}$ is the labeling estimated by the algorithm for the given values of unary potentials and edge weights, and \mathbf{x}^* is an optimal labeling. Multiplicative bounds are greater than or equal to 1, and are invariant to reparameterizations of the unary potentials. A multiplicative bound B is said to be tight if the above inequality holds as an equality for some value of unary potentials and edge weights.

4.3 Linear Programming Relaxation

An overcomplete representation of a labeling can be specified using the following variables: (i) unary variables $y_a(i) \in \{0, 1\}$ for all $X_a \in \mathbf{X}$ and $l_i \in \mathbf{L}$ such that $y_a(i) = 1$ if and only if X_a is assigned the label l_i ; and (ii) pairwise variables $y_{ab}(i, j) \in \{0, 1\}$ for all $(X_a, X_b) \in \mathbf{E}$ and $l_i, l_j \in \mathbf{L}$ such

that $y_{ab}(i, j) = 1$ if and only if X_a and X_b are assigned labels l_i and l_j respectively. This allows us to formulate semi-metric labeling as follows:

$$\begin{aligned}
 \min_{\mathbf{y}} \quad & \sum_{X_a \in \mathbf{X}} \sum_{l_i \in \mathbf{L}} \theta_a(l_i) y_a(i) + \sum_{(X_a, X_b) \in \mathbf{E}} \sum_{l_i, l_j \in \mathbf{L}} w_{ab} d(l_i, l_j) y_{ab}(i, j), \\
 \text{s.t.} \quad & \sum_{l_i \in \mathbf{L}} y_a(i) = 1, \forall X_a \in \mathbf{X}, \\
 & \sum_{l_j \in \mathbf{L}} y_{ab}(i, j) = y_a(i), \forall (X_a, X_b) \in \mathbf{E}, l_i \in \mathbf{L}, \\
 & \sum_{l_i \in \mathbf{L}} y_{ab}(i, j) = y_b(j), \forall (X_a, X_b) \in \mathbf{E}, l_j \in \mathbf{L}, \\
 & y_a(i) \in \{0, 1\}, y_{ab}(i, j) \in \{0, 1\}, \forall X_a \in \mathbf{X}, (X_a, X_b) \in \mathbf{E}, l_i, l_j \in \mathbf{L}.
 \end{aligned}$$

The first set of constraints ensures that each random variables is assigned exactly one label. The second and third sets of constraints ensure that, for binary optimization variables, $y_{ab}(i, j) = y_a(i)y_b(j)$. By relaxing the final set of constraints such that the optimization variables can take any value between 0 and 1 inclusive, we obtain a linear program (LP). The computational complexity of solving the LP relaxation is polynomial in the size of the problem.

4.4 Rounding Procedure

In order to prove theoretical guarantees of the LP relaxation, it is common to use a rounding procedure that can convert a feasible fractional solution \mathbf{y} of the LP relaxation to a feasible integer solution $\hat{\mathbf{y}}$ of the integer linear program. Several rounding procedures have been proposed in the literature. In this work, we focus on the randomized parallel rounding procedures proposed by Chekuri et al. (2001) and Kleinberg and Tardos (1999). These procedures have the property that, given a fractional solution \mathbf{y} , the probability of assigning a label $l_i \in \mathbf{L}$ to a random variable $X_a \in \mathbf{X}$ is equal to $y_a(i)$, that is,

$$\Pr(\hat{y}_a(i) = 1) = y_a(i). \quad (2)$$

We will describe the various rounding procedures in detail in Sections 5-7. For now, we would like to note that our reason for focusing on the parallel rounding of Chekuri et al. (2001) and Kleinberg and Tardos (1999) is that they provide the best known polynomial-time theoretical guarantees for semi-metric labeling. Specifically, we are interested in their approximation factor, which is defined next.

4.5 Approximation Factor

Given a distance function d , the approximation factor for a rounding procedure is said to be F if the following condition is satisfied for all feasible fractional solutions \mathbf{y} :

$$\mathbb{E} \left(\sum_{l_i, l_j \in \mathbf{L}} d(l_i, l_j) \hat{y}_a(i) \hat{y}_b(j) \right) \leq F \sum_{l_i, l_j \in \mathbf{L}} d(l_i, l_j) y_{ab}(i, j). \quad (3)$$

Here, $\hat{\mathbf{y}}$ refers to the integer solution, and the expectation is taken with respect to the randomized rounding procedure applied to the feasible solution \mathbf{y} .

Given a rounding procedure with an approximation factor of F , an optimal fractional solution \mathbf{y}^* of the LP relaxation can be rounded to a labeling $\hat{\mathbf{y}}$ that satisfies the following condition:

$$\begin{aligned} & \mathbb{E} \left(\sum_{X_a \in \mathbf{X}} \sum_{l_i \in \mathbf{L}} \theta_a(l_i) \hat{y}_a(i) + \sum_{(X_a, X_b) \in \mathbf{E}} \sum_{l_i, l_j \in \mathbf{L}} w_{ab} d(l_i, l_j) \hat{y}_a(i) \hat{y}_b(j) \right) \\ & \leq \sum_{X_a \in \mathbf{X}} \sum_{l_i \in \mathbf{L}} \theta_a(l_i) y_a^*(i) + F \sum_{(X_a, X_b) \in \mathbf{E}} \sum_{l_i, l_j \in \mathbf{L}} w_{ab} d(l_i, l_j) y_{ab}^*(i, j). \end{aligned}$$

The above inequality follows directly from properties (2) and (3). Similar to multiplicative bounds, approximation factors are always greater than or equal to 1, and are invariant to reparameterizations of the unary potentials. An approximation factor F is said to be tight if the above inequality holds as an equality for some value of unary potentials and edge weights.

Approximation factors are closely linked to the integrality gap of the LP relaxation (roughly speaking, the ratio of the optimal value of the integer linear program to the optimal value of the relaxation), which in turn is related to the computational hardness of the semi-metric labeling problem (Manokaran et al., 2008). However, establishing the integrality gap of the LP relaxation for a given distance function is beyond the scope of this work. We are only interested in designing move-making algorithms whose multiplicative bounds match the approximation factors of the parallel rounding procedures.

4.6 Submodular Energy Function

We will use the following important fact throughout this paper. Given an energy function defined using arbitrary unary potentials, non-negative edge weights and a submodular distance function, an optimal labeling can be computed in polynomial time by solving an equivalent minimum *st*-cut problem (Flach and Schlesinger, 2006). Recall that a submodular distance function d' over a label set $\mathbf{L} = \{l_1, l_2, \dots, l_h\}$ satisfies the following properties: (i) $d'(l_i, l_j) \geq 0$ for all $l_i, l_j \in \mathbf{L}$, and $d'(l_i, l_j) = 0$ if and only if $i = j$; and (ii) $d'(l_i, l_j) + d'(l_{i+1}, l_{j+1}) \leq d'(l_i, l_{j+1}) + d'(l_{i+1}, l_j)$ for all $l_i, l_j \in \mathbf{L} \setminus \{l_h\}$ (where \setminus refers to set difference).

5. Complete Rounding and Complete Move

We start with a simple rounding scheme, which we call complete rounding. While complete rounding is not very accurate, it would help illustrate the flavor of our results. We will subsequently consider its generalizations, which have been useful in obtaining the best-known approximation factors for various special cases of semi-metric labeling.

The complete rounding procedure consists of a single stage where we use the set of all unary variables to obtain a labeling (as opposed to other rounding procedures discussed subsequently). Algorithm 1 describes its main steps. Intuitively, it treats the value of the unary variable $y_a(i)$ as the probability of assigning the label l_i to the random variable X_a . It obtains a labeling by sampling from all the distributions $\mathbf{y}_a = [y_a(i), \forall l_i \in \mathbf{L}]$ simultaneously using the same random number $r \in [0, 1]$.

It can be shown that using a different random number to sample the distributions \mathbf{y}_a and \mathbf{y}_b of two neighboring random variables $(X_a, X_b) \in \mathbf{E}$ results in an infinite approximation factor. For example, let $\bar{y}_a(i) = \bar{y}_b(i) = 1/h$ for all $l_i \in \mathbf{L}$, where h is the number of labels. The pairwise variables \bar{y}_{ab} that minimize the energy function are $\bar{y}_{ab}(i, i) = 1/h$ and $\bar{y}_{ab}(i, j) = 0$ when $i \neq j$.

For the above feasible solution of the LP relaxation, the RHS of inequality (3) is 0 for any finite F , while the LHS of inequality (3) is strictly greater than 0 if $h > 1$. However, we will shortly show that using the same random number r for all random variables provides a finite approximation factor.

Algorithm 1 The complete rounding procedure.

input A feasible solution \mathbf{y} of the LP relaxation.

- 1: Pick a real number r uniformly from $[0, 1]$.
 - 2: **for all** $X_a \in \mathbf{X}$ **do**
 - 3: Define $Y_a(0) = 0$ and $Y_a(i) = \sum_{j=1}^i y_a(j)$ for all $l_i \in \mathbf{L}$.
 - 4: Assign the label $l_i \in \mathbf{L}$ to the random variable X_a if $Y_a(i-1) < r \leq Y_a(i)$.
 - 5: **end for**
-

We now turn our attention to designing a move-making algorithm whose multiplicative bound matches the approximation factor of the complete rounding procedure. To this end, we modify the range expansion algorithm proposed by Kumar and Torr (2008) for truncated convex pairwise potentials to a general semi-metric distance function. Our method, which we refer to as the complete move-making algorithm, considers all putative labels of all random variables, and provides an approximate solution in a single iteration. Algorithm 2 describes its two main steps. First, it computes a submodular overestimation of the given semi-metric distance function by solving the following optimization problem:

$$\begin{aligned}
 \bar{d} = & \underset{d'}{\operatorname{argmin}} t & (4) \\
 \text{s.t.} & d'(l_i, l_j) \leq td(l_i, l_j), \forall l_i, l_j \in \mathbf{L}, \\
 & d'(l_i, l_j) \geq d(l_i, l_j), \forall l_i, l_j \in \mathbf{L}, \\
 & d'(l_i, l_j) + d'(l_{i+1}, l_{j+1}) \leq d'(l_i, l_{j+1}) + d'(l_{i+1}, l_j), \forall l_i, l_j \in \mathbf{L} \setminus \{l_h\}.
 \end{aligned}$$

The above problem minimizes the maximum ratio of the estimated distance to the original distance over all pairs of labels, that is,

$$\max_{i \neq j} \frac{d'(l_i, l_j)}{d(l_i, l_j)}.$$

We will refer to the optimal value of problem (4) as the submodular distortion of the distance function d . Second, it replaces the original distance function by the submodular overestimation and computes an approximate solution to the original semi-metric labeling problem by solving a single minimum st -cut problem. Note that, unlike the range expansion algorithm (Kumar and Torr, 2008) that uses the readily available submodular overestimation of a truncated convex distance (namely, the corresponding convex distance function), our approach estimates the submodular overestimation via the LP (4). Since the LP (4) can be solved for any arbitrary distance function, it makes complete move-making more generally applicable.

The following theorem establishes the theoretical guarantees of the complete move-making algorithm and the complete rounding procedure.

Theorem 1 *The tight multiplicative bound of the complete move-making algorithm is equal to the submodular distortion of the distance function. Furthermore, the tight approximation factor of the complete rounding procedure is also equal to the submodular distortion of the distance function.*

Algorithm 2 The complete move-making algorithm.

input Unary potentials $\theta_a(\cdot)$, edge weights w_{ab} , distance function d .

- 1: Compute a submodular overestimation of d by solving problem (4).
- 2: Using the approach of Flach and Schlesinger (2006), solve the following problem via an equivalent minimum st -cut problem:

$$\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x} \in \mathbf{L}^n} \sum_{X_a \in \mathbf{X}} \theta_a(x_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} \bar{d}(x_a, x_b).$$

The proof of Theorem 1 is given in Appendix A. The following corollary of the above theorem was previously stated by Chekuri et al. (2001) without a formal proof.

Corollary 2 *The complete rounding procedure is tight for submodular distance functions, that is, its approximation factor is equal to 1.*

In terms of computational complexities, complete move-making is significantly faster than solving the LP relaxation. Specifically, given an MRF with n random variables and m edges, and a label set with h labels, the LP relaxation requires at least $O(m^3 h^3 \log(m^2 h^3))$ time, since it consists of $O(mh^2)$ optimization variables and $O(mh)$ constraints. In contrast, complete move-making requires $O(nmh^3 \log(m))$ time, since the graph constructed using the method of Flach and Schlesinger (2006) consists of $O(nh)$ nodes and $O(mh^2)$ arcs. Note that complete move-making also requires us to solve the linear program (4). However, since problem (4) is independent of the unary potentials and the edge weights, it only needs to be solved once beforehand in order to compute the approximate solution for any semi-metric labeling problem defined using the distance function d .

6. Interval Rounding and Interval Moves

Theorem 1 implies that the approximation factor of the complete rounding procedure is very large for distance functions that are highly non-submodular. For example, consider the truncated linear distance function defined as follows over a label set $\mathbf{L} = \{l_1, l_2, \dots, l_h\}$:

$$d(l_i, l_j) = \min\{|i - j|, M\}.$$

Here, M is a user specified parameter that determines the maximum distance. The tightest submodular overestimation of the above distance function is the linear distance function, that is, $d(l_i, l_j) = |i - j|$. This implies that the submodular distortion of the truncated linear metric is $(h - 1)/M$, and therefore, the approximation factor for the complete rounding procedure is also $(h - 1)/M$. In order to avoid this large approximation factor, Chekuri et al. (2001) proposed an interval rounding procedure, which captures the intuition that it is beneficial to assign similar labels to as many random variables as possible.

Algorithm 3 provides a description of interval rounding. The rounding procedure chooses an interval of at most q consecutive labels (step 2). It generates a random number r (step 3), and uses it to attempt to assign labels to previously unlabeled random variables from the selected interval (steps 4-7). It can be shown that the overall procedure converges in a polynomial number of iterations with a probability of 1 (Chekuri et al., 2001). Note that if we fix $q = h$ and $z = 1$, interval

rounding becomes equivalent to complete rounding. However, the analyses of Chekuri et al. (2001) and Kleinberg and Tardos (1999) shows that other values of q provide better approximation factors for various special cases.

Algorithm 3 The interval rounding procedure.

input A feasible solution \mathbf{y} of the LP relaxation.

- 1: **repeat**
 - 2: Pick an integer z uniformly from $[-q + 2, h]$. Define an interval of labels $\mathbf{I} = \{l_s, \dots, l_e\}$, where $s = \max\{z, 1\}$ is the start index and $e = \min\{z + q - 1, h\}$ is the end index.
 - 3: Pick a real number r uniformly from $[0, 1]$.
 - 4: **for all** Unlabeled random variables X_a **do**
 - 5: Define $Y_a(0) = 0$ and $Y_a(i) = \sum_{j=s}^{s+i-1} y_a(j)$ for all $i \in \{1, \dots, e - s + 1\}$.
 - 6: Assign the label $l_{s+i-1} \in \mathbf{I}$ to the X_a if $Y_a(i - 1) < r \leq Y_a(i)$.
 - 7: **end for**
 - 8: **until** All random variables have been assigned a label.
-

Our goal is to design a move-making algorithm whose multiplicative bound matches the approximation factor of interval rounding for any choice of q . To this end, we propose the interval move-making algorithm that generalizes the range expansion algorithm (Kumar and Torr, 2008), originally proposed for truncated convex distances, to arbitrary distance functions. Algorithm 4 provides its main steps. The central idea of the method is to improve a given labeling $\hat{\mathbf{x}}$ by allowing each random variable X_a to either retain its current label \hat{x}_a or to choose a new label from an interval of consecutive labels. In more detail, let $\mathbf{I} = \{l_s, \dots, l_e\} \subseteq \mathbf{L}$ be an interval of labels of length at most q (step 4). For the sake of simplicity, let us assume that $\hat{x}_a \notin \mathbf{I}$ for any random variable X_a . We define $\mathbf{I}_a = \mathbf{I} \cup \{\hat{x}_a\}$ (step 5). For each pair of neighboring random variables $(X_a, X_b) \in \mathbf{E}$, we compute a submodular distance function $\bar{d}_{\hat{x}_a, \hat{x}_b} : \mathbf{I}_a \times \mathbf{I}_b \rightarrow \mathbb{R}^+$ by solving the following linear program (step 6):

$$\begin{aligned}
 \bar{d}_{\hat{x}_a, \hat{x}_b} = & \operatorname{argmin}_{d'} t & (5) \\
 \text{s.t.} & d'(l_i, l_j) \leq td(l_i, l_j), \forall l_i \in \mathbf{I}_a, l_j \in \mathbf{I}_b, \\
 & d'(l_i, l_j) \geq d(l_i, l_j), \forall l_i \in \mathbf{I}_a, l_j \in \mathbf{I}_b, \\
 & d'(l_i, l_j) + d'(l_{i+1}, l_{j+1}) \leq d'(l_i, l_{j+1}) + d'(l_{i+1}, l_j), \forall l_i, l_j \in \mathbf{I} \setminus \{l_e\}, \\
 & d'(l_i, l_e) + d'(l_{i+1}, \hat{x}_b) \leq d'(l_i, \hat{x}_b) + d'(l_{i+1}, l_e), \forall l_i \in \mathbf{I} \setminus \{l_e\}, \\
 & d'(l_e, l_j) + d'(\hat{x}_a, l_{j+1}) \leq d'(l_e, l_{j+1}) + d'(\hat{x}_a, l_j), \forall l_j \in \mathbf{I} \setminus \{l_e\}, \\
 & d'(l_e, l_e) + d(\hat{x}_a, \hat{x}_b) \leq d'(l_e, \hat{x}_b) + d'(\hat{x}_a, l_e).
 \end{aligned}$$

Similar to problem (4), the above problem minimizes the maximum ratio of the estimated distance to the original distance. However, instead of introducing constraints for all pairs of labels, it only considers pairs of labels l_i and l_j where $l_i \in \mathbf{I}_a$ and $l_j \in \mathbf{I}_b$. Furthermore, it does not modify the distance between the current labels \hat{x}_a and \hat{x}_b (as can be seen in the last constraint of problem (5)).

Given the submodular distance functions $\bar{d}_{\hat{x}_a, \hat{x}_b}$, we can compute a new labeling $\bar{\mathbf{x}}$ by solving the following optimization problem via minimum st -cut using the method of Flach and Schlesinger

(2006) (step 7):

$$\begin{aligned} \bar{\mathbf{x}} = & \underset{\mathbf{x}}{\operatorname{argmin}} \sum_{X_a \in \mathbf{X}} \theta_a(x_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(x_a, x_b) \\ \text{s.t. } & x_a \in \mathbf{I}_a, \forall X_a \in \mathbf{X}. \end{aligned} \quad (6)$$

If the energy of the new labeling $\bar{\mathbf{x}}$ is less than that of the current labeling $\hat{\mathbf{x}}$, then we update our labeling to $\bar{\mathbf{x}}$ (steps 8-10). Otherwise, we retain the current estimate of the labeling and consider another interval. The algorithm converges when the energy does not decrease for any interval of length at most q . Note that, once again, the main difference between interval move-making and the range expansion algorithm is the use of an appropriate optimization problem, namely the LP (5), to obtain a submodular overestimation of the given distance function. This allows us to use the interval move-making algorithm for the general semi-metric labeling problem, instead of focusing on only truncated convex models.

Algorithm 4 The interval move-making algorithm.

input Unary potentials $\theta_a(\cdot)$, edge weights w_{ab} , distance function d , initial labeling \mathbf{x}^0 .

- 1: Set current labeling to initial labeling, that is, $\hat{\mathbf{x}} = \mathbf{x}^0$.
 - 2: **repeat**
 - 3: **for all** $z \in [-q + 2, h]$ **do**
 - 4: Define an interval of labels $\mathbf{I} = \{l_s, \dots, l_e\}$, where $s = \max\{z, 1\}$ is the start index and $e = \min\{z + q - 1, h\}$ is the end index.
 - 5: Define $\mathbf{I}_a = \mathbf{I} \cup \{\hat{x}_a\}$ for all random variables $X_a \in \mathbf{X}$.
 - 6: Obtain submodular overestimates $\bar{d}_{\hat{x}_a, \hat{x}_b}$ for each pair of neighboring random variables $(X_a, X_b) \in \mathbf{E}$ by solving problem (5).
 - 7: Obtain a new labeling $\bar{\mathbf{x}}$ by solving problem (6).
 - 8: **if** Energy of $\bar{\mathbf{x}}$ is less than energy of $\hat{\mathbf{x}}$ **then**
 - 9: Update $\hat{\mathbf{x}} = \bar{\mathbf{x}}$.
 - 10: **end if**
 - 11: **end for**
 - 12: **until** Energy cannot be decreased further.
-

The following theorem establishes the theoretical guarantees of the interval move-making algorithm and the interval rounding procedure.

Theorem 3 *The tight multiplicative bound of the interval move-making algorithm is equal to the tight approximation factor of the interval rounding procedure.*

The proof of Theorem 3 is given in Appendix B. While Algorithms 3 and 4 use intervals of consecutive labels, they can easily be modified to use subsets of (potentially non-consecutive) labels. Our analysis could be extended to show that the multiplicative bound of the resulting subset move-making algorithm matches the approximation factor of the subset rounding procedure. However, our reason for focusing on intervals of consecutive labels is that several special cases of Theorem 3 have previously been considered separately in the literature (Gupta and Tardos, 2000; Kumar and Koller, 2009; Kumar and Torr, 2008; Veksler, 1999). Specifically, the following known results are corollaries of the above theorem. Note that, while the following corollaries have been previously proved in the literature, our work is the first to establish the tightness of the theoretical guarantees.

Corollary 4 *When $q = 1$, the multiplicative bound of the interval move-making algorithm (which is equivalent to the expansion algorithm) for the uniform metric distance is 2.*

The above corollary follows from the approximation factor of the interval rounding procedure proved by Kleinberg and Tardos (1999), but it was independently proved by Veksler (1999).

Corollary 5 *When $q = M$, the multiplicative bound of the interval move-making algorithm for the truncated linear distance function is 4.*

The above corollary follows from the approximation factor of the interval rounding procedure proved by Chekuri et al. (2001), but it was independently proved by Gupta and Tardos (2000).

Corollary 6 *When $q = \sqrt{2}M$, the multiplicative bound of the interval move-making algorithm for the truncated linear distance function is $2 + \sqrt{2}$.*

The above corollary follows from the approximation factor of the interval rounding procedure proved by Chekuri et al. (2001), but it was independently proved by Kumar and Torr (2008). Finally, since our analysis does not use the triangular inequality of metric distance functions, we can also state the following corollary for the truncated quadratic distance.

Corollary 7 *When $q = \sqrt{M}$, the multiplicative bound of the interval move-making algorithm for the truncated quadratic distance function is $O(\sqrt{M})$.*

The above corollary follows from the approximation factor of the interval rounding procedure proved by Chekuri et al. (2001), but it was independently proved by Kumar and Torr (2008).

An interval move-making algorithm that uses an interval length of q runs for at most $O(h/q)$ iterations. This follows from a simple modification of the result by Gupta and Tardos (2000) (specifically, theorem 3.7). Hence, the total time complexity of interval move-making is $O(nmhq^2 \log(m))$, since each iteration solves a minimum st -cut problem of a graph with $O(nq)$ nodes and $O(mq^2)$ arcs. In other words, interval move-making is at most as computationally complex as complete move-making, which in turn is significantly less complex than solving the LP relaxation. Note that problem (5), which is required for interval move-making, is independent of the unary potentials and the edge weights. Hence, it only needs to be solved once beforehand for all pairs of labels $(\hat{x}_a, \hat{x}_b) \in \mathbf{L} \times \mathbf{L}$ in order to obtain a solution for any semi-metric labeling problem defined using the distance function d .

7. Hierarchical Rounding and Hierarchical Moves

We now consider the most general form of parallel rounding that has been proposed in the literature, namely the hierarchical rounding procedure (Kleinberg and Tardos, 1999). The rounding relies on a hierarchical clustering of the labels. Formally, we denote a hierarchical clustering of m levels for the label set \mathbf{L} by $\mathbf{C} = \{\mathbf{C}(i), i = 1, \dots, m\}$. At each level i , the clustering $\mathbf{C}(i) = \{\mathbf{C}(i, j) \subseteq \mathbf{L}, j = 1, \dots, h^i\}$ is mutually exclusive and collectively exhaustive, that is,

$$\bigcup_j \mathbf{C}(i, j) = \mathbf{L}, \mathbf{C}(i, j) \cap \mathbf{C}(i, j') = \emptyset, \forall j \neq j'.$$

Furthermore, for each cluster $\mathbf{C}(i, j)$ at the level $i > 2$, there exists a unique cluster $\mathbf{C}(i-1, j')$ in the level $i-1$ such that $\mathbf{C}(i, j) \subseteq \mathbf{C}(i-1, j')$. We call the cluster $\mathbf{C}(i-1, j')$ the parent of the

Algorithm 5 The hierarchical rounding procedure.

input A feasible solution \mathbf{y} of the LP relaxation.

- 1: Define $f_a^1 = 1$ for all $X_a \in \mathbf{X}$.
- 2: **for all** $i \in \{2, \dots, m\}$ **do**
- 3: **for all** $X_a \in \mathbf{X}$ **do**
- 4: Define $z_a^i(j)$ for all $j \in \{1, \dots, h^i\}$ as follows:

$$z_a^i(j) = \begin{cases} \sum_{k \text{ s.t. } l_k \in \mathbf{C}(i,j)} y_a(k) & \text{if } p(i, j) = f_a^{i-1}, \\ 0 & \text{otherwise.} \end{cases}$$

- 5: Define $y_a^i(j)$ for all $j \in \{1, \dots, h^i\}$ as follows:

$$y_a^i(j) = \frac{z_a^i(j)}{\sum_{j'=1}^{h^i} z_a^i(j')}$$

- 6: **end for**
 - 7: Using a rounding procedure (complete or interval) on $\mathbf{y}^i = [y_a^i(j), \forall X_a \in \mathbf{X}, j \in \{1, \dots, h^i\}]$, obtain an integer solution $\hat{\mathbf{y}}^i$.
 - 8: **for all** $X_a \in \mathbf{X}$ **do**
 - 9: Let $k_a \in \{1, \dots, h^i\}$ such that $\hat{y}^i(k_a) = 1$. Define $f_a^i = k_a$.
 - 10: **end for**
 - 11: **end for**
 - 12: **for all** $X_a \in \mathbf{X}$ **do**
 - 13: Let l_k be the unique label present in the cluster $\mathbf{C}(m, f_a^m)$. Assign l_k to X_a .
 - 14: **end for**
-

cluster $\mathbf{C}(i, j)$ and define $p(i, j) = j'$. Similarly, we call $\mathbf{C}(i, j)$ a child of $\mathbf{C}(i-1, j')$. Without loss of generality, we assume that there exists a single cluster at level 1 that contains all the labels, and that each cluster at level m contains a single label.

Algorithm 5 describes the hierarchical rounding procedure. Given a clustering \mathbf{C} , it proceeds in a top-down fashion through the hierarchy while assigning each random variable to a cluster in the current level. Let f_a^i be the index of the cluster assigned to the random variable X_a in the level i . In the first step, the rounding procedure assigns all the random variables to the unique cluster $\mathbf{C}(1, 1)$ (step 1). At each step i , it assigns each random variable to a unique cluster in the level i by computing a conditional probability distribution as follows. The conditional probability $y_a^i(j)$ of assigning the random variable X_a to the cluster $\mathbf{C}(i, j)$ is proportional to $\sum_{l_k \in \mathbf{C}(i, j)} y_a(l_k)$ if $p(i, j) = f_a^{i-1}$ (steps 3-6). The conditional probability $y_a^i(j) = 0$ if $p(i, j) \neq f_a^{i-1}$, that is, a random variable cannot be assigned to a cluster $\mathbf{C}(i, j)$ if it wasn't assigned to its parent in the previous step. Using a rounding procedure (complete or interval) for \mathbf{y}^i , we obtain an assignment of random variables to the clusters at level i (step 7). Once such an assignment is obtained, the values f_a^i are computed for all random variables X_a (steps 8-10). At the end of step m , hierarchical rounding would have assigned each random variable to a unique cluster in the level m . Since each cluster at level m consists of a single label, this provides us with a labeling of the MRF (steps 12-14).

Algorithm 6 The hierarchical move-making algorithm.

input Unary potentials $\theta_a(\cdot)$, edge weights w_{ab} , distance function d .

- 1: **for all** $j \in \{1, \dots, h\}$ **do**
 - 2: Let l_k be the unique label in the cluster $\mathbf{C}(m, j)$. Define $x_a^{m,j} = l_k$ for all $X_a \in \mathbf{X}$.
 - 3: **end for**
 - 4: **for all** $i \in \{2, \dots, m\}$ **do**
 - 5: **for all** $j \in \{1, \dots, h^{m-i+1}\}$ **do**
 - 6: Define $\mathbf{L}_a^{m-i+1,j} = \{x_a^{m-i+2,j'}, p(m-i+2, j') = j, j' \in \{1, \dots, h^{m-i+2}\}\}$.
 - 7: Using a move-making algorithm (complete or interval), compute the labeling $\mathbf{x}^{m-i+1,j}$ under the constraint $x_a^{m-i+1,j} \in \mathbf{L}_a^{m-i+1,j}$.
 - 8: **end for**
 - 9: **end for**
 - 10: The final solution is $\mathbf{x}^{1,1}$.
-

Our goal is to design a move-making algorithm whose multiplicative bound matches the approximation factor of the hierarchical rounding procedure for any choice of hierarchical clustering \mathbf{C} . To this end, we propose the hierarchical move-making algorithm, which extends the hierarchical graph cuts approach for hierarchically well-separated tree (HST) metrics proposed by Kumar and Koller (2009). Algorithm 6 provides its main steps. In contrast to hierarchical rounding, the move-making algorithm traverses the hierarchy in a bottom-up fashion while computing a labeling for each cluster in the current level. Let $\mathbf{x}^{i,j}$ be the labeling corresponding to the cluster $\mathbf{C}(i, j)$. At the first step, when considering the level m of the clustering, all the random variables are assigned the same label. Specifically, $x_a^{m,j}$ is equal to the unique label contained in the cluster $\mathbf{C}(m, j)$ (steps 1-3). At step i , it computes the labeling $\mathbf{x}^{m-i+1,j}$ for each cluster $\mathbf{C}(m-i+1, j)$ by using the labelings computed in the previous step. Specifically, it restricts the label assigned to a random variable X_a in the labeling $\mathbf{x}^{m-i+1,j}$ to the subset of labels that were assigned to it by the labelings corresponding to the children of $\mathbf{C}(m-i+1, j)$ (step 6). Under this restriction, the labeling $\mathbf{x}^{m-i+1,j}$ is computed

by approximately minimizing the energy using a move-making algorithm (step 7). Implicit in our description is the assumption that we will use a move-making algorithm (complete or interval) in step 7 of Algorithm 6 whose multiplicative bound matches the approximation factor of the rounding procedure (complete or interval) used in step 7 of Algorithm 5. Note that, unlike the hierarchical graph cuts approach (Kumar and Koller, 2009), the hierarchical move-making algorithm can be used for any arbitrary clustering and not just the one specified by an HST metric.

The following theorem establishes the theoretical guarantees of the hierarchical move-making algorithm and the hierarchical rounding procedure.

Theorem 8 *The tight multiplicative bound of the hierarchical move-making algorithm is equal to the tight approximation factor of the hierarchical rounding procedure.*

The proof of the above theorem is given in Appendix C. The following known result is its corollary.

Corollary 9 *The multiplicative bound of the hierarchical move-making algorithm is $O(1)$ for an HST metric distance.*

The above corollary follows from the approximation factor of the hierarchical rounding procedure proved by Kleinberg and Tardos (1999), but it was independently proved by Kumar and Koller (2009). It is worth noting that the above result was also used to obtain an approximation factor of $O(\log h)$ for the general metric labeling problem by Kleinberg and Tardos (1999) and a matching multiplicative bound of $O(\log h)$ by Kumar and Koller (2009).

The time complexity of the hierarchical move-making algorithm depends on two factors: (i) the hierarchy, which defines the subproblems; and (ii) whether we use a complete or an interval move-making algorithm to solve the subproblems. In what follows, we analyze its worst case time complexity over all possible hierarchies. Clearly, since complete move-making is more expensive than interval move-making, the worst case complexity of hierarchical move-making will be achieved when we use complete move-making to solve each subproblem. We begin by noting that each problem is defined using a smaller label set. For simplicity, let us assume that the clusters of the hierarchy are balanced, and that each subproblem is solved over $h' \leq h$ labels. It follows that the total number of subproblems that we would be required to solve is $O((h/h')^{(\log(h)/\log(h'))-1})$. Next, observe that the complexity of the complete move-making algorithms is cubic in the number of labels. Thus, it follows that in order to maximize the total time complexity of hierarchical move-making, we need to set $h' = h$. In other words, hierarchical move-making algorithm is at most as computationally complex as the complete move-making algorithm. Recall that the complete move-making complexity is $O(nmh^3 \log(m))$. Hence, hierarchical move-making is significantly faster than solving the LP relaxation.

8. Experiments

We demonstrate the efficacy of rounding-based moves by comparing them to several state of the art methods using both synthetic and real data.

8.1 Synthetic Experiments

8.1.1 DATA

We generated two synthetic data sets to conduct our experiments. The first data set consists of random grid MRFs of size 100×100 , where each random variable can take one of 10 labels. The unary

potentials were sampled from a uniform distribution over $[0, 10]$. The edge weights were sampled from a uniform distribution over $[0, 3]$. We considered four types of pairwise potentials: (i) truncated linear metric, where the truncation is sampled from a uniform distribution over $[1, 5]$; (ii) truncated quadratic semi-metric, where the truncation is sampled from a uniform distribution over $[1, 25]$; (iii) random metrics, generated by computing the shortest path on graphs whose vertices correspond to the labels and whose edge lengths are uniformly distributed over $[1, 10]$; (iv) random semi-metrics, where the distance between two labels is sampled from a uniform distribution over $[1, 10]$. For each type of pairwise potentials, we generated 500 different MRFs.

The second data set is similar to the first one, except that it is defined on a smaller grid of size 20×20 . The smaller grid size allows us to test the mixture-of-tree algorithm (Andrew et al., 2011) that is closely related to the serial rounding procedure of Archer et al. (2004). We generated 5 different grids by sampling their edge weights from a uniform distribution over $[5, 100]$. For each set of edge weights, we use the cut-based decomposition method of Racke (2008) to approximate the original grid graph using a mixture of tree-structured graphs. We generated three types of unary potentials by uniformly sampling from the range $[0, 100]$, $[100, 1000]$ and $[1000, 10000]$ respectively. Similar to the first data set, we considered four types of pairwise potentials. For each pair of unary potential type and pairwise potential type, we generated 100 different MRFs. This provided us with 1200 MRFs for each set of edge weights, and a total of 6000 MRFs to perform our experiments.

8.1.2 METHODS

We report results obtained by the following state of the art methods using the first data set: (i) belief propagation (BP) (Pearl, 1998); (ii) sequential tree-reweighted message passing (TRW) (Kolmogorov, 2006), which optimizes the dual of the LP relaxation, and provides comparable results to other LP relaxation based approaches; (iii) expansion algorithm (EXP) (Boykov et al., 1999); and (iv) swap algorithm (SWAP) (Boykov et al., 1998). We compare the above methods to a hierarchy move-making algorithm (HIER), where a set of hierarchies is obtained by approximating a given semi-metric as a mixture of r -HST metrics using the method proposed by Fakcharoenphol et al. (2003). We refer the reader to (Fakcharoenphol et al., 2003; Kumar and Koller, 2009) for details. Each subproblem of the hierarchical move-making algorithm is solved by interval move-making with interval length $q = 1$ (which corresponds to the expansion algorithm). In addition, for the truncated linear and truncated quadratic cases, we present results of interval move-making (INT) using the optimal interval length reported by Kumar and Torr (2008).

For the second data set, we report the results for all the above methods, as well as the mixture-of-tree (MOT) algorithm. The smaller size of the grid in the second data set allows us to obtain a mixture of tree-structured MRFs using the method of Racke (2008), which was not possible for the larger 100×100 grid used in the first data set.

8.1.3 RESULTS

Figure 1 shows the results for the first data set. In terms of the energy, TRW is the most accurate. However, it is slow as it optimizes the dual of the LP relaxation. The labelings obtained by BP have high energy values. The standard move-making algorithms, EXP and SWAP, are fast due to the use of efficient minimum st -cut solvers. However, they are not as accurate as TRW. For the truncated linear and quadratic pairwise potentials, INT provides labelings with comparable energy to those of TRW, and is also computationally efficient. However, for general metrics and semi-metrics,

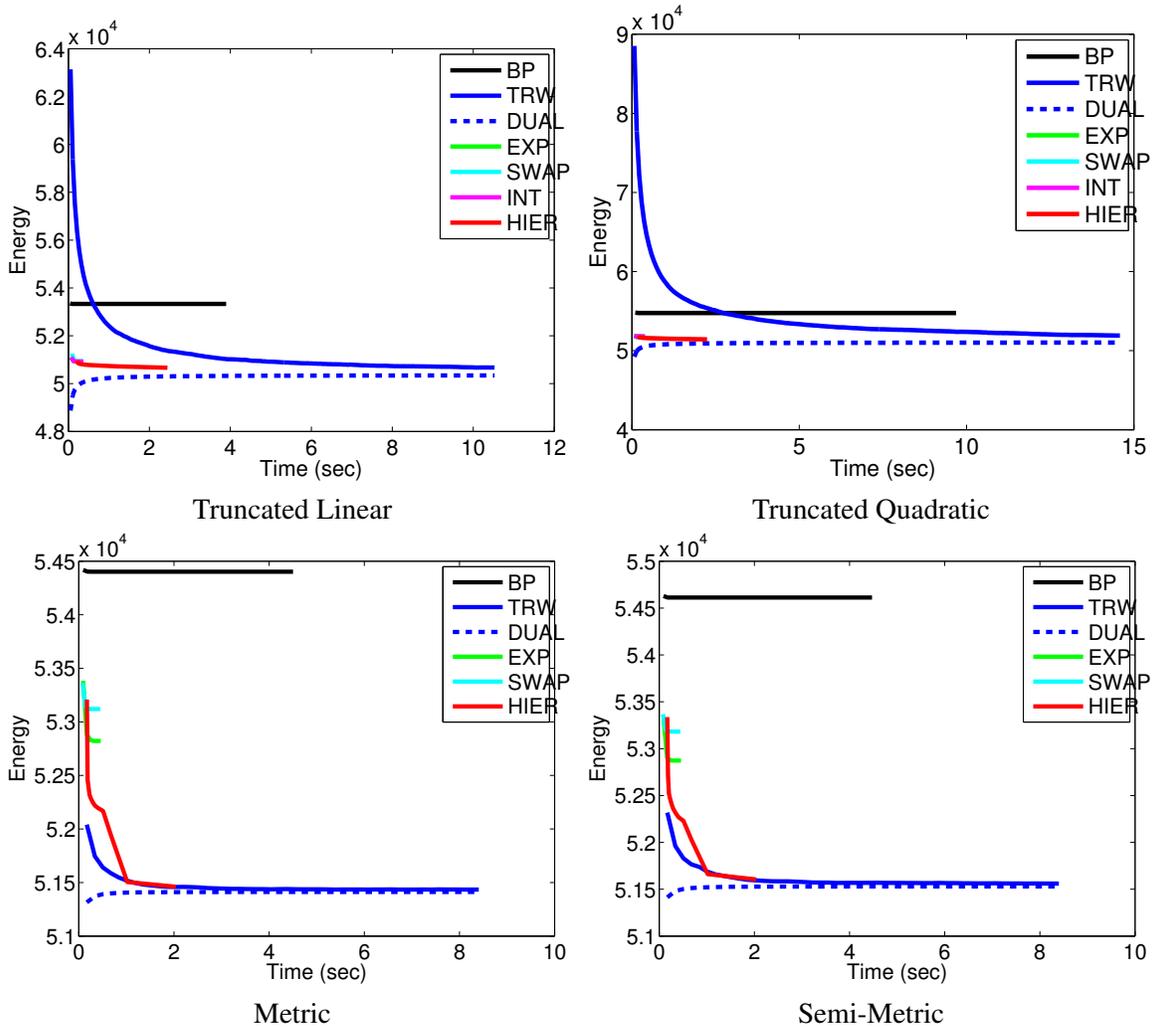


Figure 1: Results for the first synthetic data set. The x-axis shows the time in seconds, while the y-axis shows the energy value. The dashed line shows the value of the dual of the LP obtained by TRW. Best viewed in color.

it is not obvious how to obtain the optimal interval length. The HIER method is more generally applicable as there exist standard methods to approximate a semi-metric with a mixture of r-HST metrics (Fakcharoenphol et al., 2003). It provides very accurate labelings (comparable to TRW), and is efficient in practice as it relies on solving each subproblem using an iterative move-making algorithm.

Figure 2 shows the results for the second data set. The results of all the methods are similar to the first data set. However, the additional MOT algorithm performs very poorly compared to all other methods. This may be explained by the fact that, while MOT provides the same worst-case guarantees as serial rounding over all possible metric distance functions, it does not match the accuracy of serial rounding for a given distance function.

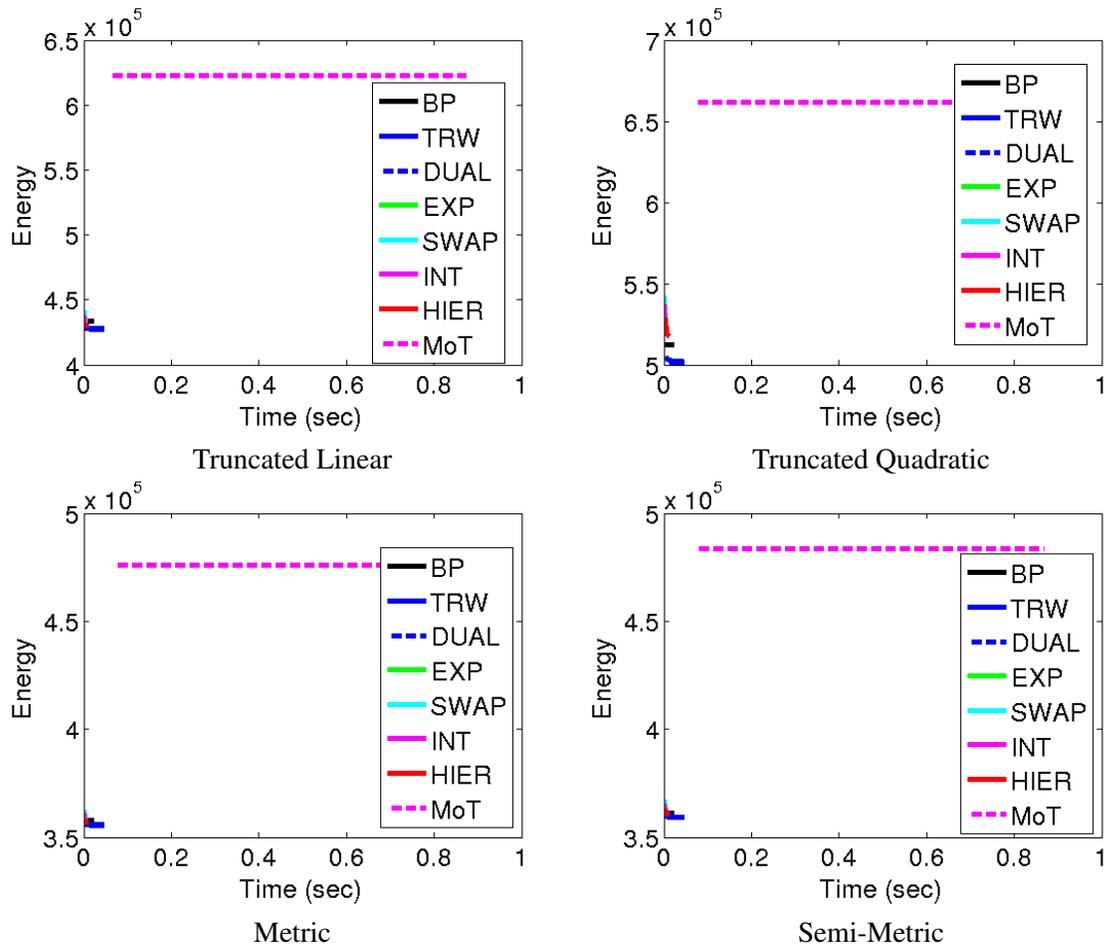


Figure 2: Results for the second synthetic data set. The x-axis shows the time in seconds, while the y-axis shows the energy value. The dashed line shows the value of the dual of the LP obtained by TRW. Best viewed in color.

8.2 Dense Stereo Correspondence

8.2.1 DATA

Given two epipolar rectified images of the same scene, the problem of dense stereo correspondence requires us to obtain a correspondence between the pixels of the images. This problem can be modeled as semi-metric labeling, where the random variables represent the pixels of one of the images, and the labels represent the disparity values. A disparity label l_i for a random variable X_a representing a pixel (u_a, v_a) of an image indicates that its corresponding pixel lies in location $(u_a + i, v_a)$. For the above problem, we use the unary potentials and edge weights that are specified by Szeliski et al. (2008). We use two types of pairwise potentials: (i) truncated linear with the truncation set at 4; and (ii) truncated quadratic with the truncation set at 16.

8.2.2 METHODS

We report results on all the baseline methods that were used in the synthetic experiments, namely, BP, TRW, EXP, and SWAP. Since the pairwise potentials are either truncated linear or truncated quadratic, we report results for the interval move-making algorithm INT, which uses the optimal value of the interval length. We also show the results obtained by the hierarchical move-making algorithm (HIER), where once again the hierarchies are obtained by approximating the semi-metric as a mixture of r-HST metrics.

8.2.3 RESULTS

Figure 3-Figure 8 shows the results for various standard pairs of images. Note that, similar to the synthetic experiments, TRW is the most accurate in terms of energy, but it is computationally inefficient. The results obtained by BP are not accurate. The standard move-making algorithms, EXP and SWAP, are fast but not as accurate as TRW. Among the rounding-based move-making algorithms INT is slower as it solves a minimum st -cut problem on a large graph at each iteration. In contrast, HIER uses an interval length of 1 for each subproblem and is therefore more efficient. The energy obtained by HIER is comparable to TRW.

9. Discussion

For any general distance function that can be used to specify the semi-metric labeling problem, we proved that the approximation factor of a large family of parallel rounding procedures is matched by the multiplicative bound of move-making algorithms. This generalizes previously known results on the guarantees of move-making algorithms in two ways: (i) in contrast to previous results (Kumar and Koller, 2009; Kumar and Torr, 2008; Veksler, 1999) that focused on special cases of distance functions, our results are applicable to arbitrary semi-metric distance functions; and (ii) the guarantees provided by our theorems are tight. Our experiments confirm that the rounding-based move-making algorithms provide similar accuracy to the LP relaxation, while being significantly faster due to the use of efficient minimum st -cut solvers.

Several natural questions arise. What is the exact characterization of the rounding procedures for which it is possible to design matching move-making algorithms? Can we design rounding-based move-making algorithms for other combinatorial optimization problems? Answering these questions

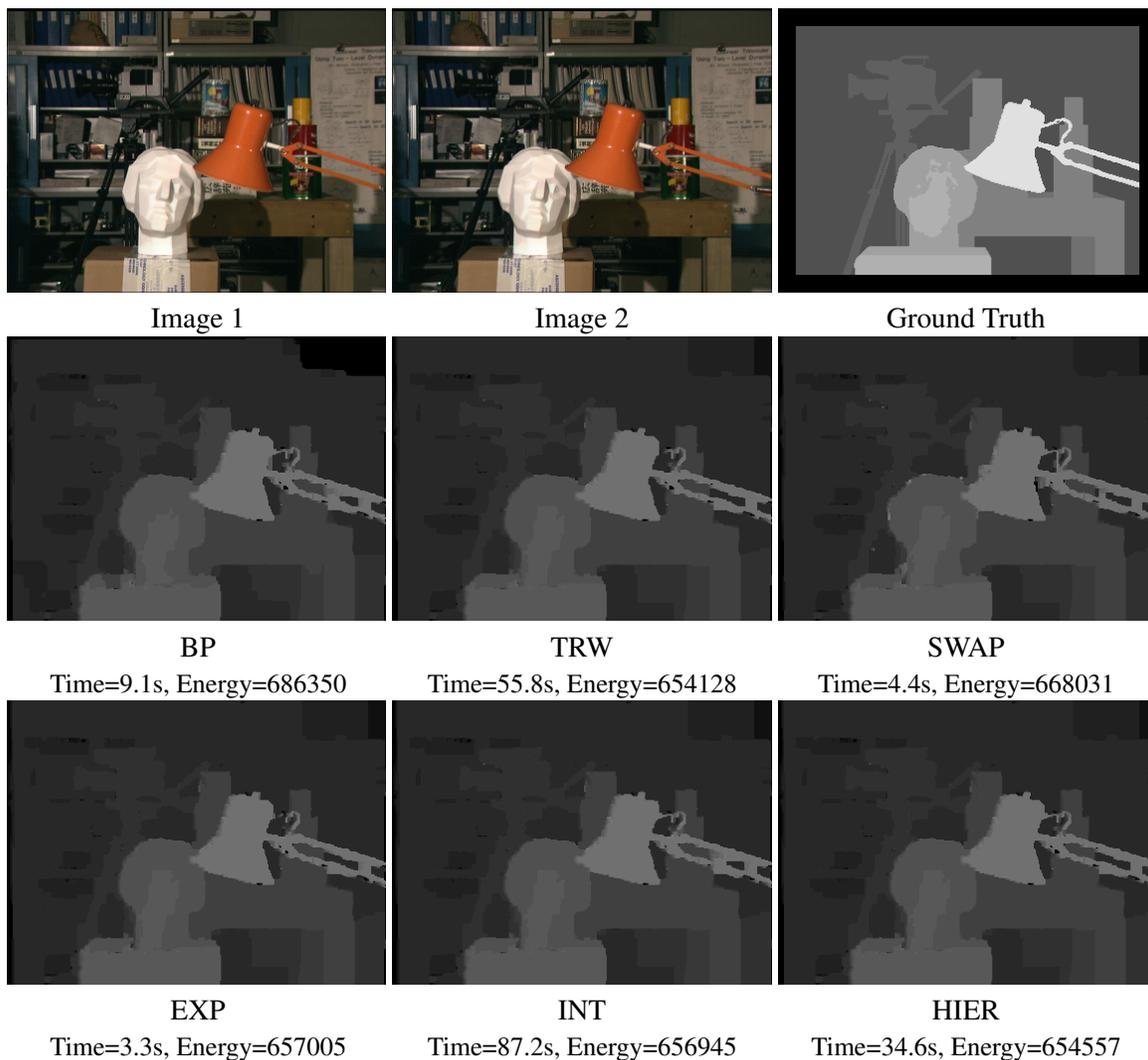


Figure 3: Results for the ‘tsukuba’ image pair with truncated linear pairwise potentials.

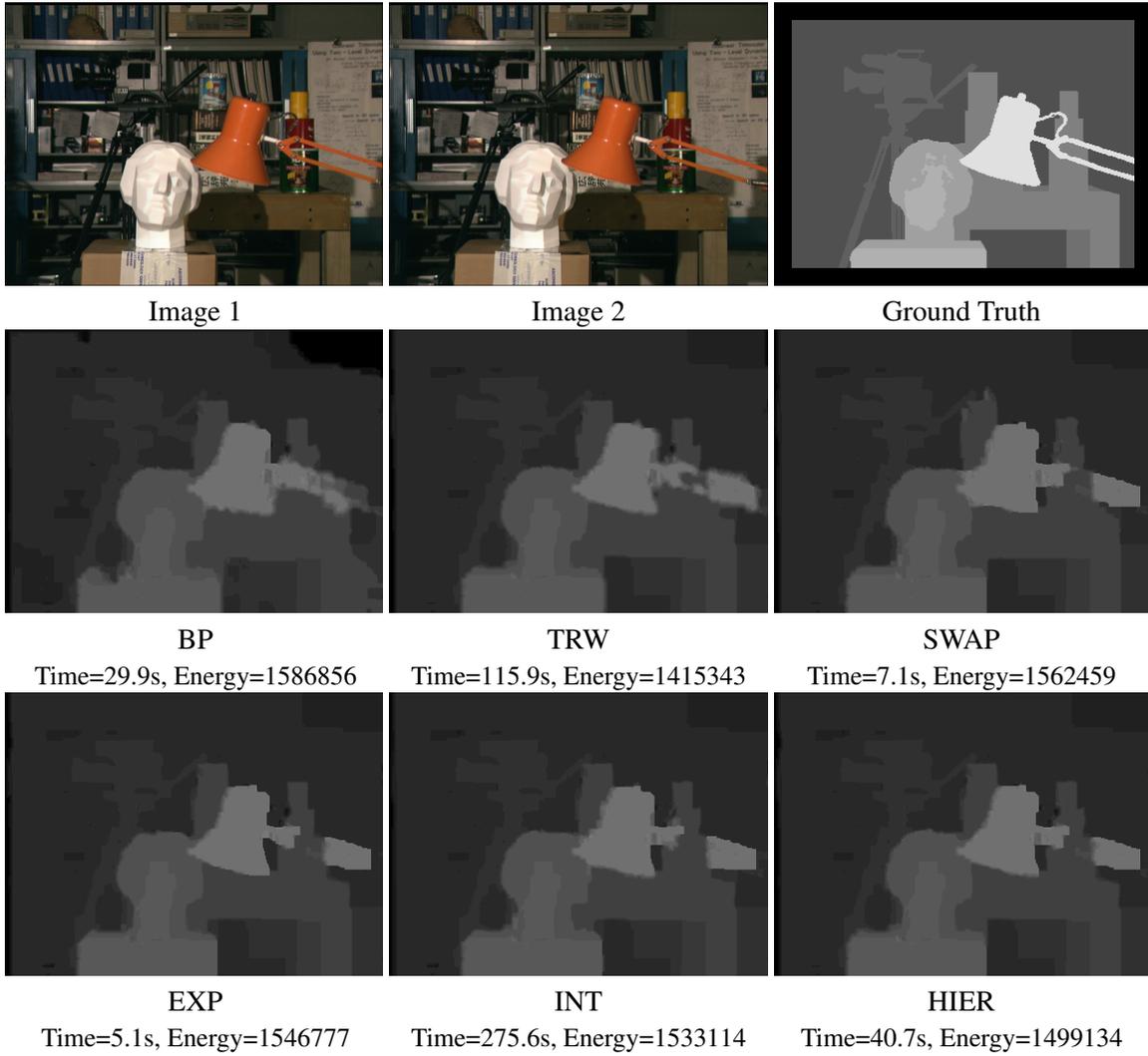


Figure 4: Results for the ‘tsukuba’ image pair with truncated quadratic pairwise potentials.

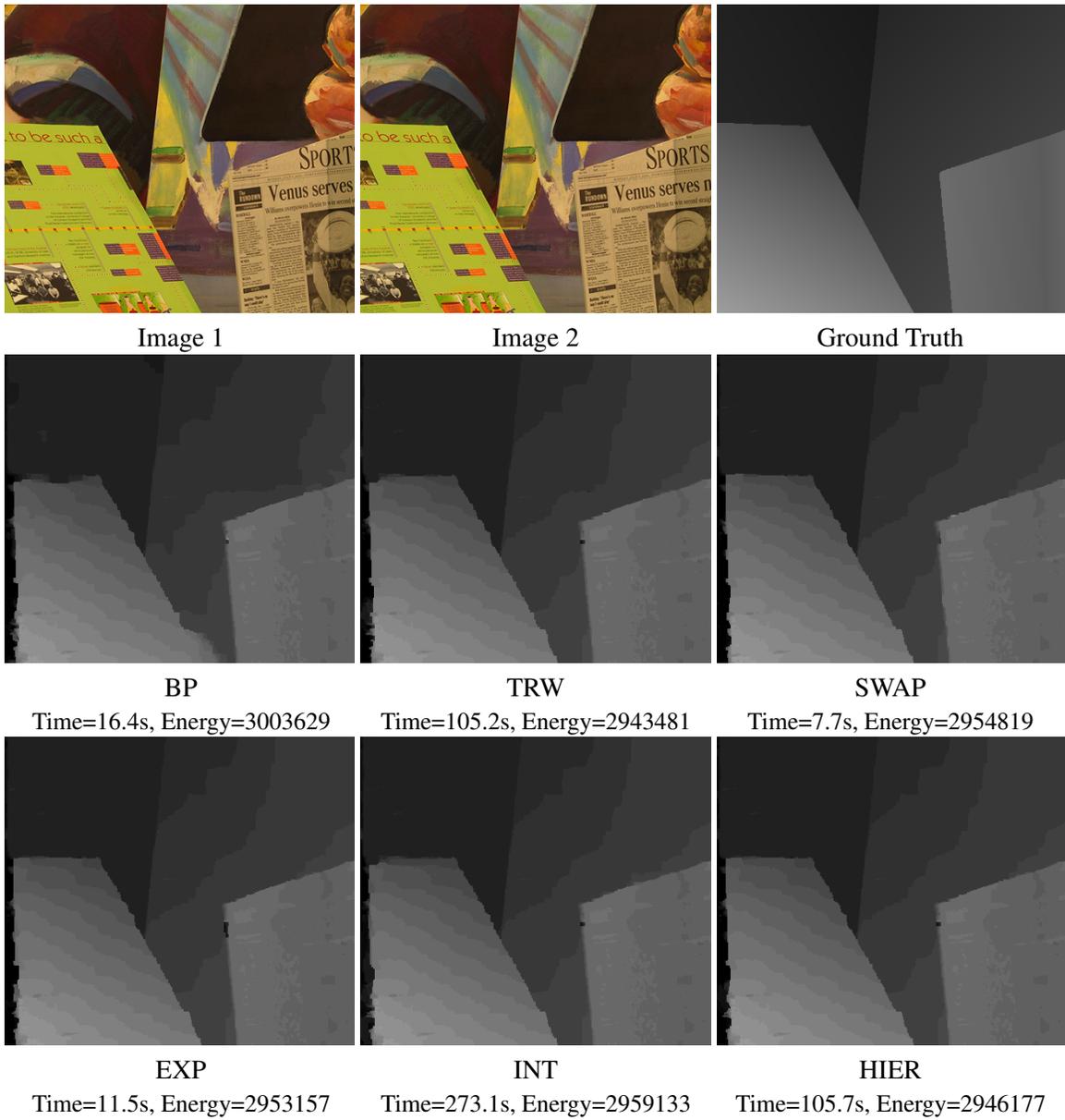


Figure 5: Results for the ‘venus’ image pair with truncated linear pairwise potentials.

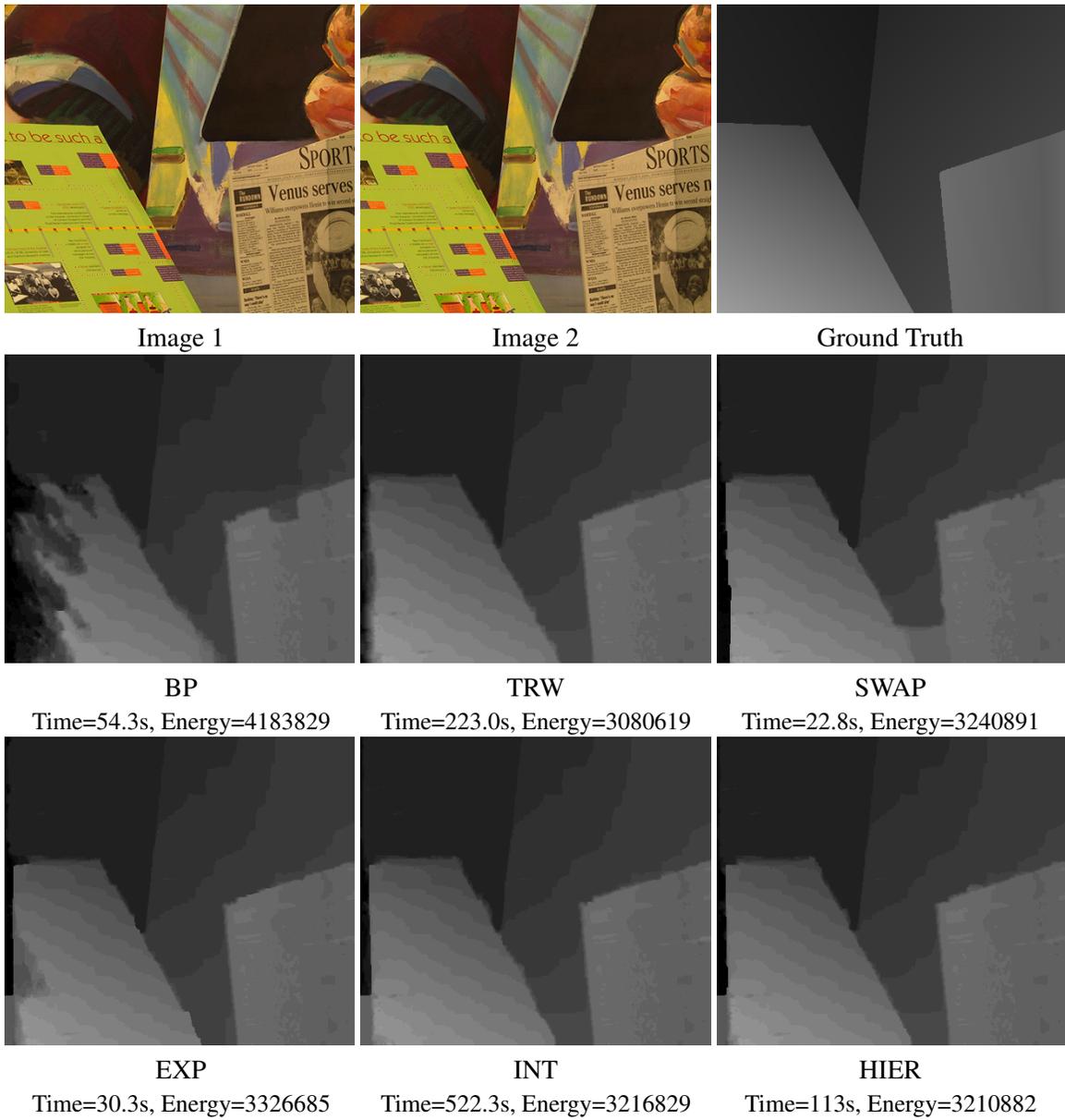


Figure 6: Results for the ‘venus’ image pair with truncated quadratic pairwise potentials.

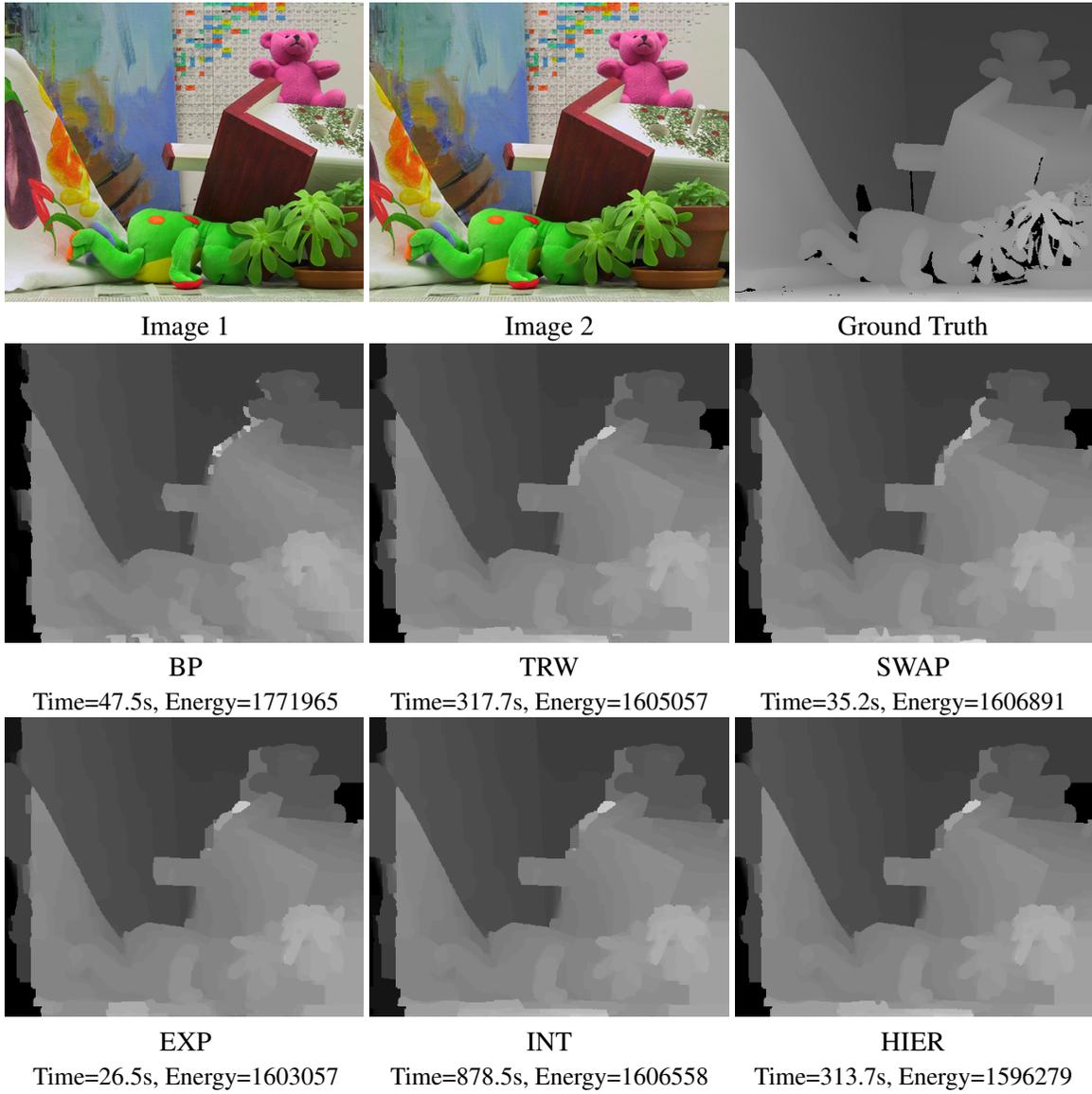


Figure 7: Results for the ‘teddy’ image pair with truncated linear pairwise potentials.

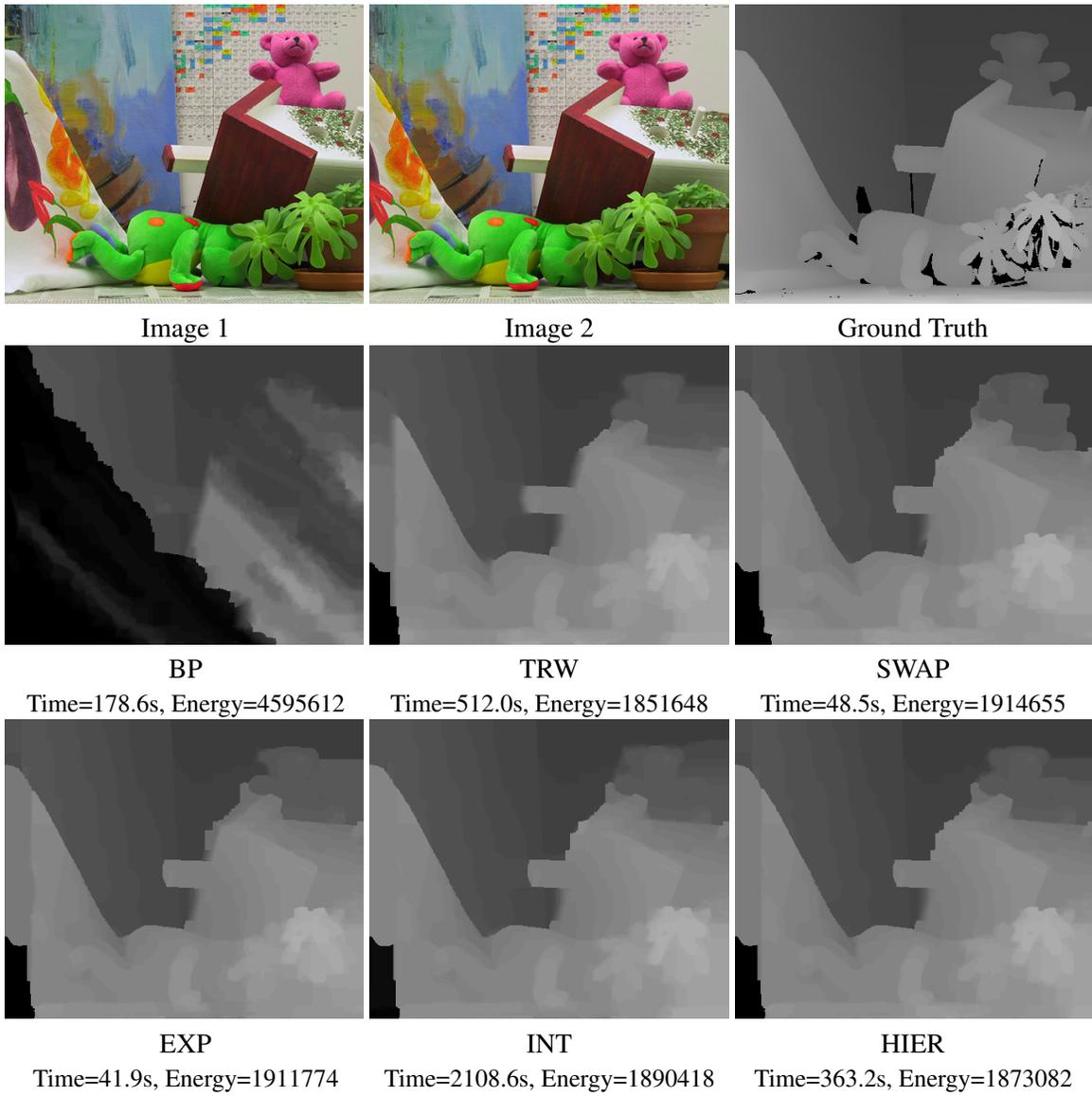


Figure 8: Results for the ‘teddy’ image pair with truncated quadratic pairwise potentials.

will not only expand our theoretical understanding, but also result in the development of efficient and accurate algorithms.

Acknowledgments

This work is partially funded by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC Grant agreement number 259112.

Appendix A. Proof of Theorem 1

We first establish the theoretical property of the complete move-making algorithm using the following lemma.

Lemma 10 *The tight multiplicative bound of the complete move-making algorithm is equal to the submodular distortion of the distance function.*

Proof The submodular distortion of a distance function d is obtained by computing its tightest submodular overestimation as follows:

$$\begin{aligned} \bar{d} = & \operatorname{argmin}_{d'} t & (7) \\ \text{s.t. } & d'(l_i, l_j) \leq td(l_i, l_j), \forall l_i, l_j \in \mathbf{L}, \\ & d'(l_i, l_j) \geq d(l_i, l_j), \forall l_i, l_j \in \mathbf{L}, \\ & d'(l_i, l_j) + d'(l_{i+1}, l_{j+1}) \leq d'(l_i, l_{j+1}) + d'(l_{i+1}, l_j), \forall l_i, l_j \in \mathbf{L} \setminus \{l_h\}. \end{aligned}$$

In order to prove the theorem, it is important to note that the definition of submodular distance function implies the following:

$$\bar{d}(l_i, l_j) + \bar{d}(l_{i'}, l_{j'}) \leq \bar{d}(l_i, l_{j'}) + \bar{d}(l_{i'}, l_j), \forall i' > i, j' > j.$$

A simple proof for the above claim can be found in (Flach and Schlesinger, 2006).

We denote the submodular distortion of d by B . By definition, it follows that

$$d(l_i, l_j) \leq \bar{d}(l_i, l_j) \leq Bd(l_i, l_j), \forall l_i, l_j \in \mathbf{L}. \quad (8)$$

We denote an optimal labeling of the original semi-metric labeling problem as \mathbf{x}^* , that is,

$$\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \mathbf{L}^n} \sum_{X_a \in \mathbf{X}} \theta_a(x_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab}d(x_a, x_b). \quad (9)$$

As the semi-metric labeling problem is NP-hard, an optimal labeling \mathbf{x}^* cannot be computed efficiently using any known algorithm. In order to obtain an approximate solution $\hat{\mathbf{x}}$, the complete move-making algorithm replaces the original distance function d by its submodular overestimation \bar{d} , that is,

$$\hat{\mathbf{x}} = \operatorname{argmin}_{\mathbf{x} \in \mathbf{L}^n} \sum_{X_a \in \mathbf{X}} \theta_a(x_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab}\bar{d}(x_a, x_b). \quad (10)$$

Since the pairwise potentials in the above problem are submodular, the approximate solution $\hat{\mathbf{x}}$ can be obtained by solving a single minimum *st*-cut problem using the method of Flach and Schlesinger (2006). Using inequality (8), it follows that

$$\begin{aligned}
 & \sum_{X_a \in \mathbf{X}} \theta_a(\hat{x}_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} d(\hat{x}_a, \hat{x}_b) \\
 \leq & \sum_{X_a \in \mathbf{X}} \theta_a(\hat{x}_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} \bar{d}(\hat{x}_a, \hat{x}_b) \\
 \leq & \sum_{X_a \in \mathbf{X}} \theta_a(x_a^*) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} \bar{d}(x_a^*, x_b^*) \\
 \leq & \sum_{X_a \in \mathbf{X}} \theta_a(x_a^*) + B \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} d(x_a^*, x_b^*).
 \end{aligned}$$

The above inequality proves that the multiplicative bound of the complete move-making algorithm is at most B . In order to prove that it is exactly equal to B , we need to construct an example for which the bound is tight. To this end, let l_k and $l_{k'}$ be two labels in the set \mathbf{L} such that $k < k'$ and

$$\frac{\bar{d}(l_k, l_{k'})}{d(l_k, l_{k'})} = B.$$

Since B is the minimum possible value of the maximum ratio of the estimated distance \bar{d} to the original distance d , such a pair of labels must exist (otherwise, the submodular distortion can be reduced further). Let us assume that there exists an $l_j \in \mathbf{L}$ such that $j < k$. Other cases (where $j > k'$ or $k < j < k'$) can be handled similarly. Note that since \bar{d} is submodular, it follows that

$$\bar{d}(l_k, l_j) + \bar{d}(l_j, l_{k'}) \geq \bar{d}(l_k, l_{k'}). \quad (11)$$

We define a semi-metric labeling problem over two random variables X_a and X_b connected by an edge with weight $w_{ab} = 1$. The unary potentials are defined as follows:

$$\theta_a(i) = \begin{cases} 0, & \text{if } i = k, \\ \frac{\bar{d}(l_k, l_{k'}) + \bar{d}(l_k, l_j) - \bar{d}(l_j, l_{k'})}{2}, & \text{if } i = j, \\ \infty & \text{otherwise,} \end{cases}$$

$$\theta_b(i) = \begin{cases} 0, & \text{if } i = k', \\ \frac{\bar{d}(l_k, l_{k'}) - \bar{d}(l_k, l_j) + \bar{d}(l_j, l_{k'})}{2}, & \text{if } i = j, \\ \infty & \text{otherwise.} \end{cases}$$

For the above semi-metric labeling problem, it can be verified that an optimal solution \mathbf{x}^* of problem (9) is the following: $x_a^* = l_k$ and $x_b^* = l_{k'}$. Furthermore, using inequality (11), it can be shown that the following is an optimal solution of problem (10): $\hat{x}_a = l_j$ and $\hat{x}_b = l_j$. In other words, $\hat{\mathbf{x}}$ is a valid approximate labeling provided by the complete move-making algorithm. The labelings \mathbf{x}^* and $\hat{\mathbf{x}}$ satisfy the following equality:

$$\sum_{X_a \in \mathbf{X}} \theta_a(\hat{x}_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} d(\hat{x}_a, \hat{x}_b) = \sum_{X_a \in \mathbf{X}} \theta_a(x_a^*) + B \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} d(x_a^*, x_b^*).$$

Therefore, the tight multiplicative bound of the complete move-making algorithm is exactly equal to the submodular distortion of the distance function d . \blacksquare

We now turn our attention to the complete rounding procedure for the LP relaxation. Before we can establish its tight approximation factor, we need to compute the expected distance between the labels assigned to a pair of neighboring random variables. Recall that, in our notation, we denote a feasible solution of the LP relaxation by \mathbf{y} . For any feasible solution \mathbf{y} , we define \mathbf{y}_a as the vector whose elements are the unary variables of \mathbf{y} for the random variable $X_a \in \mathbf{X}$, that is,

$$\mathbf{y}_a = [y_a(i), \forall l_i \in \mathbf{L}]. \quad (12)$$

Similarly, we define \mathbf{y}_{ab} as the vector whose elements are the pairwise variables of \mathbf{y} for the neighboring random variables $(X_a, X_b) \in \mathbf{E}$, that is,

$$\mathbf{y}_{ab} = [y_{ab}(i, j), \forall l_i, l_j \in \mathbf{L}]. \quad (13)$$

Furthermore, using \mathbf{y}_a , we define \mathbf{Y}_a as

$$Y_a(i) = \sum_{j=1}^i y_a(j).$$

In other words, if \mathbf{y}_a is interpreted as the probability distribution over the labels of X_a , then \mathbf{Y}_a is the corresponding cumulative distribution.

Given a feasible solution \mathbf{y} , we denote the integer solution obtained using the complete rounding procedure as $\hat{\mathbf{y}}$. The distance between the two labels encoded by vectors $\hat{\mathbf{y}}_a$ and $\hat{\mathbf{y}}_b$ will be denoted by $\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b)$. In other words, if \hat{f}_a and \hat{f}_b are the indices of the labels assigned to X_a and X_b (that is, $\hat{y}_a(\hat{f}_a) = 1$ and $\hat{y}_b(\hat{f}_b) = 1$), then $\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b) = d(l_{\hat{f}_a}, l_{\hat{f}_b})$.

The following shorthand notation would be useful for our analysis.

$$\begin{aligned} D_1(i) &= \frac{1}{2} (d(l_i, l_1) + d(l_i, l_h) - d(l_{i+1}, l_1) - d(l_{i+1}, l_h)), \forall i \in \{1, \dots, h-1\}, \\ D_2(i, j) &= \frac{1}{2} (d(l_i, l_{j+1}) + d(l_{i+1}, l_j) - d(l_i, l_j) - d(l_{i+1}, l_{j+1})), \forall i, j \in \{1, \dots, h-1\}. \end{aligned} \quad (14)$$

Using the above notation, we can state the following lemma on the expected distance of the rounded solution for two neighboring random variables.

Lemma 11 *Let \mathbf{y} be a feasible solution of the LP relaxation, \mathbf{Y}_a and \mathbf{Y}_b be cumulative distributions of \mathbf{y}_a and \mathbf{y}_b , and $\hat{\mathbf{y}}$ be the integer solution obtained by the complete rounding procedure for \mathbf{y} . Then, the following equation holds true:*

$$\mathbb{E}(\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b)) = \sum_{i=1}^{h-1} Y_a(i) D_1(i) + \sum_{j=1}^{h-1} Y_b(j) D_1(j) + \sum_{i=1}^{h-1} \sum_{j=1}^{h-1} |Y_a(i) - Y_b(j)| D_2(i, j).$$

Proof We define \hat{f}_a and \hat{f}_b to be the indices of the labels assigned to X_a and X_b by the rounded integer solution $\hat{\mathbf{y}}$. In other words, $\hat{y}_a(i) = 1$ if and only if $i = \hat{f}_a$ and $\hat{y}_b(j) = 1$ if and only if $j = \hat{f}_b$. We define binary variables $z_a(i)$ and $z_b(j)$ as follows:

$$z_a(i) = \begin{cases} 1 & \text{if } i \leq \hat{f}_a, \\ 0 & \text{otherwise,} \end{cases} \quad z_b(j) = \begin{cases} 1 & \text{if } j \leq \hat{f}_b, \\ 0 & \text{otherwise.} \end{cases}$$

For complete rounding, it can be verified that

$$\mathbb{E}(z_a(i)) = Y_a(i). \quad (15)$$

Furthermore, we also define binary variables $z_{ab}(i, j)$ that indicate whether i and j are contained within the interval defined by \hat{f}_a and \hat{f}_b . Formally,

$$z_{ab}(i, j) = \begin{cases} 1 & \text{if } \min\{i, j\} \geq \min\{\hat{f}_a, \hat{f}_b\} \text{ and } \max\{i, j\} < \max\{\hat{f}_a, \hat{f}_b\}, \\ 0 & \text{otherwise.} \end{cases}$$

For complete rounding, it can be verified that

$$\mathbb{E}(z_{ab}(i, j)) = |Y_a(i) - Y_b(j)|. \quad (16)$$

Using the result of Flach and Schlesinger (2006), we know that

$$\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b) = \sum_{i=1}^{h-1} z_a(i) D_1(i) + \sum_{j=1}^{h-1} z_b(j) D_1(j) + \sum_{i=1}^{h-1} \sum_{j=1}^{h-1} z_{ab}(i, j) D_2(i, j).$$

The proof of the lemma follows by taking the expectation of the LHS and the RHS of the above equation and simplifying the RHS using the linearity of expectation and equations (15) and (16). ■

In order to state the next lemma, we require the definition of uncrossing pairwise variables. Given unary variables \mathbf{y}_a and \mathbf{y}_b , the pairwise variable vector \mathbf{y}'_{ab} is called uncrossing with respect to \mathbf{y}_a and \mathbf{y}_b if it satisfies the following properties:

$$\begin{aligned} \sum_{j=1}^h y'_{ab}(i, j) &= y_a(i), \forall i \in \{1, 2, \dots, h\}, \\ \sum_{i=1}^h y'_{ab}(i, j) &= y_b(j), \forall j \in \{1, 2, \dots, h\}, \\ y'_{ab}(i, j) &\geq 0, \forall i, j \in \{1, 2, \dots, h\}, \\ \min\{y'_{ab}(i, j'), y'_{ab}(i', j)\} &= 0, \forall i, j, i', j' \in \{1, 2, \dots, h\}, i < i', j < j'. \end{aligned} \quad (17)$$

The following lemma establishes a connection between the expected distance between the labels assigned by complete rounding and the pairwise cost specified by uncrossing pairwise variables.

Lemma 12 *Let \mathbf{y} be a feasible solution of the LP relaxation, and $\hat{\mathbf{y}}$ be the integer solution obtained by the complete rounding procedure for \mathbf{y} . Furthermore, let \mathbf{y}'_{ab} be uncrossing pairwise variables with respect to \mathbf{y}_a and \mathbf{y}_b . Then, the following equation holds true:*

$$\mathbb{E}(\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b)) = \sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) y'_{ab}(i, j).$$

Proof We define \mathbf{Y}_a and \mathbf{Y}_b to be the cumulative distributions corresponding to \mathbf{y}_a and \mathbf{y}_b respectively. We claim that the uncrossing property (17) implies the following condition:

$$\sum_{i=1}^{i'} \sum_{j=1}^{j'} y'_{ab}(i, j) = \min\{Y_a(i'), Y_b(j')\}, \forall i', j' \in \{1, \dots, h\}. \quad (18)$$

To prove this claim, assume that $Y_a(i') < Y_b(j')$. The other cases can be handled similarly. Since \mathbf{y}'_{ab} satisfies the constraints of the LP relaxation, it follows that:

$$\begin{aligned} \sum_{i=1}^h \sum_{j=1}^{j'} y'_{ab}(i, j) &= Y_b(j'), \\ \sum_{i=1}^{i'} \sum_{j=1}^h y'_{ab}(i, j) &= Y_a(i'). \end{aligned} \quad (19)$$

Since the LHS of equality (18) is less than or equal to the LHS of both the above equations, it follows that

$$\sum_{i=1}^{i'} \sum_{j=1}^{j'} y'_{ab}(i, j) \leq \min\{Y_a(i'), Y_b(j')\}. \quad (20)$$

Therefore, there must exist a $k > i'$ and $k' \leq j'$ such that $y'_{ab}(k, k') \neq 0$. Otherwise, the LHS in the above inequality will be exactly equal to $Y_b(j')$, which would result in a contradiction. By the uncrossing property (17), we know that $\min\{y'_{ab}(i, j), y'_{ab}(k, k')\} = 0$ if $i \leq i'$ and $j > j'$. Therefore, $y'_{ab}(i, j) = 0$ for all $i \leq i'$ and $j > j'$, which proves the claim.

Combining equations (19) and (20), we get the following:

$$\sum_{i=1}^{i'} \sum_{j=j'+1}^h y'_{ab}(i, j) + \sum_{i=i'+1}^h \sum_{j=1}^{j'} y'_{ab}(i, j) = |Y_a(i') - Y_b(j')|, \forall i', j' \in \{1, \dots, h\}.$$

By solving for \mathbf{y}'_{ab} using the above equations, we get

$$\sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) y'_{ab}(i, j) = \sum_{i=1}^{h-1} Y_a(i) D_1(i) + \sum_{j=1}^{h-1} Y_b(j) D_1(j) + \sum_{i=1}^{h-1} \sum_{j=1}^{h-1} |Y_a(i) - Y_b(j)| D_2(i, j).$$

Using the previous lemma, this proves that

$$\mathbb{E}(\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b)) = \sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) y'_{ab}(i, j).$$

■

Our next lemma establishes that uncrossing pairwise variables are optimal for submodular distance functions.

Lemma 13 *Let \mathbf{y}'_{ab} be the uncrossing pairwise variables with respect to the unary variables \mathbf{y}_a and \mathbf{y}_b . Let $\bar{d} : \mathbf{L} \times \mathbf{L} \rightarrow \mathbb{R}^+$ be a submodular distance function. Then the following condition holds true:*

$$\begin{aligned} \mathbf{y}'_{ab} = & \operatorname{argmin}_{\bar{\mathbf{y}}_{ab}} \sum_{i=1}^h \sum_{j=1}^h \bar{d}(i, j) \bar{y}_{ab}(i, j), & (21) \\ \text{s.t.} & \sum_{j=1}^h \bar{y}_{ab}(i, j) = y_a(i), \forall i \in \{1, \dots, h\}, \\ & \sum_{i=1}^h \bar{y}_{ab}(i, j) = y_b(j), \forall j \in \{1, \dots, h\}, \\ & \bar{y}_{ab}(i, j) \geq 0, \forall i, j \in \{1, \dots, h\}. \end{aligned}$$

Proof We prove the lemma by contradiction. Suppose that the optimal solution to the above problem is \mathbf{y}''_{ab} , which is not uncrossing. Let

$$\min\{y''_{ab}(i, j'), y''_{ab}(i', j)\} = \lambda \neq 0,$$

where $i < i'$ and $j < j'$. Since \bar{d} is submodular, it implies that

$$\bar{d}(l_i, l_j) + \bar{d}(l_{i'}, l_{j'}) \leq \bar{d}(l_i, l_{j'}) + \bar{d}(l_{i'}, l_j).$$

Therefore the objective function of problem (21) can be reduced further by the following modification:

$$\begin{aligned} y''_{ab}(i, j') &\leftarrow y''_{ab}(i, j') - \lambda, y''_{ab}(i', j) \leftarrow y''_{ab}(i', j) - \lambda, \\ y''_{ab}(i, j) &\leftarrow y''_{ab}(i, j) + \lambda, y''_{ab}(i', j') \leftarrow y''_{ab}(i', j') + \lambda. \end{aligned}$$

The resulting contradiction proves our claim that the uncrossing pairwise variables \mathbf{y}'_{ab} are an optimal solution of problem (21). ■

Using the above lemmas, we will now obtain the tight approximation factor of the complete rounding procedure.

Lemma 14 *The tight approximation factor of the complete rounding procedure is equal to the submodular distortion of the distance function.*

Proof We denote a feasible fractional solution of the LP relaxation by \mathbf{y} and the rounded solution by $\hat{\mathbf{y}}$. Consider a pair of neighboring random variables $(X_a, X_b) \in \mathbf{X}$. We define uncrossing pairwise variables \mathbf{y}'_{ab} with respect to \mathbf{y}_a and \mathbf{y}_b . Using Lemmas 12 and 13, the approximation factor of the

complete rounding procedure can be shown to be at most B as follows:

$$\begin{aligned}
 \mathbb{E}(d(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b)) &= \sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) y'_{ab}(i, j) \\
 &\leq \sum_{i=1}^h \sum_{j=1}^h \bar{d}(l_i, l_j) y'_{ab}(i, j) \\
 &\leq \sum_{i=1}^h \sum_{j=1}^h \bar{d}(l_i, l_j) y_{ab}(i, j) \\
 &\leq B \sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) y_{ab}(i, j).
 \end{aligned}$$

In order to prove that the approximation factor of the complete rounding is exactly B , we need an example where the above inequality holds as an equality. The key to obtaining a tight example lies in the Lagrangian dual of problem (7). In order to specify its dual, we need three types of dual variables. The first type, denoted by $\alpha(i, j)$, corresponds to the constraint

$$d'(l_i, l_j) \leq td(l_i, l_j).$$

The second type, denoted by $\beta(i, j)$, corresponds to the constraint

$$d'(l_i, l_j) \geq d(l_i, l_j).$$

The third type, denoted by $\gamma(i, j)$, corresponds to the constraint

$$d'(l_i, l_j) + d'(l_{i+1}, l_{j+1}) \leq d'(l_i, l_{j+1}) + d'(l_{i+1}, l_j).$$

Using the above variables, the dual of problem (7) is given by

$$\begin{aligned}
 \max \quad & \sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) \beta(i, j) & (22) \\
 \text{s.t.} \quad & \sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) \alpha(i, j) = 1, \\
 & \beta(i, j) = \alpha(i, j) - \gamma(i, j-1) - \gamma(i-1, j) + \gamma(i-1, j-1) + \gamma(i, j), \\
 & \forall i, j \in \{1, \dots, h\}, \\
 & \alpha(i, j) \geq 0, \beta(i, j) \geq 0, \gamma(i, j) \geq 0, \forall i, j \in \{1, \dots, h\}.
 \end{aligned}$$

We claim that the above dual problem has an optimal solution $(\alpha^*, \beta^*, \gamma^*)$ that satisfies the following property:

$$\min\{\beta^*(i, j'), \beta^*(i', j)\} = 0, \forall i, i', j, j' \in \{1, \dots, h\}, i < i', j < j'. \quad (23)$$

We refer to the optimal dual solution β^* that satisfies the above property as uncrossing dual variables as it is analogous to uncrossing pairwise variables. The above claim, namely, the existence of an uncrossing optimal dual solution, can be proved by contradiction as follows. Suppose there exists no

optimal solution that satisfies the above property. Then consider the following problem, which is the dual of the problem of finding the tightest submodular overestimate of the submodular function \bar{d} :

$$\begin{aligned}
 \max \quad & \sum_{i=1}^h \sum_{j=1}^h \bar{d}(l_i, l_j) \beta(i, j) \\
 \text{s.t.} \quad & \sum_{i=1}^h \sum_{j=1}^h \bar{d}(l_i, l_j) \alpha(i, j) = 1, \\
 & \beta(i, j) = \alpha(i, j) - \gamma(i, j-1) - \gamma(i-1, j) + \gamma(i-1, j-1) + \gamma(i, j), \\
 & \forall i, j \in \{1, \dots, h\}, \\
 & \alpha(i, j) \geq 0, \beta(i, j) \geq 0, \gamma(i, j) \geq 0, \forall i, j \in \{1, \dots, h\}.
 \end{aligned} \tag{24}$$

By strong duality, problem (24) has an optimal value of 1. However, the optimal solution of problem (22), which is also a feasible solution for problem (24), provides a value strictly greater than 1. This results in a contradiction that proves our claim.

The optimal dual variables that satisfy property (23) allow us to construct an example that proves that the approximation factor B of the complete rounding procedure is tight. Specifically, we define

$$\begin{aligned}
 \bar{y}_{ab}(i, j) &= \frac{\alpha^*(i, j)}{\sum_{i'=1}^h \sum_{j'=1}^h \alpha^*(i', j')}, \forall i, j \in \{1, \dots, h\}, \\
 \bar{y}_a(i) &= \sum_{j=1}^h \bar{y}_{ab}(i, j), \forall i \in \{1, \dots, h\}, \\
 \bar{y}_b(j) &= \sum_{i=1}^h \bar{y}_{ab}(i, j), \forall j \in \{1, \dots, h\}.
 \end{aligned}$$

Note that the pairwise variables \bar{y}_{ab} must minimize the pairwise potential corresponding to the unary variables \mathbf{y}_a and \mathbf{y}_b , that is,

$$\begin{aligned}
 \bar{\mathbf{y}}_{ab} &= \operatorname{argmin}_{\mathbf{y}_{ab}} \sum_{l_i, l_j \in \mathbf{L}} d(l_i, l_j) y_{ab}(i, j) \\
 \text{s.t.} \quad & \sum_{l_j \in \mathbf{L}} y_{ab}(i, j) = \bar{y}_a(i), \forall l_i \in \mathbf{L} \\
 & \sum_{l_i \in \mathbf{L}} y_{ab}(i, j) = \bar{y}_b(j), \forall l_j \in \mathbf{L} \\
 & y_{ab}(i, j) \geq 0, \forall l_i, l_j \in \mathbf{L}.
 \end{aligned} \tag{25}$$

If the above statement was not true, then the value of the dual problem (22) could be increased further.

We also define the following pairwise variables:

$$y'_{ab}(i, j) = \frac{\beta^*(i, j)}{\sum_{i'=1}^h \sum_{j'=1}^h \beta^*(i', j')}, \forall i, j \in \{1, \dots, h\},$$

$$y'_a(i) = \sum_{j=1}^h y'_{ab}(i, j), \forall i \in \{1, \dots, h\},$$

$$y'_b(j) = \sum_{i=1}^h y'_{ab}(i, j), \forall j \in \{1, \dots, h\}.$$

It can be verified that

$$y'_a(i) = \bar{y}_a(i), y'_b(j) = \bar{y}_b(j), \forall i, j \in \{1, \dots, h\}.$$

The above condition follows from the constraints of problem (22). Due to the uncrossing property of β^* , the pairwise variables y'_{ab} are uncrossing with respect to \bar{y}_a and \bar{y}_b . By Lemma 12, this implies that

$$\mathbb{E}(\hat{d}(\hat{y}_a, \hat{y}_b)) = \sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) y'_{ab}(i, j),$$

where \hat{y}_a and \hat{y}_b are integer solutions obtained by the complete rounding procedure. By strong duality, it follows that

$$\mathbb{E}(\hat{d}(\hat{y}_a, \hat{y}_b)) = B \sum_{i=1}^h \sum_{j=1}^h d(l_i, l_j) \bar{y}_{ab}(i, j). \quad (26)$$

The existence of an example that satisfies properties (25) and (26) implies that the tight approximation factor of the complete rounding procedure is B . \blacksquare

Lemmas 10 and 14 together prove Theorem 1.

Appendix B. Proof of Theorem 3

We begin by establishing the theoretical properties of the interval-move making algorithm. Recall that, given an interval of labels $\mathbf{I} = \{l_s, \dots, l_e\}$ of at most q consecutive labels and a labeling \hat{x} , we define $\mathbf{I}_a = \mathbf{I} \cup \{\hat{x}_a\}$ for all random variables $X_a \in \mathbf{X}$. In order to use the interval move-making algorithm, we compute a submodular distance function $\bar{d}_{\hat{x}_a, \hat{x}_b} : \mathbf{I}_a \times \mathbf{I}_b \rightarrow \mathbb{R}^+$ for all pairs of neighboring random variables $(X_a, X_b) \in \mathbf{E}$ as follows:

$$\begin{aligned} \bar{d}_{\hat{x}_a, \hat{x}_b} = & \operatorname{argmin}_{d'} t & (27) \\ \text{s.t.} & d'(l_i, l_j) \leq t d(l_i, l_j), \forall l_i \in \mathbf{I}_a, l_j \in \mathbf{I}_b, \\ & d'(l_i, l_j) \geq d(l_i, l_j), \forall l_i \in \mathbf{I}_a, l_j \in \mathbf{I}_b, \\ & d'(l_i, l_j) + d'(l_{i+1}, l_{j+1}) \leq d'(l_i, l_{j+1}) + d'(l_{i+1}, l_j), \forall l_i, l_j \in \mathbf{I} \setminus \{l_e\}, \\ & d'(l_i, l_e) + d'(l_{i+1}, \hat{x}_b) \leq d'(l_i, \hat{x}_b) + d'(l_{i+1}, l_e), \forall l_i \in \mathbf{I} \setminus \{l_e\}, \\ & d'(l_e, l_j) + d'(\hat{x}_a, l_{j+1}) \leq d'(l_e, l_{j+1}) + d'(\hat{x}_a, l_j), \forall l_j \in \mathbf{I} \setminus \{l_e\}, \\ & d'(l_e, l_e) + d(\hat{x}_a, \hat{x}_b) \leq d'(l_e, \hat{x}_b) + d'(\hat{x}_a, l_e). \end{aligned}$$

For any interval \mathbf{I} and labeling \mathbf{x} , we define the following sets:

$$\begin{aligned}\mathbf{V}(\mathbf{x}, \mathbf{I}) &= \{X_a | X_a \in \mathbf{X}, x_a \in \mathbf{I}\}, \\ \mathbf{A}(\mathbf{x}, \mathbf{I}) &= \{(X_a, X_b) | (X_a, X_b) \in \mathbf{E}, x_a \in \mathbf{I}, x_b \in \mathbf{I}\}, \\ \mathbf{B}_1(\mathbf{x}, \mathbf{I}) &= \{(X_a, X_b) | (X_a, X_b) \in \mathbf{E}, x_a \in \mathbf{I}, x_b \notin \mathbf{I}\}, \\ \mathbf{B}_2(\mathbf{x}, \mathbf{I}) &= \{(X_a, X_b) | (X_a, X_b) \in \mathbf{E}, x_a \notin \mathbf{I}, x_b \in \mathbf{I}\}, \\ \mathbf{B}(\mathbf{x}, \mathbf{I}) &= \mathbf{B}_1(\mathbf{x}) \cup \mathbf{B}_2(\mathbf{x}).\end{aligned}$$

In other words, $\mathbf{V}(\mathbf{x}, \mathbf{I})$ is the set of all random variables whose label belongs to the interval \mathbf{I} . Similarly, $\mathbf{A}(\mathbf{x}, \mathbf{I})$ is the set of all neighboring random variables such that the labels assigned to both the random variables belong to the interval \mathbf{I} . The set $\mathbf{B}(\mathbf{x}, \mathbf{I})$ contains the set of all neighboring random variables such that only one of the two labels assigned to the two random variables belongs to the interval \mathbf{I} . Given the set of all intervals \mathcal{I} and a labeling $\hat{\mathbf{x}}$, we define the following for all $x_a, x_b \in \mathbf{L}$:

$$\begin{aligned}D(x_a, x_b; \hat{x}_a, \hat{x}_b) &= \sum_{\mathbf{I} \in \mathcal{I}, \mathbf{A}(\mathbf{x}, \mathbf{I}) \ni (X_a, X_b)} d_{\hat{x}_a, \hat{x}_b}(x_a, x_b) \\ &+ \sum_{\mathbf{I} \in \mathcal{I}, \mathbf{B}_1(\mathbf{x}, \mathbf{I}) \ni (X_a, X_b)} d_{\hat{x}_a, \hat{x}_b}(x_a, \hat{x}_b) \\ &+ \sum_{\mathbf{I} \in \mathcal{I}, \mathbf{B}_2(\mathbf{x}, \mathbf{I}) \ni (X_a, X_b)} d_{\hat{x}_a, \hat{x}_b}(\hat{x}_a, x_b).\end{aligned}$$

Using the above notation, we are ready to state the following lemma on the theoretical guarantee of the interval move-making algorithm.

Lemma 15 *The tight multiplicative bound of the interval move-making algorithm is equal to*

$$\frac{1}{q} \max_{x_a, x_b, \hat{x}_a, \hat{x}_b \in \mathbf{L}, x_a \neq x_b} \frac{D(x_a, x_b; \hat{x}_a, \hat{x}_b)}{d(x_a, x_b)}.$$

Proof We denote an optimal labeling by \mathbf{x}^* and the estimated labeling by $\hat{\mathbf{x}}$. Let $t \in [1, q]$ be a uniformly distributed random integer. Using t , we define the following set of non-overlapping intervals:

$$\mathcal{I}_t = \{[1, t], [t+1, t+q], \dots, [\cdot, h]\}.$$

For each interval $\mathbf{I} \in \mathcal{I}_t$, we define a labeling $\mathbf{x}^{\mathbf{I}}$ as follows:

$$x_a^{\mathbf{I}} = \begin{cases} x_a^* & \text{if } x_a^* \in \mathbf{I}, \\ \hat{x}_a & \text{otherwise.} \end{cases}$$

Since $\hat{\mathbf{x}}$ is the labeling obtained after the interval move-making algorithm converges, it follows that

$$\sum_{X_a \in \mathbf{X}} \theta_a(\hat{x}_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(\hat{x}_a, \hat{x}_b) \leq \sum_{X_a \in \mathbf{X}} \theta_a(x_a^{\mathbf{I}}) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(x_a^{\mathbf{I}}, x_b^{\mathbf{I}}).$$

By canceling out the common terms, we can simplify the above inequality as

$$\begin{aligned}
 & \sum_{X_a \in \mathbf{V}(\mathbf{x}^*, \mathbf{I})} \theta_a(\hat{x}_a) \\
 & + \sum_{(X_a, X_b) \in \mathbf{A}(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(\hat{x}_a, \hat{x}_b) \\
 & + \sum_{(X_a, X_b) \in \mathbf{B}_1(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(\hat{x}_a, \hat{x}_b) \\
 & + \sum_{(X_a, X_b) \in \mathbf{B}_2(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(\hat{x}_a, \hat{x}_b) \\
 \leq & \sum_{X_a \in \mathbf{V}(\mathbf{x}^*, \mathbf{I})} \theta_a(x_a^*) \\
 & + \sum_{(X_a, X_b) \in \mathbf{A}(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(x_a^*, x_b^*) \\
 & + \sum_{(X_a, X_b) \in \mathbf{B}_1(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(x_a^*, \hat{x}_b) \\
 & + \sum_{(X_a, X_b) \in \mathbf{B}_2(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(\hat{x}_a, x_b^*).
 \end{aligned}$$

We now sum the above inequality over all the intervals $\mathbf{I} \in \mathcal{I}_t$. Note that the resulting LHS is at least equal to the energy of the labeling $\hat{\mathbf{x}}$ when the distance function between the random variables (X_a, X_b) is $\bar{d}_{\hat{x}_a, \hat{x}_b}$. This implies that

$$\begin{aligned}
 & \sum_{X_a \in \mathbf{X}} \theta_a(\hat{x}_a) + \sum_{(X_a, X_b) \in \mathbf{E}} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(\hat{x}_a, \hat{x}_b) \\
 \leq & \sum_{X_a \in \mathbf{X}} \theta_a(x_a^*) \\
 & + \sum_{\mathbf{I} \in \mathcal{I}_t} \sum_{(X_a, X_b) \in \mathbf{A}(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(x_a^*, x_b^*) \\
 & + \sum_{\mathbf{I} \in \mathcal{I}_t} \sum_{(X_a, X_b) \in \mathbf{B}_1(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(x_a^*, \hat{x}_b) \\
 & + \sum_{\mathbf{I} \in \mathcal{I}_t} \sum_{(X_a, X_b) \in \mathbf{B}_2(\mathbf{x}^*, \mathbf{I})} w_{ab} \bar{d}_{\hat{x}_a, \hat{x}_b}(\hat{x}_a, x_b^*).
 \end{aligned}$$

Taking the expectation on both sides of the above inequality with respect to the uniformly distributed random integer $t \in [1, q]$ proves that the multiplicative bound of the interval move-making algorithm is at most equal to

$$\frac{1}{q} \max_{x_a, x_b, \hat{x}_a, \hat{x}_b \in \mathbf{L}, x_a \neq x_b} \frac{D(x_a, x_b; \hat{x}_a, \hat{x}_b)}{d(x_a, x_b)}.$$

A tight example with two random variables X_a and X_b with $w_{ab} = 1$ can be constructed similar to the one shown in Lemma 10. ■

We now turn our attention to the interval rounding procedure. Let \mathbf{y} be a feasible solution of the LP relaxation, and $\hat{\mathbf{y}}$ be the integer solution obtained using interval rounding. Once again, we define the unary variable vector \mathbf{y}_a and the pairwise variable vector \mathbf{y}_{ab} as specified in equations (12) and (13) respectively. Similar to the previous appendix, we denote the expected distance between $\hat{\mathbf{y}}_a$ and $\hat{\mathbf{y}}_b$ as $\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b)$.

Given an interval $\mathbf{I} = \{l_s, \dots, l_e\}$ of at most q consecutive labels, we define a vector $\mathbf{Y}_a^{\mathbf{I}}$ for each random variable X_a as follows:

$$Y_a^{\mathbf{I}}(i) = \sum_{j=s}^{s+i-1} y_a(j), \forall i \in \{1, \dots, e-s+1\}.$$

In other words, $\mathbf{Y}_a^{\mathbf{I}}$ is the cumulative distribution of \mathbf{y}_a within the interval \mathbf{I} . Furthermore, for each pair of neighboring random variables $(X_a, X_b) \in \mathbf{E}$ we define

$$\begin{aligned} Z_{ab}^{\mathbf{I}} &= \max\{Y_a^{\mathbf{I}}(e-s+1), Y_b^{\mathbf{I}}(e-s+1)\} - \min\{Y_a^{\mathbf{I}}(e-s+1), Y_b^{\mathbf{I}}(e-s+1)\}, \\ Z_a^{\mathbf{I}}(i) &= \min\{Y_a^{\mathbf{I}}(i), Y_b^{\mathbf{I}}(e-s+1)\}, \forall i \in \{1, \dots, e-s+1\}, \\ Z_b^{\mathbf{I}}(j) &= \min\{Y_b^{\mathbf{I}}(j), Y_a^{\mathbf{I}}(e-s+1)\}, \forall j \in \{1, \dots, e-s+1\}. \end{aligned}$$

The following shorthand notation would be useful to concisely specify the exact form of $\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b)$.

$$\begin{aligned} D_0^{\mathbf{I}} &= \max_{l_i, l_j, |\{l_i, l_j\} \cap \mathbf{I}|=1} d(l_i, l_j), \\ D_1^{\mathbf{I}}(i) &= \frac{1}{2} (d(l_{s+i-1}, l_s) + d(l_{s+i-1}, l_e) - d(l_{s+i}, l_s) - d(l_{s+i}, l_e)), \forall i \in \{1, \dots, e-s\}, \\ D_2^{\mathbf{I}}(i, j) &= \frac{1}{2} (d(l_{s+i-1}, l_{s+j}) + d(l_{s+i}, l_{s+j-1}) - d(l_{s+i-1}, l_{s+j-1}) - d(l_{s+i}, l_{s+j})), \\ &\quad \forall i, j \in \{1, \dots, e-s\}. \end{aligned}$$

In other words, $\mathbf{D}_0^{\mathbf{I}}$ is the maximum distance between two labels such that only one of the two labels lies in the interval \mathbf{I} . The terms $\mathbf{D}_1^{\mathbf{I}}$ and $\mathbf{D}_2^{\mathbf{I}}$ are analogous to the terms defined in equation (14). Using the above notation, we can state the following lemma on the expected distance of the rounded solution for two neighboring random variables.

Lemma 16 *Let \mathbf{y} be a feasible solution of the LP relaxation, and $\hat{\mathbf{y}}$ be the integer solution obtained by the interval rounding procedure for \mathbf{y} using the set of intervals \mathcal{I} . Then, the following inequality holds true:*

$$\begin{aligned} \mathbb{E}(\hat{d}(\hat{\mathbf{y}}_a, \hat{\mathbf{y}}_b)) &\leq \frac{1}{q} \left(\sum_{\mathbf{I}=\{l_s, \dots, l_e\} \in \mathcal{I}} Z_{ab}^{\mathbf{I}} D_0^{\mathbf{I}} + \sum_{i=1}^{e-s} Z_a^{\mathbf{I}}(i) D_1^{\mathbf{I}}(i) + \sum_{j=1}^{e-s} Z_b^{\mathbf{I}}(j) D_1^{\mathbf{I}}(j) + \right. \\ &\quad \left. \sum_{i=1}^{e-s} \sum_{j=1}^{e-s} |Z_a^{\mathbf{I}}(i) - Z_b^{\mathbf{I}}(j)| D_2^{\mathbf{I}}(i, j) \right). \end{aligned}$$

Proof We begin the proof by establishing the probability of a random variable X_a being assigned a label in an iteration of the interval rounding procedure. The total number of intervals in the set

\mathcal{I} is $h + q - 1$. Out of all the intervals, each label l_i is present in q intervals. Thus, the probability of choosing an interval that contains the label l_i is $q/(h + q - 1)$. Once an interval containing l_i is chosen, the probability of assigning the label l_i to X_a is $y_a(i)$. Thus, the probability of assigning a label l_i to X_a is $y_a(i)q/(h + q - 1)$. Summing over all labels l_i , we observe that the probability of assigning a label to X_a in an iteration of the interval rounding procedure is $q/(h + q - 1)$.

Now we consider two neighboring random variables $(X_a, X_b) \in \mathbf{E}$. In the current iteration of interval rounding, given an interval \mathbf{I} , the probability of exactly one of the two random variables getting assigned a label in \mathbf{I} is exactly equal to $Z_{ab}^{\mathbf{I}}$. In this case, we have to assume that the expected distance between the two variables will be the maximum possible, that is, $D_0^{\mathbf{I}}$. If both the variables are assigned a label in \mathbf{I} , then a slight modification of Lemma 11 gives us the expected distance as

$$\sum_{i=1}^{e-s} Z_a^{\mathbf{I}}(i) D_1^{\mathbf{I}}(i) + \sum_{j=1}^{e-s} Z_b^{\mathbf{I}}(j) D_1^{\mathbf{I}}(j) + \sum_{i=1}^{e-s} \sum_{j=1}^{e-s} |Z_a^{\mathbf{I}}(i) - Z_b^{\mathbf{I}}(j)| D_2^{\mathbf{I}}(i, j).$$

Thus, the expected distance between the labels of X_a and X_b conditioned on at least one of the two random variables getting assigned in an iteration is equal to

$$\frac{1}{(h + q - 1)} \left(\sum_{\mathbf{I}=\{l_s, \dots, l_e\} \in \mathcal{I}} Z_{ab}^{\mathbf{I}} D_0^{\mathbf{I}} + \sum_{i=1}^{e-s} Z_a^{\mathbf{I}}(i) D_1^{\mathbf{I}}(i) + \sum_{j=1}^{e-s} Z_b^{\mathbf{I}}(j) D_1^{\mathbf{I}}(j) + \sum_{i=1}^{e-s} \sum_{j=1}^{e-s} |Z_a^{\mathbf{I}}(i) - Z_b^{\mathbf{I}}(j)| D_2^{\mathbf{I}}(i, j) \right),$$

where the term $1/(h + q - 1)$ corresponds to the probability of choosing an interval from the set \mathcal{I} . In order to compute $\hat{d}(\hat{y}_a, \hat{y}_b)$, we need to divide the above term with the probability of at least one of the two random variables getting assigned a label in the current iteration. Since the two cumulative distributions $\mathbf{Y}_a^{\mathbf{I}}$ and $\mathbf{Y}_b^{\mathbf{I}}$ can be arbitrarily close to each other without being exactly equal, we can only lower bound the probability of at least one of X_a and X_b being assigned a label at the current iteration by the probability of X_a being assigned a label in the current iteration. Since the probability of X_a being assigned a label is $q/(h + q - 1)$, it follows that

$$\mathbb{E}(\hat{d}(\hat{y}_a, \hat{y}_b)) \leq \frac{1}{q} \left(\sum_{\mathbf{I}=\{l_s, \dots, l_e\} \in \mathcal{I}} Z_{ab}^{\mathbf{I}} D_0^{\mathbf{I}} + \sum_{i=1}^{e-s} Z_a^{\mathbf{I}}(i) D_1^{\mathbf{I}}(i) + \sum_{j=1}^{e-s} Z_b^{\mathbf{I}}(j) D_1^{\mathbf{I}}(j) + \sum_{i=1}^{e-s} \sum_{j=1}^{e-s} |Z_a^{\mathbf{I}}(i) - Z_b^{\mathbf{I}}(j)| D_2^{\mathbf{I}}(i, j) \right).$$

■

Using the above lemma, we can now establish the theoretical property of the interval rounding procedure.

Lemma 17 *The tight approximation factor of the interval rounding procedure is equal to*

$$\frac{1}{q} \max_{x_a, x_b, \hat{x}_a, \hat{x}_b \in \mathbf{L}, x_a \neq x_b} \frac{D(x_a, x_b; \hat{x}_a, \hat{x}_b)}{d(x_a, x_b)}.$$

Proof Similar to Lemma 14, the key to proving the above lemma lies in the dual of problem (27). Given the current labels \hat{x}_a and \hat{x}_b of the random variables X_a and X_b respectively, as well as an interval $\mathbf{I} = \{l_s, \dots, l_e\}$, problem (27) computes the tightest submodular overestimation $\bar{d}_{\hat{x}_a, \hat{x}_b} : \mathbf{I}_a \times \mathbf{I}_b \rightarrow \mathbb{R}^+$ where $\mathbf{I}_a = \mathbf{I} \cup \{\hat{x}_a\}$ and $\mathbf{I}_b = \mathbf{I} \cup \{\hat{x}_b\}$. Similar to problem (22), the dual of problem (27) consists of three types of dual variables. The first type, denoted by $\alpha(i, j; \hat{x}_a, \hat{x}_b, \mathbf{I})$ corresponds to the constraint

$$d'(l_i, l_j) \leq td(l_i, l_j).$$

The second type, denoted by $\beta(i, j; \hat{x}_a, \hat{x}_b, \mathbf{I})$, corresponds to the constraint

$$d'(l_i, l_j) \geq d(l_i, l_j).$$

The third type, denoted by $\gamma(i, j; \hat{x}_a, \hat{x}_b, \mathbf{I})$, corresponds to the constraint

$$d'(l_i, l_j) + d'(l_{i+1}, l_{j+1}) \leq d'(l_i, l_{j+1}) + d'(l_{i+1}, l_j).$$

Let $(\alpha^*(\hat{x}_a, \hat{x}_b, \mathbf{I}), \beta^*(\hat{x}_a, \hat{x}_b, \mathbf{I}), \gamma^*(\hat{x}_a, \hat{x}_b, \mathbf{I}))$ denote an optimal solution of the dual. A modification of Lemma 14 can be used to show that there must exist an optimal solution such that $\beta^*(\hat{x}_a, \hat{x}_b, \mathbf{I})$ is uncrossing.

We consider the values of \hat{x}_a and \hat{x}_b for which we obtain the tight multiplicative bound of the interval move-making algorithm. For these values of \hat{x}_a and \hat{x}_b , we define

$$\begin{aligned} \bar{y}_{ab}(i, j; \hat{x}_a, \hat{x}_b) &= \frac{\sum_{\mathbf{I} \in \mathcal{I}} \alpha^*(i, j; \hat{x}_a, \hat{x}_b, \mathbf{I})}{\sum_{i', j'} \sum_{\mathbf{I} \in \mathcal{I}} \alpha^*(i', j'; \hat{x}_a, \hat{x}_b, \mathbf{I})}, \forall i, j \in \{1, \dots, h\}, \\ \bar{y}_a(i; \hat{x}_a, \hat{x}_b) &= \sum_{j=1}^h \bar{y}_{ab}(i, j; \hat{x}_a, \hat{x}_b), \forall i \in \{1, \dots, h\}, \\ \bar{y}_b(j; \hat{x}_a, \hat{x}_b) &= \sum_{i=1}^h \bar{y}_{ab}(i, j; \hat{x}_a, \hat{x}_b), \forall j \in \{1, \dots, h\}, \\ y'_{ab}(i, j; \hat{x}_a, \hat{x}_b) &= \frac{\sum_{\mathbf{I} \in \mathcal{I}} \beta^*(i, j; \hat{x}_a, \hat{x}_b, \mathbf{I})}{\sum_{i', j'} \sum_{\mathbf{I} \in \mathcal{I}} \beta^*(i', j'; \hat{x}_a, \hat{x}_b, \mathbf{I})}, \forall i, j \in \{1, \dots, h\}, \\ y'_a(i; \hat{x}_a, \hat{x}_b) &= \sum_{j=1}^h y'_{ab}(i, j; \hat{x}_a, \hat{x}_b), \forall i \in \{1, \dots, h\}, \\ y'_b(j; \hat{x}_a, \hat{x}_b) &= \sum_{i=1}^h y'_{ab}(i, j; \hat{x}_a, \hat{x}_b), \forall j \in \{1, \dots, h\}. \end{aligned}$$

The constraints of the dual of problem (27) ensure that

$$\begin{aligned} \bar{y}_a(i; \hat{x}_a, \hat{x}_b) &= y'_a(i; \hat{x}_a, \hat{x}_b), \forall i \in \{1, \dots, h\}, \\ \bar{y}_b(j; \hat{x}_a, \hat{x}_b) &= y'_b(j; \hat{x}_a, \hat{x}_b), \forall j \in \{1, \dots, h\}. \end{aligned}$$

Given the distributions $\bar{y}_a(\hat{x}_a, \hat{x}_b)$ and $\bar{y}_b(\hat{x}_a, \hat{x}_b)$, we claim that the pairwise variables $\bar{y}_{ab}(\hat{x}_a, \hat{x}_b)$ must minimize the pairwise potentials corresponding to these distributions, that is,

$$\begin{aligned} \bar{y}_{ab}(\hat{x}_a, \hat{x}_b) = & \operatorname{argmin}_{\mathbf{y}_{ab}} \sum_{l_i, l_j \in \mathbf{L}} d(l_i, l_j) y_{ab}(i, j) \\ \text{s.t.} & \sum_{l_j \in \mathbf{L}} y_{ab}(i, j) = \bar{y}_a(i; \hat{x}_a, \hat{x}_b), \forall l_i \in \mathbf{L} \\ & \sum_{l_i \in \mathbf{L}} y_{ab}(i, j) = \bar{y}_b(j; \hat{x}_a, \hat{x}_b), \forall l_j \in \mathbf{L} \\ & y_{ab}(i, j) \geq 0, \forall l_i, l_j \in \mathbf{L}. \end{aligned}$$

If the above claim was not true, then we could further increase the value of at least one of the duals of problem (27) corresponding to some interval \mathbf{I} . Furthermore, since $\mathbf{y}'_{ab}(\hat{x}_a, \hat{x}_b)$ is constructed using $\beta^*(\hat{x}_a, \hat{x}_b, \mathbf{I})$, which is uncrossing, a modification of Lemma 12 can be used to show that

$$\hat{d}(\hat{\mathbf{y}}_a(\hat{x}_a, \hat{x}_b), \hat{\mathbf{y}}_b(\hat{x}_a, \hat{x}_b)) = \sum_{l_i, l_j \in \mathbf{L}} d(l_i, l_j) y'_{ab}(i, j; \hat{x}_a, \hat{x}_b).$$

Here $\hat{\mathbf{y}}_a(\hat{x}_a, \hat{x}_b)$ and $\hat{\mathbf{y}}_b(\hat{x}_a, \hat{x}_b)$ are the rounded solutions obtained by the interval rounding procedure applied to $\bar{y}_a(\hat{x}_a, \hat{x}_b)$ and $\bar{y}_b(\hat{x}_a, \hat{x}_b)$.

By duality, it follows that

$$\frac{\sum_{l_i, l_j \in \mathbf{L}} d(l_i, l_j) y'_{ab}(i, j; \hat{x}_a, \hat{x}_b)}{\sum_{l_i, l_j \in \mathbf{L}} d(l_i, l_j) \bar{y}_{ab}(i, j; \hat{x}_a, \hat{x}_b)} \leq \frac{1}{q} \max_{x_a, x_b, \hat{x}_a, \hat{x}_b \in \mathbf{L}, x_a \neq x_b} \frac{D(x_a, x_b; \hat{x}_a, \hat{x}_b)}{d(x_a, x_b)}.$$

Strong duality implies that there must exist a set of variables for which the above inequality holds as an equality. This proves the lemma. \blacksquare

Lemmas 15 and 17 together prove Theorem 3.

Appendix C. Proof of Theorem 8

Proof The proof of the above theorem proceeds in a similar fashion to the previous two theorems, namely, by computing the dual of the problems that are used to obtain the tightest submodular overestimation of the given distance function. In what follows, we provide a brief sketch of the proof and omit the details.

We start with the hierarchical move-making algorithm. Recall that this algorithm computes a labeling $\mathbf{x}^{i,j}$ for each cluster $\mathbf{C}(i, j) \subseteq \mathbf{L}$, where $\mathbf{C}(i, j)$ denotes the j -th cluster at the i -th level. In order to obtain a labeling $\mathbf{x}^{i,j}$ it makes use of the labelings computed at the $(i+1)$ -th level. Specifically, let $\mathbf{L}_a^{i,j} = \{x_a^{i+1, j'}, p(i+1, j') = j, j' \in \{1, \dots, h^{i+1}\}\}$. In other words, $\mathbf{L}_a^{i,j}$ is the set of labels that were assigned to the random variable X_a by the labelings corresponding to the children of the current cluster $\mathbf{C}(i, j)$. In order to compute the labeling $\mathbf{x}^{i,j}$, the algorithm has to choose one label from the set $\mathbf{L}_a^{i,j}$. This is achieved either using the complete move-making algorithm or the interval move-making algorithm. The algorithm is initialized by define $x_a^{m,j} = k$ for all $X_a \in \mathbf{X}$ where k is the unique label in the cluster $\mathbf{C}(m, j)$. The algorithm terminates by computing $\mathbf{x}^{1,1}$, which is the final labeling.

Let the multiplicative bound corresponding to the computation of $\mathbf{x}^{i,j}$ be denoted by $B^{i,j}$. From the arguments of Theorems 1 and 3, it follows that there must exist dual variables $(\alpha^*(i, j), \beta^*(i, j), \gamma^*(i, j))$ that provide an approximation factor that is exactly equal to $B^{i,j}$ and $\beta^*(i, j)$ is uncrossing. For each cluster $\mathbf{C}(i, j)$, we define variables $\delta_a(k; i, j)$ for $k \in \mathbf{L}_a^{i,j}$ as follows:

$$\delta_a(k; i, j) = \frac{\sum_{k' \in \mathbf{L}_b^{i,j}} \alpha^*(k, k'; i, j)}{\sum_{l \in \mathbf{L}_a^{i,j}} \sum_{l' \in \mathbf{L}_b^{i,j}} \alpha^*(l, l'; i, j)}.$$

Using the above variables we define unary variables $y_a(k; i, j)$ for each $k \in \mathbf{L}^{i,j}$ as follows:

$$\begin{aligned} y_a(k; 1, 1) &= \delta_a(k; 1, 1), \\ y_a(k; i, j) &= y_a(x_a^{i-1, p(i,j)}; i-1, p(i, j)) \delta_a(k; i, j). \end{aligned}$$

Similarly, we define variables $\delta_b(k'; i, j)$ for each $k' \in \mathbf{L}_b^{i,j}$ as follows:

$$\delta_b(k'; i, j) = \frac{\sum_{k \in \mathbf{L}_a^{i,j}} \alpha^*(k, k'; i, j)}{\sum_{l \in \mathbf{L}_a^{i,j}} \sum_{l' \in \mathbf{L}_b^{i,j}} \alpha^*(l, l'; i, j)}.$$

Using the above variables, we define unary variables $y_b(k'; i, j)$ for each $k' \in \mathbf{L}_b^{i,j}$ as follows:

$$\begin{aligned} y_b(k'; 1, 1) &= \delta_b(k'; 1, 1), \\ y_b(k'; i, j) &= y_a(x_a^{i-1, p(i,j)}; i-1, p(i, j)) \delta_b(k'; i, j). \end{aligned}$$

Since each cluster in the m -th level contains a single unique label, it follows that

$$\begin{aligned} \mathbf{y}_a &= [y_a(x_a^{m,j}; m, j), \forall j = 1, \dots, h], \\ \mathbf{y}_b &= [y_b(x_b^{m,j}; m, j), \forall j = 1, \dots, h], \end{aligned}$$

are valid unary variables for the LP relaxation. It can be shown that applying the hierarchical rounding procedure on the variables \mathbf{y}_a and \mathbf{y}_b provides an approximation factor that is exactly equal to the multiplicative bound of the hierarchical move-making algorithm. \blacksquare

References

- T. Ajanthan, R. Hartley, M. Salzmann, and H. Li. Iteratively reweighted graph cut for multi-label MRFs with non-convex priors. *CoRR*, 2014.
- M. Andrew, M. Hajiaghayi, H. Karloff, and A. Moitra. Capacitated metric labeling. In *SODA*, 2011.
- A. Archer, J. Fakcharoenphol, C. Harrelson, R. Krauthgamer, K. Talvar, and E. Tardos. Approximate classification via earthmover metrics. In *SODA*, 2004.
- Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 2004.
- Y. Boykov, O. Veksler, and R. Zabih. Markov random fields with efficient approximations. In *CVPR*, 1998.

- Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *ICCV*, 1999.
- C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Approximation algorithms for the metric labeling problem via a new linear programming formulation. In *SODA*, 2001.
- P. Dokania and M. P. Kumar. Parsimonious labeling. In *ICCV*, 2015.
- J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. In *STOC*, 2003.
- B. Flach and D. Schlesinger. Transforming an arbitrary minsum problem into a binary one. Technical report, TU Dresden, 2006.
- A. Globerson and T. Jaakkola. Fixing max-product: Convergent message passing algorithms for MAP LP-relaxations. In *NIPS*, 2007.
- A. Gupta and E. Tardos. A constant factor approximation algorithm for a class of classification problems. In *STOC*, 2000.
- T. Hazan and A. Shashua. Convergent message-passing algorithms for inference over general graphs with convex free energy. In *UAI*, 2008.
- J. Kappes, B. Andres, F. Hamprecht, C. Schnoerr, S. Nowozin, D. Batra, S. Kim, B. Kausler, T. Kroeger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A comparative study of modern inference techniques for structured discrete energy minimization problems. *IJCV*, 2015.
- J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *STOC*, 1999.
- V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 2006.
- N. Komodakis, N. Paragios, and G. Tziritas. MRF optimization via dual decomposition: Message-passing revisited. In *ICCV*, 2007.
- N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal dual strategies. *CVIU*, 2008.
- A. Koster, C. van Hoesel, and A. Kolen. The partial constraint satisfaction problem: Facets and lifting theorems. *Operations Research Letters*, 1998.
- M. P. Kumar. Rounding-based moves for metric labeling. In *NIPS*, 2014.
- M. P. Kumar and D. Koller. MAP estimation of semi-metric MRFs via hierarchical graph cuts. In *UAI*, 2009.
- M. P. Kumar and P. Torr. Improved moves for truncated convex models. In *NIPS*, 2008.
- R. Manokaran, J. Naor, P. Raghavendra, and R. Schwartz. SDP gaps and UGC hardness for multiway cut, 0-extension and metric labeling. In *STOC*, 2008.
- P. Pansari and M. P. Kumar. Truncated max-of-convex models. *CoRR*, 2015.

- J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufman, 1998.
- H. Racke. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC*, 2008.
- P. Ravikumar, A. Agarwal, and M. Wainwright. Message-passing for graph-structured linear programs: Proximal projections, convergence and rounding schemes. In *ICML*, 2008.
- M. Schlesinger. Syntactic analysis of two-dimensional visual signals in noisy conditions. *Kibernetika*, 1976.
- 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 with smoothness-based priors. *PAMI*, 2008.
- D. Tarlow, D. Batra, P. Kohli, and V. Kolmogorov. Dynamic tree block coordinate ascent. In *ICML*, 2011.
- O. Veksler. *Efficient graph-based energy minimization methods in computer vision*. PhD thesis, Cornell University, 1999.
- O. Veksler. Graph cut based optimization for MRFs with truncated convex priors. In *CVPR*, 2007.
- M. Wainwright, T. Jaakkola, and A. Willsky. MAP estimation via agreement on trees: Message passing and linear programming. *Transactions on Information Theory*, 2005.
- C. Wang, N. Komodakis, and N. Paragios. Markov random field modeling, inference and learning in computer vision and image understanding: A survey. *CVIU*, 2013.
- Y. Weiss, C. Yanover, and T. Meltzer. MAP estimation, linear programming and belief propagation with convex free energies. In *UAI*, 2007.
- T. Werner. A linear programming approach to max-sum problem: A review. *PAMI*, 2007.
- T. Werner. Revisiting the linear programming relaxation approach to Gibbs energy minimization and weighted constraint satisfaction. *PAMI*, 2010.