Parallel Imaging Problem

Thành Nguyen[⋆] and Éva Tardos[⋆⋆]

Cornell University
{thanh,eva}@cs.cornell.edu

Abstract. Metric Labeling problems have been introduced as a model for understanding noisy data with pair-wise relations between the data points. One application of labeling problems with pair-wise relations is image understanding, where the underlying assumption is that physically close pixels are likely to belong to the same object.
In this paper we consider a variant of this problem, we will call Parallel Imaging, where instead of directly observing the noisy data, the data undergoes a simple linear transformation first, such as adding different images. This class of problems arises in a wide range of imaging problems. Our study has been motivated by the Parallel Imaging problem in Magnetic Resonance Image (MRI) reconstruction. We give a constant factor approximation algorithm for the case of speedup of two with the truncated linear metric, motivated by the MRI reconstruction problem. Our method uses local search and graph cut techniques.

1 Introduction

In this paper we propose a problem, we will call Parallel Imaging, that combines features of the Metric Labeling problem with linear algebra. Metric Labeling problems have been studied extensively [8, 2, 5, 6] with one of the primary applications for image processing. In the image processing application the data observed is associated with each pixel of an image (such as intensity, depth, etc). Given noisy data about the image we want to recover the most likely original data. Motivated by a model of images via Markov Random Field [9] this imaging problem is often solved via an energy minimization approach called the Metric Labeling problem that combines features of the assignment and graph partitioning problems. In this paper, we consider a variant of this problem where there are multiple independent data values associated with each node of the graph, and we are observing different linear combinations of the associated data at each node.
Our original motivation for studying this problem came from Magnetic Resonance Imaging (MRI), an important medical technology widely used for both clinical and research applications. For many medical applications its essential to reduce scan time mainly due to reducing motion artifacts.

[⋆] Supported in part by ONR grant N00014-98-1-0589 and ITR grant CCR-0325453.
[⋆⋆] Supported in part by ITR grant CCR-0325453, ONR grant N00014-98-1-0589 and NSF grant CCR-0729006.

To speed up the image acquisition MRI technology uses multiple parallel scans. Raj, Singh and Zabih [10] show how this parallel MRI imaging problem gives rise to the kind of the labeling problem discussed in this paper where the number of observations associated with each node is the speedup factor used in the process. Our main result is a constant factor approximation algorithm for the case of this problem speeding up image acquisition by a factor of two.

The Metric Labeling Problem. The Metric Labeling problem is defined as follows. We are given a graph G and a label set L. Our goal is to associate a label value $f(v) \in L$ with each node v of the graph G. There are two competing factors in defining the most appropriate label for a node. First, the label $f(v)$ should be identical, or at least similar to an observed data value $o(v)$. Second, neighboring nodes in the graph should obtain similar labels. These goals are expressed by an "energy" function of the form: Assignment Cost + Separation Cost =

$$\sum_{v \in V} A_v(f(v), o(v)) + \sum_{e=(u,v) \in E} w_e d(f(v), f(u)).$$

Here $A_v(\alpha, o(v))$, or sometimes we only write $A_v(\alpha)$, is an assignment cost of assigning label α to a node v with observed label $o(v)$. The assignment cost is trying to encourage a labeling that is close to the observation. The goal of the second term is to encourage neighboring nodes to have the same or similar labels. The function $d(\alpha, \beta)$ is the distance between labels, and we will assume that this distance is a metric, and the value w_e for an edge $e = (u, v)$ expresses the strength of the connection between u and v, the strength of the penalty for assigning u and v (very) different labels.

Image Reconstruction Problem. In imaging problems, an image is modeled as a graph whose vertices are the pixels of the image, and the neighbors of a vertex are either 4 or 8 pixels around. Labels can represent a variety of properties of the associated object, such as color, intensity (darkness), or distance from the camera. In this paper we consider black-and-white imaging problems, such as the images obtained by the MRI technology, and will use integers (intensity or darkness) as labels. For such problems, the most commonly used term for the assignment penalty function $A_v(.,.)$ is the square norm $(f(v) - o(v))^2$, where assignment cost is then

$$\sum_{v \in V} A_v(f(v), o(v)) = ||f - o||_2^2.$$

This assignment cost is motivated via the probabilistic roots of the labeling problem. If the observed labeling $o(v)$ is generated from a labeling $f(v)$ by adding Gaussian random noise, then the above assignment cost is proportional to the probability that $f(v)$ is the original labeling, assuming we see observation $o(v)$.

The separation term expresses the goal that most pairs of neighboring nodes should have same or similar labels. In the context of imagining, this is a reasonable expectation as most useful images have only a few objects (e.g., the organ that is subject of the MRI, and some of the neighboring organs), and pixels belonging to the same object or same body part typically have the same or similar labels. One simple option is to use the linear metric $d(u,v) = |f(v) - f(u)|$. However, a more robust metric is the *truncated linear metric* $d(\alpha, \beta) = \min(M, |\alpha - \beta|)$ where M is a parameter of the problem. Here, small changes in labeling come with a small penalty, as small change in label can reflect gradual changes in the object, but the bigger the change is, the more likely that we have an object boundary. However, once the difference of the assigned labels is large enough, it does suggest an object boundary independent of the actual size of the difference. Our goal is to only have few object boundaries. This suggests that after some difference, the penalty should remain constant and not grow further with the difference in labels. This robustness of the truncated linear metric helps in obtaining sharp object boundaries of the resulting images.

Our problem: Parallel Imaging Problem. In MRI, when an image is scanned with double speed, instead of getting the data on each pixel, we obtain a linear transform of the image. In particular, a receiver is observing an image whose width is equal to half of the the width of the original image, obtained by adding these two strips of the original image with some positive coefficients depending on the position of the receiver. See [10] for more details. Hence, in order to reconstruct an image, we need data on more than one receiver. Assume we have k receivers with the corresponding linear transform $H_1, .., H_k$ and the observed data vectors $o_1, .., o_k$.

Now, we model an image as two *identical* copies G_1, G_2 corresponding to the strips of the image. And we use the following convention if v_1 is a node in G_1 then v_2 is its copy in G_2 and vice versa. Let $f : V_1 \cup V_2 \to L$ be a labeling. Without noise, the linear transform H_j at receiver j adds the labeling of v_1 and v_2 with some positive coefficients h_{jv_1} and h_{jv_2} for all pairs of vertices to get $o_j(v) = h_{jv_1} f(v_1) + h_{jv_2} f(v_2)$.

Similar to the traditional Metric Labeling problem, our goal now is to find a labeling f to minimize the sum of assignment cost and separation cost. The assignment cost is : $\sum_j ||H_j f - o_j||^2$. And the separation cost is the sum of the separation cost of f in G_1 and G_2 as discussed above. More precisely, we need to find a labeling to minimize

$$\sum_j ||H_j f - o_j||^2 + \sum_{uv \in G_1 \cup G_2} w_{uv} d(f(v), f(v)).$$

We call this problem *Parallel Metric Labeling* if d is an arbitrary metric, and *Parallel Imaging* if d is a linear truncated metric. The main result of this paper is a constant approximation for the Parallel Imaging problem.

Note. We simplify the problem a bit by considering each “strip” of the original image as a separate graph G_i, that are not related to each other. This simplification ignores the edges connecting the strips, but only a very small fraction of the edges are ignored, so the method should still lead to good image quality.

Related works. In this paper, we consider a class of applications where data is observed via a linear transformation. Traditional reconstruction models for such data do not use priors. Given the data f, the observation is through a linear transformation Hf, where H is a non-negative matrix. The traditional reconstruction uses a least squares application to reconstruct the data: Given observed data o, the method looks for the labeling f that minimizes $||o - Hf||_2^2$. We consider a problem, which adds priors to the above least squares image reconstruction. This problem combines the combinatorial problem of Metric Labeling with added linear algebraic features. The problem was introduced by Raj, Singh and Zabih [10]. In this work they developed practical heuristics for this problem based on ideas from Metric Labeling.

The Metric Labeling problem was introduced in [8], and studied extensively [2, 5, 6, 8]. The problem is $\mathcal{NP}$-hard for general metric. The best known approximation algorithm for the problem is an $O(\log |L|)$ [8] and has no $\Omega(\sqrt{\log n})$ approximation unless $\mathcal{NP}$ has quasi-polynomial time algorithms [2]. Many special cases of the Metric Labeling problem have been considered: [3, 7, 4, 6], among which [3] and [6] are the closest to ours.

Boykov, O. Veksler, and R. Zabih [3] were the first to develop a local search technique, called α-extension, for the Metric Labeling problem. For linear truncated metrics, Gupta and Tardos [6] extend the "α-extension" local move to "interval" local move to obtain a constant approximation. These techniques are discussed in details in section 4. Our result is a nontrivial extension of [6]: our assignment cost depends on *pairs of nodes*. As will be shown, there is a method to reduce our problem to the traditional Metric Labeling on a different metric. However, this metric does not have an embedding with a constant distortion to tree metric as in the case of truncated one and therefore, one cannot get a constant approximation using this approach. In order to give a constant approximation, we need to develop a new graph construction for the new assignment cost, and extend the interval-local move introduced in [6].

Organization. The rest of our paper is organized as follows: In section 2 we show that our problem can be reduced to the traditional Metric Labeling, but this only gives a logarithmic approximation. In section 3, we give a graph construction for our new assignment cost. This is a basic step for us to develop a local search algorithm in section 4.

2 The Parallel Metric Labeling Problem

In this section, we show that the Parallel Metric Labeling problem can be thought of as a traditional labeling problem with a larger label set and a different metric. Viewing the Parallel Metric Labeling as a traditional labeling problem on a bigger label set immediately implies that known techniques for Metric Labeling to give a logarithmic approximation for the problem. However, using the larger label set does not allow us to take advantage of the special structure of the linear truncated metric needed for a constant factor approximation, but this way of thinking about the Parallel Labeling problem will still be useful in the subsequent sections on defining our combinatorial algorithm (based on graph cuts and local search).

Now, if instead of considering a labeling as a function on the vertices of G_1 and G_2, we consider it as a mapping from the vertices of a *single* copy to pairs of labels. The new set of labels is now the set of pairs of numbers. The assignment cost depending on these two numbers is now a function of a new label. Given two vertices u an v with the labeling (α_1, β_1) and (α_2, β_2), the separation cost is

$$d_2((\alpha_1, \alpha_2), (\beta_1, \beta_2)) = d(\alpha_1, \beta_1) + d(\alpha_2, \beta_2).$$

Now, if d is a metric on L then d_2 defined as above is also a metric on L^2. Therefore, given a Parallel Metric Labeling problem with a metric d, we can understand the problem as an instance of a traditional Metric Labeling problem with the metric d_2 on a larger set of labels. Using the approximation algorithm for the Metric Labeling problem in [8], we have the following result:

Theorem 1. *There is a $O(\log |L|)$ approximation for the Parallel Metric Labeling problem with any metric d. When d is a truncated linear metric, which is the Parallel Imaging problem, we get a $O(\log M)$ approximation, where M is the truncation parameter.*

Proof. Due to [8], there exists an algorithm for the Metric Labeling problem that approximates the best solution up to a factor of $O(distort)$, where *distort* is the distortion of a probabilistic embedding of the metric in to random trees.

Any metric on K vertices can be probabilistically embedded into to random trees with distortion $\log K$. This gives the $O(\log |L|^2) = O(\log |L|)$ approximation. It is not hard to see that the 2 dimensional truncated linear metric can be embedded into random trees with distortion of $O(\log M)$, which proves the theorem. $\square$

3 The Problem with Linear Metric is Solvable

We now consider the Parallel Metric Labeling problem where the metric is the linear metric, i.e., $d(\alpha, \beta) = |\alpha - \beta|$. We will give a graph cut construction that allows us to get the optimal labeling.

Ideas of using graph cut to find a best labeling have been used in Boykov et al. [3], Ishikawa and Geiger [7], and Gupta and Tardos [6] for the traditional labeling problem. These approaches use the following basic construction:

The basic construction. The basic idea is to build a chain for each vertex as follows. For each vertex v, build a chain $v^1, v^2, ..., v^L$ of length L representing the possible labels that the vertex can be assigned to. We also add a super source s and a super sink t to the network as shown

in Figure 1(a). Consider the following correspondence between (s,t)-cuts and labelings:
A labeling $f: V \to L$ assigns v to α, $(f(v) = \alpha)$ if and only if the edge $v^\alpha v^{\alpha+1}$ is in the corresponding cut. Here we use the notation that $v^{L+1} = t$ for all nodes v. To make sure that the mapping between the labelings and the graph cuts is well defined, we need to guarantee that for every node v there is exactly one edge of its corresponding chain in the cut. To do this, we add the edges $v^{\alpha+1}v^\alpha$ with infinite capacity. Further, we do not want to cut edge (s, v^1), so we also assign an infinite capacity to this edge in both directions. Now we have a one-to-one mapping between *finite* cuts and all the possible labelings.

New assignment cost. In the traditional Metric Labeling problem, assignment cost is a function of the label of one vertex: $A_v(\alpha)$. This can be captured easily by assigning a capacity of $A_v(\alpha)$ to the edges $(v^\alpha, v^{\alpha+1})$ [3, 7]. Now, recall from section 2, our new assignment cost is a function on *pairs* of labels: If v_1 is labeled α; v_2 is labeled β, i.e., $f(v) = (\alpha, \beta)$, then the assignment cost is

$$A_v(f(v)) = \sum_{j=1}^{k} ||H_j f(v) - o_j(v)||^2 = \sum_{j=1}^{k} (h_{jv_1}\alpha + h_{jv_2}\beta - o_j(v))^2. \quad (1)$$

This assignment cost cannot be separated into terms that depends on α and β separately, as there is the term $2\sum_{j=1}^{k} h_{jv_1}h_{jv_2}\alpha\beta = c_v\alpha\beta$. Note that $c_v > 0$ by the assumption that all the matrices H_j are nonnegative. The new observation is that we can modify the chain construction above to capture this assignment cost: For node $v_1 \in G_1$ we connect the chain of nodes $v_1^1, \dots v_1^L$ as described by the above construction. However, for node $v_2 \in G_2$ we connect the chain backwards, reversing the rolls of s and t in the construction for the copy G_2 as shown in Figure 1(b). If a cut separates nodes v_i^α and $v_i^{\alpha+1}$, then the corresponding labeling assigns v_i to label α. Note that the chain for v_2 starts its numbering from the t side, and increases towards the s side, and we are using the notation that $v_1^{L+1} = t$ and $v_2^{L+1} = s$ for all nodes v. To model the assignment cost, we now add a complete graph between the chains of v_1 and v_2 with capacity $c_v/2$ on each edge. The cut corresponding to labeling $f(v_1) = \alpha$ and $f(v_2) = \beta$ cuts $\alpha\beta + (L-\alpha)(L-\beta)$ of these edges, so these edges contribute a total of $\frac{c_v}{2}(2\alpha\beta - L\alpha - L\beta + L^2)$ to the capacity of the cut. After adding the above edges between the pairs of chains, the remaining terms in the assignment cost can be written as the sum of two functions depending on α and β separately. Any such assignment cost can be captured exactly via the capacity of the edges in the two chains. However, these costs may now be negative. Consider a chain, say corresponding to node v_i that has some of the resulting edge capacity negative. To make all capacities nonnegative, we will add the same positive number to every edge on the chain. This change adds a constant to the capacity of all cuts, as each finite cut uses exactly one edge in every chain.

Separation cost. As shown in [3, 7], the technique of building chains can also help us to capture some classes of separation cost function. For simplicity we consider the case of linear separation cost $d(\alpha, \beta) = |\alpha - \beta|$. In this case, consider an edge of the original graph $e = (u, v)$, we know that the chains corresponding to u, v are in the same order, because they are both in the same graph G_1 or G_2. We add edges $u^\alpha v^\alpha$, $\forall \alpha \in L$ with the capacity of w_e. Note that if a cut uses $u^\alpha u^{\alpha+1}$ and $v^\beta v^{\beta+1}$, then these edges contribute to the cut a total capacity of $w_e|\alpha - \beta|$, which is exactly the separation cost of the corresponding labeling. See Figure 1(c).

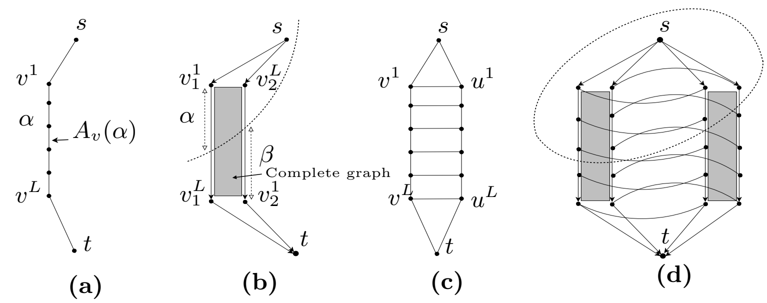

Fig. 1. Constructions for the problem with linear metric.

We combine the constructions described above as shown on Figure 1(d). We use the construction for new assignment cost and the construction of for separation cost to get the following theorem:

Theorem 2. *The Parallel Metric Labeling problem with linear metric and can be solved in polynomial time via a graph construction such that the minimum cut of this graph is equal to the minimum cost of the labeling plus a constant.* □

4 Constant Approximation via Local Search

In the previous section, we give a new graph construction whose minimum cut gives the optimal solution of the Parallel Metric Labeling problem with linear metric. The problem is, however, $\mathcal{NP}$ hard for the truncated linear metric. Local search has been proved to be a successful method for this type of problems. Our algorithm also uses this approach. First let us explain some of previous works that use the local search method for the traditional Metric Labeling problem.

Local Search. For uniform metric d, ($d(i,j) = 1$ iff $i \neq j$), Boykov et al [3] give a local search algorithm that tries to relabel some vertices to a label α. Such best local move can be found via graph cut algorithm. The graph construction is shown in Figure 2 (a): For each edge (u, v) we build a node p_{uv} and connect it with u, v and the super sink s. The weight of sp_{uv}, up_{uv}, vp_{uv} are $d(f(u), f(v)), d(f(v), \alpha), d(f(u), \alpha)$ respectively,

where $f(u), f(v)$ are the current labeling of u and v. The edges su, sv have weights of the current assignment cost of u and v, while tu, tv have weights of the assignment cost of u and v if assigned to the new label α. Given a cut X such that $s \in X$, in its corresponding relabeling, a vertex u is relabeled iff $u \in X$. The main observation is the following: Because d is a metric, the weights of sp_{uv}, up_{uv}, vp_{uv} satisfy the triangle inequality. Thus, any minimum cut separating s and t would cut at most 1 edge among these three edges. And therefore a minimum cut captures exactly the cost of the corresponding relabeling. As a result, a best local move can be found by a single minimum cut algorithm.
For truncated metric with the truncated parameter M, Gupta and Tardos [6] extended the local move above to an "interval" local move, where at every step they pick a random interval I of length M and try to reduce the cost of a current labeling by changing some vertices to labels in I. They construct a graph to find a local move. See Figure 2 (b). It is shown that although a best labeling could not be found, approximate optimal local moves are enough for a constant approximation. The idea of the analysis is analogous to that of Boykov et. al. For a locally optimal labeling, consider a random interval I and use the fact that relabeling all nodes that have labels $\alpha \in I$ in the optimum, does not yield a cut with improved cost. The main new observation is that for a random interval, the probability that there is an error in the term associated with an edge $e = uv$ in the cut construction is $1/M$ times the distance $d(f^*(u), f^*(v))$ between the labels assigned to its nodes in the optimum. This fact is used to show that the expected error is proportional to the optimal separation cost.

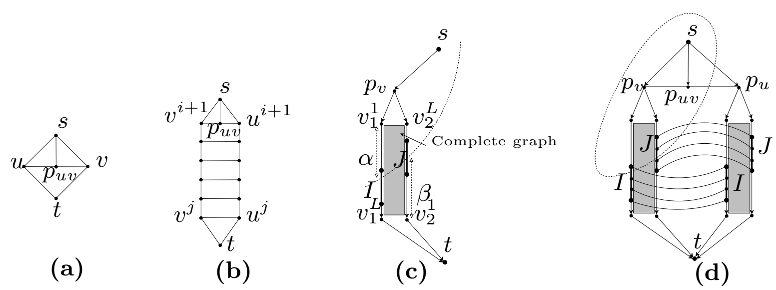

Fig. 2. Constructions for the local search technique.

Our local move and the construction for a local move

Our approach is similar to these results. There are, however, some difficulties that we need to overcome. First, our assignment cost depends on pairs of nodes. To solve this we use the idea developed in the previous section to construct a new graph whose cuts approximately capture the cost of relabeling. Second, we need to modify the "interval" local move in [6] for our problem. To see why a naive extension of the interval local move does not work, consider a local move that picks a random interval I from $1, .., L$ and finds an approximately good relabeling to this interval, i.e., changing some labels to labels in I. This will not give us a provable approximation algorithm. For example, when reassigning a node in Metric Labeling, we capture the assignment cost exactly, while here a node v in G needs two labels: one of v_1 and one for v_2, and maybe only one of the two is in the interval I. To capture the two different labels for a node v, we do the following:

> Pick two random intervals I, J of length at most M each. We will allow in one local move to relabel pairs of vertices (v_1, v_2) where v_1, v_2 are copies of vertex v in the original graph G, to a new pair of labels $(\alpha, \beta) \in I \times J$.

To describe the construction let us fix intervals I, J. Our main goal is to give a construction where local moves correspond to cuts and the capacity of each cut has only a small additive error compared to the corresponding local move. The construction is shown on Figures 2 (c) and (d). For vertex v, we add two chains $\{v_1^1, .., v_1^L\}$ and $\{v_2^L, .., v_2^1\}$, as shown on Figure 2(c), to the graph $G_{I\times J}$. We also add vertex p_v connecting to v_1^1 and v_2^L. Connect the super source s to p_v with the capacity $A_v(f(v))$, where $f(v)$ is the label currently assigned to vertex v, and $A_v(f(v))$ is the corresponding assignment cost. Connect v_1^L and v_2^1 to the super sink t. We use the notation: $v_1^{L+1} = t, v_2^{L+1} = p_v$.
We would like to have the following one-to-one correspondence between cuts and labelings. The cut C uses the edge sp_v if and only if v retains its label $f(v)$ and uses some pair of edges $v_1^\alpha v_1^{\alpha+1}$ and $v_2^{\beta+1} v_2^\beta$, where $(\alpha, \beta) \in I \times J$, if v_1 and v_2 is reassigned to labels (α, β) in the corresponding relabeling. To make sure exactly one of the conditions above occurs, we need to assign infinite capacity to some other edges. This step is similar to the constructions above. We now need to add more edges to capture the assignment cost and separation cost.

Assignment cost. This construction is similar to the construction of the linear case in the previous section. We want to capture the assignment cost of v as given in the formula (1): We add a complete bipartite graph between $\{v_1^1, .., v_1^L\}$ and $\{v_2^L, .., v_2^1\}$ as shown on Figure 2(c). The edges go both ways between v_1^α and v_2^β with capacity $\frac{c_v}{2}$. The contribution of these edges to the cut corresponding to the relabeling to (α, β) is $\frac{c_v}{2}(\alpha\beta + (L-\alpha)(L-\beta)) = c_v\alpha\beta + \frac{c_v}{2}(L^2 - L\alpha - L\beta)$. The remaining terms in the assignment cost depend on α and β separately, so we can use the edges of the chain to capture the assignment cost. To make sure all capacities are nonnegative, we may have to add a constant to the cost.

Separation cost. For each edge in the original graph $e = (u, v)$ let us consider the following construction shown on Figure 2(d): Add edges between the chains of u and v: $u_1^\alpha v_1^\alpha$, $u_2^\beta v_2^\beta$ for all $\alpha \in I, \beta \in J$. These edges are bidirectional and have a capacity of w_e. We also add one more vertex p_{uv}, and connect p_{uv} with p_u, p_v and s in both directions. The capacity of the edge sp_{uv} is $w_e d_2(f(u), f(v))$, (that is f's separation cost on the edge e); of $p_{uv}p_v$ $p_{uv}p_u$ are $2Mw_e$. These three numbers satisfy the triangle inequality. Therefore, at most one of the three edges adjacent to p_{uv} is cut. This implies that for a minimum cut C and for each edge (u, v) one of the followings could happen.

- All of p_{uv}, p_u and p_v are on the s side of the cut. In the corresponding relabeling both u and v get new labels. In this case, the cut perfectly captures the separation cost.
- s is on one side and p_u, p_v, p_{uv} are on the other side. In the corresponding relabeling the label of u and v do not change, and the capacity of the edge sp_{uv} captures exactly the separation cost.
- p_{uv} is on the s side, and so is exactly one of p_u or p_v, say p_v. See Figure 2(d). In the corresponding relabeling, u keeps its original label, while v gets a new label. For this case, the cut does not capture exactly the separation cost. However, it is not hard to see that the capacity exceeds the separation cost, and the total corresponding capacity is $2w_eM$ from edge (p_{uv}, p_u) and at most $w_e(2M)$ from the edges between the chains, while the cost of the corresponding labeling is $w_e d(f'(v), f(u))$, where $f'(v)$ is the new label that v gets.

We have seen that the construction above does not capture exactly the cost of a local move. For an edge $e = (u, v)$, when one of the nodes (say u) retains its label and v is labeled with some element $f'(v) \in I \times J$. In this case the separation cost is always greater or equal than the real separation cost of e. And this value is at most $4Mw_e$. We summarize this in the following theorem:

Theorem 3. *Given a labeling f, and $I \times J \subset L^2$ where I and J are of size at most M, then there exists a network $G_{I\times J}$ and a constant $CONST$ with the following properties. All labelings g that can be reached by relabeling to $I \times J$ correspond to cuts in $G_{I\times J}$ with capacity at least the $Q(g) + CONST$. If g is a labeling obtained by a local move, then the capacity of the corresponding cut overestimates $Q(g) + CONST$ by replacing the separation cost of $w_e d(f'(u), f'(v))$ for edges $e = (u, v)$ where exactly one end receives a label in $I \times J$ by a possibly larger term which is at most $4w_eM$. And finally, the minimum $s - t$ cut in $G_{I\times J}$ corresponds to a labeling.* □

The algorithm and its analysis

We use the above construction in a local search algorithm. At each step we pick two intervals I, J randomly according to Definition 1 below and find the minimum cut in the graph $G_{I\times J}$. If the corresponding labeling has a lower cost, then change the labeling to this new one. The algorithm stops if we cannot find an improving move.

Definition 1. *Given a parameter D, a random partitioning process of the grid L^2 into smaller grids according to D is defined as follow. Pick $r = (\alpha, \beta) \in \{1, .., D\}^2$ uniform-randomly, for each $k \equiv \alpha \pmod D$, and $l \equiv \beta \pmod D$, delete all edges in the row k, and column l of the grid. The L^2 grid now is partitioned into smaller parts, each of them is a grid* *of the form $I \times J$ for some intervals I, J. We call this set of small grids $Partition(r)$. A random rectangle is understood as a random grid taken uniformly from a random $Partition(r)$.*

We prove the following main theorem:

Theorem 4. *Any local optima f of the local search algorithm above with $D = M$ has cost $Q(f)$ that is at most constant times the optimal cost: $9Q(f^*)$. For any constant $\epsilon > 0$, a solution with cost at most $(9+\epsilon)Q(f^*)$ can be reached in polynomial time.*

Proof sketch We first need the following notation. For any subset $X \subset V$, let $A^*(X)$ and $A(X)$ be the optimum and the current labeling assignment cost of the vertices in X. For any subset of edges $Y \subset E$, let $S^*(Y), S(Y)$ be the separation cost for those edges paid by the optimum and the current labeling respectively. Clearly, $Q(f) = A(V) + S(E), Q(f^*) = A^*(V) + S^*(E)$. Consider the case where the algorithm chooses a random rectangle $I \times J$. Let V_{IJ} be the set of nodes in G to which f^* assigns labels from $I \times J$. Let E_{IJ} be the set of edges in G such that both ends are assigned to $I \times J$ in f^*. Let σ_{IJ} be the set of edge in G such that exactly one end is assigned to $I \times J$ in f^*. For a random partition r, let σ_r be the edges in E_{IJ} for all rectangles $I \times J$ in $Partition(r)$, we have: $\sigma_r = \frac{1}{2}\sum_{IJ \in Partition(r)} \sigma_{IJ}$.
Given two intervals I, J, define the following local move.

$$f'(v) = \begin{cases} f^*(v) & \text{if } f^*(v) \in I \times J \\ f(v) & \text{otherwise.} \end{cases}$$

The 9-approximation guarantee follows from comparing $Q(f)$ and $Q(f')$ for every I, J. If we could find a minimum local move exactly then the local optimality of f would imply the inequality : $Q(f) \leq Q(f')$ for every I, J. However, the graph cut construction only gives us an approximate minimum local move (Theorem 3). As a result, we only have a weaker inequality. For every I, J the cost $Q(f)$ is no more than the cost of the labeling corresponding to the minimum cut in $G_{I \times J}$. The cost of all cuts corresponding to local moves have the same additive constant. In addition, the cost $Q(f')$ of the local move f' can have an additive cost at most $4M\sum_{e\in\sigma_{IJ}} w_e - S'(\sigma_{IJ})$, where $S'(\sigma_{IJ})$ denotes the separation cost of f' on the set of edges σ_{IJ}. The labeling f is locally optimal, and hence with this additive cost the local move f' does not correspond to a smaller capacity cut. This gives us the following inequality

$$Q(f) \leq Q(f') + 4M \sum_{e\in\sigma_{IJ}} w_e - S'(\sigma_{IJ}).$$

Expressing the terms contributing to $Q(f)$ and $Q(f')$, and deleting terms that contribute to both we get the following:

$$A(V_{IJ}) + S(E_{IJ}) + S(\sigma_{IJ}) \leq A^*(V_{IJ}) + S^*(E_{IJ}) + 4M \sum_{e\in\sigma_{IJ}} w_e.$$

Summing this inequality over all the rectangles in $Partition(r)$, we get the following, as the edges in σ_{IJ} each contribute to two interval pairs.

$$Q(f) + S(\sigma_r) \leq A^*(V) + S^*(E - \sigma_r) + 8M \sum_{e\in\sigma_r} w_e.$$

$$\Rightarrow Q(f) + S(\sigma_r) \leq A^*(V) + S^*(E) + 8M \sum_{e \in \sigma_r} w_e.$$

Now, we need to bound the error term $8M \sum_{e \in \sigma_r} w_e$. This value depends on the partition. Recall that we picked $r \in \{1, .., M\}^2$ at random. We will bound the expected value of this term. What is the probability that an edge $e = (u, v)$ is in σ_r? This probability depends on how far the coordinates of $f^*(u)$,$f^*(v)$ are from each other. More precisely, if $f^*(u) = (\alpha_1, \beta_1), f^*(v) = (\alpha_2, \beta_2)$, then $(u, v) \in \sigma(r)$ if either α_1, α_2 or β_1, β_2 are separated by a random border as described in Definition 1. The probability of this event is at most:

$$\frac{\min\{M, |\alpha_1 - \alpha_2|\}}{M} + \frac{\min\{M, |\beta_1 - \beta_2|\}}{M} = \frac{d(f^*(u), f^*(v))}{M} = \frac{d_e}{M}.$$

Thus, one has the expected value of $8M \sum_{e \in \sigma_r} w_e$ is at most $8S^*(E) = 8 \sum_{e \in \sigma_r} w_e d_e$. Taking the expected value of the inequality above we have:

$$Q(f) \leq Q(f) + S(\sigma_r) \leq A^*(V) + S^*(E) + 8S^*(E) \leq 9Q(f^*),$$

which proves that a local optimum is a 9-approximate solution.
Taking only big enough improvements, the algorithm will find in polynomial time a solution that is within an $9 + \epsilon$ factor to the optimum value. □

Acknowledgments. We thank Ramin Zabih for introducing this problem, and Gurmeet Singh and Ashish Goel for numerous discussions.

References

1. A. Archer J. Fakcharoenphol, C. Harrelson, R. Krauthgamer, K. Talvar and E. Tardos: *Approximate Classification via Earthmover Metrics.* Proc. SODA 2004.
2. J. Chuzhoy and S. Naor: *The Hardness of Metric Labeling.* Proc. FOCS 2004.
3. Y. Boykov, O. Veksler, and R. Zabih: *Markov Random Fields with efficient approximations.* Proc. CVPR 1998.
4. Y. Boykov, O. Veksler, and R. Zabih: *Fast approximate energy minimization via graph cuts.* Proc. ICCV 1999.
5. C. Chekuri, S. Khanna, S. Naor, and L. Zosin: *A Linear Programming Formulation and Approximation Algorithms for the Metric Labeling Problem.* SIAM J. on Discrete Mathematics, 606-635, 2004.
6. A. Gupta and E. Tardos: *A Constant Factor Approximation Algorithm for a Class of Classification Problems.* Proc. STOC 2000.
7. Hiroshi Ishikawa and Davi Geiger: *Segmentation by grouping junctions.* Proc. CVPR 1998.
8. J. Kleinberg and E. Tardos: *Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Partitioning and Markov Random Fields.* J.ACM 49(5): 616-639 (2002).
9. S. Li: *Makrov Random Field modeling in Computer Vision.* Springer-Verlag, 1995.
10. A. Raj, G. Singh and R. Zabih: *MRF's for MRI's: Bayesian Reconstruction of MR Images via Graph Cuts.* Proc. CVPR 2006.