Energy Minimization Under Constraints on Label Counts

Yongsub Lim
KAIST
yongsub@kaist.ac.kr

Kyomin Jung
KAIST
kyomin@kaist.edu

Pushmeet Kohli
Microsoft Research
pkohli@microsoft.com

Abstract. Many computer vision problems such as object segmentation or reconstruction can be formulated in terms of labeling a set of pixels or voxels. In certain scenarios, we may know the number of pixels or voxels which can be assigned to a particular label. For instance, in the reconstruction problem, we may know size of the object to be reconstructed. Such label count constraints are extremely powerful and have recently been shown to result in good solutions for many vision problems.
Traditional energy minimization algorithms used in vision cannot handle label count constraints. This paper proposes a novel algorithm for minimizing energy functions under constraints on the number of variables which can be assigned to a particular label. Our algorithm is deterministic in nature and outputs ε-approximate solutions for all possible counts of labels. We also develop a variant of the above algorithm which is much faster, produces solutions under almost all label count constraints, and can be applied to all submodular quadratic pseudo-boolean functions. We evaluate the algorithm on the two-label (foreground/background) image segmentation problem and compare its performance with the state-of-the-art parametric maximum flow and max-sum diffusion based algorithms. Experimental results show that our method is practical and is able to generate impressive segmentation results in reasonable time.

1 Introduction

Algorithms for energy minimization have become an indispensable tool in computer vision. These algorithms enable inference of the Maximum a Posteriori (MAP) solutions of labelling problems such as image segmentation, optical flow, and stereo. Due to its wide applicability, the energy minimization problem has received a lot of interest from both the theoretical computer science [1] and machine learning communities [2–5].

Most energy minimization methods used in computer vision such as Max-product Belief Propagation (BP) [6, 7], Tree Reweighted message passing (TRW) [8], and Graph Cuts [9] operate on energy functions defined over discrete variables. They consider unconstrained minimization of the energy function over the discrete domain of random variables, and do not allow any direct way of enforcing constraints on the solutions. In contrast, this paper studies the problem of constrained energy minimization. Specifically, we address the problem of minimizing energy functions under the constraint that the number of variables assigned on any given label is equal to some known constant. This constrained minimization problem is useful for many labelling problems in computer vision. For instance, in the image segmentation problem, it enables us to obtain a segmentation of any desired size.

Related Work During the past few years, researchers have considered practical optimization problems under relevant constraints. One of the most effective examples is the case of 3D reconstruction, where the silhouette constraint was introduced [10, 11]. This constraint ensured that a ray emanating from any silhouette pixel must pass through one voxel which belongs to the 'object'. It was proven to be an effective replacement for the ballooning term [12] and led to improved results.

Segment *connectivity* is another example of an equally powerful but much more sophisticated constraint that was introduced for the two label (foreground/background) segmentation problem. The energy function corresponding to the segmentation problem is composed of unary and pairwise potential functions [13] and is well known to be *submodular*. This property allows the energy function to be minimized in polynomial time using efficient maximum flow based algorithms. The connectivity constraint enforces that all variables that have been assigned in the foreground label form one connected component. Vicente *et al.* [14] showed that enforcing connectivity while minimizing the submodular segmentation energy makes the problem NP-hard.

Constraints on Label Counts Minimization under the so-called *label counting* constraints is not new to the computer science community. Unconstrained energy minimization was studied in theoretical computer science in the context of Metric labelling. This is the problem of minimizing an energy function where the pairwise potential functions are defined in terms of the weighted uniform distance function that is defined on the label set $[k]$. Recently, Naor and Schwartz [15] obtained an approximation algorithm for a constrained version of this problem, which they called the *balanced metric labeling problem*.

Balanced labeling means that the number of variables assigned to any particular label is at most ℓ. They obtain an $O\left(\frac{\log n}{\varepsilon}\right)$-approximation randomized algorithm that runs in polynomial time over n and $\frac{1}{\varepsilon}$. The algorithm guarantees that at most $O\left(\log k \frac{1+\varepsilon}{1-\varepsilon}\right) \cdot \ell$ many variables in the final solution are assigned to each label. The Naor-Schwartz algorithm works for any underlying graph G, but it cannot be applied for general fixed label counting constraints. Due to randomness of the assignment, the counting guarantee from their method is still far from the exact counting constraint we want to achieve. Furthermore, the approximation ratio between its answer and the optimal solution is not small enough to be useful in practice.

Counting Constraints in Computer Vision Werner [16] were one of the first to introduce constraints on label counts in energy minimization. They proposed a n-ary max-sum diffusion algoithm for solving these problems, and demonstrated its performance on the binary image denoising problem. However, their algorithm could only produce solutions or some label counts. It was not able to guarantee an output for any arbitrary label count desired by the user.

A number of other recent vision papers have also demonstrated how knowledge about label counts can be used as a useful prior. For instance, Woodford *et al.* [17] recently showed how potentials for enforcing a particular distribution in label counts can be used to improve results in labelling problems such as image denoising and texture synthesis. They proposed a number of sophisticated algorithms which were able

to minimize energy functions containing higher order potentials encouraging particular counts of labels. However, their algorithms were not able to enforce label counts as a hard constraint, and also lacked any worst-case bounds on the quality of the obtained solution.

The method most closely related to ours is that of Kolmogorov *et al.* [18]. They showed that for submodular energy functions, the parametric maxflow algorithm [19] can be used for energy minimization with label counting constraints. However, this algorithm outputs optimal solutions for only some label counts, and is not guaranteed to output solutions for any arbitrary count of labels.

Our Results We propose a new method for performing energy minimization under constraints on the label counts. Our algorithm is deterministic in nature and outputs ε-approximate solutions in a grid graph with N vertices. For all possible labels, the algorithm runs in $O\left(Nk^{\frac{1}{\varepsilon}}\left(\frac{1}{\varepsilon}\right)^{2k+2} + N^k\left(\frac{1}{\varepsilon}\right)^2\right)$ time, where k is the number of labels and N is the number of pixels. This algorithm can also minimize energy functions containing potentials depending on label counts such as the ones use in [17] as it outputs the minimum energy solution under all possible label counting constraints.

We also develop a variant of the above algorithm which is much faster and produces solutions under almost all label count constraints, but can only be applied to submodular quadratic pseudoboolean functions. We call this algorithm *decomposed parametric*. It is inspired from the parametric maxflow based method for obtaining solutions under label counts. As mentioned earlier, the vanilla parametric maxflow method finds optimal solutions for a small number of label counts. We propose a new algorithm which dramatically increases the number of label counts for which a solution can be found. We first decompose the original image into a number of subimages. The vanilla parametric maxflow algorithm is run on each subimage. In this way, we obtain a set of assignments for each sub-image with minimum energy under some label counts. These sets are merged to obtain the set of assignments for the whole image with minimum energy under all label count constraints. Experiments on the binary image segmentation problem show that our method dramatically outperforms the standard parametric maxflow and max-sum diffusion based methods for obtaining solutions under label count constraints.

1.1 Organization

Remainder of the paper is organized as follows. We define the problem setup and provide some preliminaries in section 2. In section 3, we state our main theorem about the multiplicative error bound guaranteed by our approach. Our parametric maxflow based algorithm is explained in section 4. In section 5, we provide the results of our experiments on the image segmentation problem, and compare them with those obtained using state-of-the-art methods. We conclude by discussing ideas for future work in section 6.

2 Preliminaries and Setup

Energy Minimization Many labeling problems in computer vision can be formulated using energy minimization. Energy functions are defined on a pixel-grid graph $G = (V, E)$,

and have the form

$$H(\mathbf{x}) = \sum_{v \in V} \phi_v(x_v) + \sum_{(v,w) \in E} \phi_{vw}(x_v, x_w), \tag{1}$$

where $\phi_{vw} : [k]^2 \to \mathbb{R}^+ \triangleq \{x \in \mathbb{R} : x \geq 0\}$ and $\phi_v : [k] \to \mathbb{R}^+$ are assumed to be arbitrary non-negative real-valued functions defined over variables taking values from the label set $[k] = \{1, \cdots, k\}$.

We use the positivity of ϕ_v's and ϕ_{vw}'s in the proof of our multiplicative approximation guarantee. However, our algorithm can also be applied to energy functions with negative ϕ_{vw} values in the same manner.

In this paper, we are interested in finding an assignment $\mathbf{x}$ minimizing H under a label count defined as follows.

Definition 1 (label count). *For an assignment $\mathbf{x} \in [k]^N$ and $j \in [k]$, define count$(\mathbf{x}, j)$ to be the number of variables $x_v : v \in V$ such that $x_v = j$. Let $\mathcal{C}(N)$ be the collection of $C = (C_1, C_2, \ldots, C_k) \in {\mathbb{Z}_+}^k$ such that $C_1 + C_2 + \ldots + C_k = N$. We call $C \in \mathcal{C}(N)$* a label count. *For $C \in \mathcal{C}(N)$, let*

$$\mathcal{R}(C) = \{\mathbf{x} \in [k]^N | \forall j \in [k],\ \textit{count}(\mathbf{x}, j) = C_j\}.$$

Our problem is to find such an assignment $\mathbf{x}^*(C)$ that minimizes the energy function $H(\mathbf{x})$ among $\mathbf{x} \in \mathcal{R}(C)$. The problem of finding an assignment that minimizes energy for fixed label count even for a submodular $H(\mathbf{x})$ is known to be NP-hard [20]. Hence we consider the following approximation problem.

Definition 2 (ε-approximation). *Let $0 < \varepsilon < 1$. An assignment $\hat{\mathbf{x}}$ is called ε-approximation of the energy with the label count C, if $\hat{\mathbf{x}} \in \mathcal{R}(C)$ and*

$$(1 - \varepsilon) H(\hat{\mathbf{x}}) \leq H(\mathbf{x}^*(C)) \leq H(\hat{\mathbf{x}}).$$

Definition 3 (submodular function). *A pseudoboolean function $g(x_1, x_2) : \{0, 1\}^2 \to \mathcal{R}$ is submodular if the following holds.*

$$g(0, 0) + g(1, 1) \leq g(0, 1) + g(1, 0).$$

An energy function is called submodular if all its pairwise terms are submodular. If H is a submodular, an assignment with the minimum energy can be computed efficiently by the graph-cut algorithm [21]. Submodular energy functions are widely used for labeling problems in computer vision.

Parametric maxflow Parametric maxflow algorithm [18] is known that it gives some $\mathbf{x}$'s minimizing H under some label counts and can be applied when x_v's are in $\{0, 1\}$ and H is submodular [22]. It deals with parameterized energy function rather than the original one.

Parametric maxflow

Let $G = (V, E)$ be an undirected graph. Parametric maxflow is to minimize energy function $H^\lambda(\mathbf{x})$ for parameter $\lambda \in I$ in the interval $I \in \mathbb{R}$ where

$$H^\lambda(\mathbf{x}) = H(\mathbf{x}) + \lambda \sum_{v \in V} x_v. \tag{2}$$

Lemma 1. *If an assignment* $\mathbf{x}$ *minimizes the energy function* H^λ *for some* λ*,* $H(\mathbf{x})$ *is minimum under the same label count as* $\mathbf{x}$*.*

Proof. Assume that $\mathbf{x}$ does not minimize H under the label count. Let $\mathbf{x}'$ minimize H under the same label count as $\mathbf{x}$. Because $\mathbf{x}$ and $\mathbf{x}'$ have same number of 1's, they also have the same second term in (2), therefore $H^\lambda(\mathbf{x}) > H^\lambda(\mathbf{x}')$. It contradicts that $\mathbf{x}$ minimizes H^λ.

For a fixed λ, note that H^λ is also submodular, hence (2) can be solved in polynomial time using the graph-cut algorithm. Because $F(\lambda) = \min_{\mathbf{x}}(H^\lambda(\mathbf{x}))$ is a piecewise-linear concave function of λ, it is enough to compute $\mathbf{x}$'s at all breakpoints of F rather than for every λ, where a breakpoint is the intersection point of two line segments of $F(\lambda)$. The following algorithm finds $\mathbf{x}$'s at each breakpoint and they are minimum of H under the label counts as that of $\mathbf{x}$.

Parametric maxflow algorithm

– Input : Energy function H.
1. Let $I = [\lambda_{min}, \lambda_{max}]$.
2. Compute $\mathbf{x}_{min}$ and $\mathbf{x}_{max}$ solutions for λ_{min} and λ_{max}, respectively.
3. **if** $\mathbf{x}_{min} = \mathbf{x}_{max}$
4. **then** Initialize L as $(\mathbf{x}_{min}, [\lambda_{min}, \lambda_{max}])$.
5. **else** Initialize L as $(\mathbf{x}_{min}, \{\lambda_{min}\})$, $(\mathbf{x}_{max}, \{\lambda_{max}\})$.
6. **while** there are adjacent items $(\mathbf{x}_i, I_i)$, $(\mathbf{x}_j, I_j)$ such that $\sup I_i < \inf I_j$
7. $\lambda_i = \sup I_i$, $\lambda_j = \inf I_j$.
8. Compute λ^*, a solution of $H^\lambda(\mathbf{x}_i) = H^\lambda(\mathbf{x}_j)$.
9. **if** $\lambda_i = \lambda^*$ **then** $I_j = I_j \cup [\lambda_i, \lambda_j]$.
10. **else if** $\lambda_j = \lambda^*$ **then** $I_i = I_i \cup [\lambda_i, \lambda_j]$.
11. **else** λ^* must be in (λ_i, λ_j).
12. Compute $\mathbf{x}^*$ minimizing $H^{\lambda^*}(\mathbf{x})$.
13. **if** $\mathbf{x}^* = \mathbf{x}_i$ **or** $\mathbf{x}^* = \mathbf{x}_j$
14. **then** $I_i = I_i \cup [\lambda_i, \lambda^*]$, $I_j = I_j \cup [\lambda^*, \lambda_j]$.
15. **else** $(\mathbf{x}, \{\lambda^*\})$ is inserted to L between $(\mathbf{x}_i, I_i)$ and $(\mathbf{x}_j, I_j)$

– Output : list L of pairs $(\mathbf{x}_i, I_i)$ where $H^\lambda(\mathbf{x}_i)$ is minimum for $\lambda \in I_i$.

This algorithm uses graph-cut algorithm at most $(2B + 2)$ many times where B is the number of breakpoints. In the worst case, there are at most $|V| + 1$ breakpoints since there is at most one break point for each label count. In the rest of this paper, we will call this parametric maxflow algorithm as *the pure parametric* algorithm (PP), to compare it with our new algorithms.

3 Approximation Algorithm for Energy Minimization

In this section, we provide an algorithm that is guaranteed to compute an ε-approximate solutions for all label counts in $O(|V|)$ time. Our algorithm can be generalized to more general class of graphs including 8-connected grid graph and planar graphs, by designing a collection of graph decompositions satisfying the properties of Lemma 2. For example, when the graph is a planar graph, the collection of decompositions in [23] satisfies the properties. In this paper, we will prove our result for grid graph for easy of explanation. Let G be the grid graph of size $n \times n$. Our algorithm is based on a decomposition of G into small components; computing an array of solutions in each of these components, then producing a global solution. The algorithm works by exploiting the sparseness of the graph G. It reduces the original problem with large tree-width to a number of smaller problems with low tree width. In this process, it inserts a error in the estimation. We have shown that this error is small because the graph has limited connectivity.

Theorem 1. *There is a deterministic algorithm that outputs ε-approximate solutions for all $C \in \mathcal{C}(N)$, which runs in time $O\left(Nk^{\frac{1}{\varepsilon}}\left(\frac{1}{\varepsilon}\right)^{2k+2} + N^k\left(\frac{1}{\varepsilon}\right)^2\right)$.*

We will call our algorithm DD (decomposed dynamic), since its main procedure is based on graph decompositions and dynamic programming on each graph components. A brief sketch of DD is as follows. First, we will obtain a family $\mathcal{D}$ of graph decompositions of G. It satisfies that each decomposition $D \in \mathcal{D}$ is obtained by removing some edges of G, and $|\mathcal{D}| = \frac{1}{\varepsilon^2}$. The following is a pseudo-code of our graph decomposition.

Graph Decomposition

- Inputs : $G = (V, E)$, $\varepsilon > 0$, and $k_1, k_2 \in \{0, 1, 2, \ldots, \frac{1}{\varepsilon}\}$.

1. Remove all the edges of G that connects vertices of the form (a, b) and $(a+1, b)$ where $a \equiv k_1 \pmod{\frac{1}{\varepsilon}}$.
2. Remove all the edges of G that connects vertices of the form (a, b) and $(a, b+1)$ where $b \equiv k_2 \pmod{\frac{1}{\varepsilon}}$.
3. The remaining graph is decomposed into connected components.

- Output : G.

Let $\mathcal{D}$ be the collection of the decompositions computed for all $k_1, k_2 \in \{0, 1, 2, \ldots, \frac{1}{\varepsilon}\}$.

Lemma 2. *$\mathcal{D}$ satisfies the following properties.*

(1) For all $D \in \mathcal{D}$, the size of each connected component of D is at most $\frac{1}{\varepsilon^2}$.

(2) For all edge e of G, the number of decompositions in $\mathcal{D}$ that remove e is at most $\varepsilon|\mathcal{D}|$.

Now, fix $D \in \mathcal{D}$. Let $R_1, R_2, \ldots, R_\ell$ be the connected components of D. For each $R_i = (V_i, E_i)$, DD computes the following g_i values for all $C_{(i)} = (C_{i1}, C_{i2}, \ldots, C_{ik}) \in \mathcal{C}(|V_i|)$ by dynamic programming on R_i.

$$g_i(C_{(i)}) = \min_{\mathbf{x} \in \mathcal{R}(C_{(i)})} \left[\sum_{v \in V_i} \phi_v(x_v) + \sum_{(v,w) \in E_i} \phi_{vw}(x_v, x_w) \right].$$

Note that the tree width of the subgraph R_i is at most $\frac{1}{\varepsilon}$. The description of computation of g_i for all $C_{(i)} \in \mathcal{C}(|V_i|)$ by dynamic programming is in Appendix II in the supplementary material.

Computation of each g_i for all $C_{(i)} \in \mathcal{C}(|V_i|)$ takes $O\left(k^{\frac{1}{\varepsilon}}\left(\frac{1}{\varepsilon}\right)^{2k}\right)$ time. Since there are at most $O(N)$ many R_i's, its total computation time for a fixed $D \in \mathcal{D}$ is $O\left(Nk^{\frac{1}{\varepsilon}}\left(\frac{1}{\varepsilon}\right)^{2k}\right)$.

Now, Let $E_D \subset E$ be the union of all the edges of R_i's. Then for each $C \in \mathcal{C}(N)$, we compute

$$g_D(C) = \min_{\mathbf{x} \in \mathcal{R}(C)} \left[\sum_{v \in V} \phi_v(x_v) + \sum_{(v,w) \in E_D} \phi_{vw}(x_v, x_w) \right],$$

using $g_i(\cdot)'s$. This can be done in time $O(N^k)$ by the merging process described below.

Merging

- We begin with the regions $R_1, R_2 \ldots R_\ell$, and their corresponding functions g_i's.
- Repeat the following process until there remains just one region.
  - If there are more than one regions, choose any two of them, say R_a and R_b. Let g_a and g_b be their corresponding functions.
  - Let $R_c = R_a \cup R_b$ in the sense of graph union. I.e, for $R_a = (V_a, E_a)$ and $R_b = (V_b, E_b)$, let $R_c = (V_c, E_c)$ with $V_c = V_a \cup V_b$ and $E_c = E_a \cup E_b$.
  - For each $C_{(c)} \in \mathcal{C}(|V_c|)$, let

$$g_c(C_{(c)}) = \min \left[g_a(C_{(a)}) + g_b(C_{(b)}) \right], \tag{3}$$

    where the minimization is over all $C_{(a)} \in \mathcal{C}(|V_a|)$, $C_{(b)} \in \mathcal{C}(|V_b|)$ such that $C_{ai} + C_{bi} = C_{ci}$ for all $1 \leq i \leq k$.
  - Replace the two regions R_a and R_b by R_c. Also replace g_a and g_b by g_c.
- Output the resulting function for the entire graph.

In the above process, under the assumption that for all $C_{(a)} \in \mathcal{C}(|V_a|)$,

$$g_a(C_{(a)}) = \min_{\mathbf{x} \in \mathcal{R}(C_{(a)})} \left[\sum_{v \in V_a} \phi_v(x_v) + \sum_{(v,w) \in E_a} \phi_{vw}(x_v, x_w) \right],$$

and that for all $(C_{(b)}) \in \mathcal{C}(|V_b|)$,

$$g_b(C_{(b)}) = \min_{\mathbf{x} \in \mathcal{R}(C_{(b)})} \left[\sum_{v \in V_b} \phi_v(x_v) + \sum_{(v,w) \in E_b} \phi_{vw}(x_v, x_w) \right],$$

it is straightforward to see that for all $C_{(c)}$,

$$g_c(C_{(c)}) = \min_{\mathbf{x} \in \mathcal{R}(C_{(c)})} \left[\sum_{v \in V_c} \phi_v(x_v) + \sum_{(v,w) \in E_c} \phi_{vw}(x_v, x_w) \right].$$

Hence by induction, we obtain that the above procedure outputs $g_D(\cdot)$, and its total running time is $O(N^k)$.

Let $\hat{\mathbf{x}}_D(C)$ be the assignment corresponding to $g_D(C)$. Then let

$$\hat{\mathbf{x}}(C) = \text{argmin}_{D \in \mathcal{D}} \, H(\hat{\mathbf{x}}_D(C))$$

be the output of DD for $C \in \mathcal{C}(N)$. Approximation proof of DD is in Appendix I.

4 Decomposed Parametric

Although DD is guaranteed to output ε-approximation for all label counts, it runs slowly when the decomposed subimage size becomes large. In this section we provide a more practical algorithm that works for any submodular pseudoboolean energy function.

Our new algorithm, DP (decomposed parametric) runs on a decomposed image like in DD. The basic idea is to apply the parametric maxflow to each decomposed subimage rather than the dynamic programming. While DD outputs optimal assignments under all label counts in each subimage, DP outputs some of those assignments. However, by merging the partially optimal results of parametric maxflow, we obtain assignments under almost all label counts. Although our algorithm can be applied to an image of arbitrary size, to make explanation easy, we assume a square size image in this section. Note that since DP uses the parametric maxflow on each subimage, it is only applicable to binary labeling with submodular while DD can be applied to general labeling with any energy function.

DP decomposes a given $n \times n$ image to $\lceil \frac{n}{m} \rceil^2$ many subimages of size $m \times m$. Let I_{ij} be the subimage $[(i-1) \times m + 1, i \times m] \times [(j-1) \times m + 1, j \times m]$ and H_{ij} be the energy function H restricted to I_{ij} where $1 \leq i, j \leq \lceil \frac{n}{m} \rceil$. Figure 1 depicts this decomposition. We apply parametric maxflow algorithm to every subimage I_{ij} to compute assignments minimizing H_{ij} under some label counts. Here we assume that the output of the pure parametric is a form of an array A such that $A[k]$ is an assignment having label count k. Then arrays of size $m^2 + 1$ are created as results of the parametric maxflow algorithm on each subimage, and we obtain an array about the whole image by merging those arrays.

Decomposed Parametric

– Inputs : an image I of size $n \times n$, an integer m.
1. $A = \emptyset$.
2. Decompose I to subimages of size $m \times m$.
3. **for** i 1 to $\lceil n/m \rceil$
4. **for** j 1 to $\lceil n/m \rceil$
5. A_{ij} = *Parametric maxflow*(H_{ij}).
6. A = *Merge*(A, A_{ij}).

– Output : A.

Merge

– Inputs : two arrays A_1 and A_2
1. Let n_1 and n_2 be the size of A_1 and A_2, respectively.
2. Let A be a new array of size $n_1 + n_2 - 1$.
3. Every element of A is set to *empty* assignment.
4. **for** every $(i, j) \in \{0, \cdots, n_1 - 1\} \times \{0, \cdots, n_2 - 1\}$
5. **if** both $A_1[i]$ and $A_2[j]$ are not *empty* assignment
5. $\mathbf{x} = A_1[i]$ concatenated by $A_2[j]$.
6. **if** $H(\mathbf{x}) < H(A[i + j])$
7. $A[i + j] = \mathbf{x}$

– Output : A.

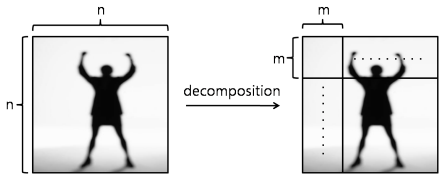

Fig. 1. Decomposition of a $n \times n$ image to subimages of size $m \times m$.

When two arrays are merged, we get $\ell_1 \times \ell_2$ new assignments where ℓ_1 is the number of assignments in the first array and ℓ_2 is that in the second array. Although there are some overlap of label counts, the new array has bigger proportion of *nonempty* assignments than the two merged arrays. We observe that when m is about $\frac{n}{3}$, for almost all label counts DP outputs assignments.

5 Experimental Results

We did experiments to compare our two algorithms with PP. To that end, we measured three values: the number of label counts for which the methods returned a solution, the energy values for the computed assignments, and the running time.

For our image segmentation experiments, we considered the energy function H defined in (1). The potential functions ϕ_i of H are obtained using the method described in [24] which exploits user given hints about the appearance of foreground and background segments. The pairwise potentials defined over edges connected in a 4-neighbourhood systems are defined as

$$\phi_{ij} = |x_i - x_j|(\lambda_1 + \lambda_2 \times g(i, j)),$$

where λ_1 and λ_2 are parameters of the model, and $g(i, j)$ is proportional to the distance of i and j's RGB colors. We have conducted experiments for various values of λ_1 and λ_2. In our experiments, we use DP_k and DD_k to denote DP and DD , respectively, with the decomposition of an image into $k \times k$ number of subimages. The parameter values used in the experiments are specified by $\mu = \frac{\lambda_2}{\lambda_1}$.

Experiment 1 Our first experiment compares the average number of assignments minimizing the energy function H under some label counts, and the average energy values of DP with optimal solutions obtained by PP. We used 8 images from [25] for computing the average each of which was a 300×300 size, and simulated DP_k, $1 \leq k \leq 5$, and PP for all images. For each algorithm, we did tests for $\lambda_1 = 2^i$, $0 \leq i \leq 7$, and $\mu = 10, 20$ and 30.

Experimental results are shown in Table 1 and Figure 2. Table 1 shows the average ratio of the number of computed label counts over N, the number of pixels of the image. DP_3 outputs assignments under almost all label counts while PP about 20% of the label counts. DP_5 outputs assignments under label counts more than 99% regardless of λ_1 and λ_2.

	$\mu = 20$		
λ_1	PP	DP_3	DP_5
1	0.2786	0.9828	0.9998
2	0.2552	0.9819	0.9997
4	0.2231	0.9795	0.9995
8	0.1862	0.9767	0.9994
16	0.1498	0.9736	0.9986
32	0.1164	0.9698	0.9970
64	0.0875	0.9650	0.9951
128	0.0642	0.9544	0.9925

Table 1. Comparison of the number of label counts for the experiment 1. Each value is the average over 8 images. The ratio of the number of label counts for which each algorithm computes a solution over the number of possible label counts. For $\mu = 10$ and $\mu = 30$, the results are almost the same as those with $\mu = 20$. DP_3 outputs solutions for almost all label counts regardless of μ.

Figure 2 shows the average energy value ratio of each simulation compared to that of the optimal solutions obtained by PP . In Figure 2, the energy value of DP over the optimal energy tends to have bigger error as λ_1 or λ_2 increases, or the image is decomposed to more subimages. But in almost all cases, the error is quite small, especially when $\lambda_1 = 8$, which is typically used in the image segmentation, the error is less than 0.5% regardless of λ_2. In short, we obtain the results that DP_3 is enough to obtain assignments under almost all label counts with only small error.

Table 2 shows the running time of PP and DPs. Note that DP is quite fast and its running time is competative to that of PP .

Examples of segmented images of the algorithms, PP and DP, are shown in Figure 3. Note that the output image of DP_3 is almost the same to that of PP. By comparing with the ground truth, we observe that DP_3 is nearly optimal.

We note that adding a constant to the energy function affects the approximation ratio of the results. In our simulation, we adjusted the constant term for each vertex v so that one of $\phi_v(0)$ and $\phi_v(1)$ becomes 0.

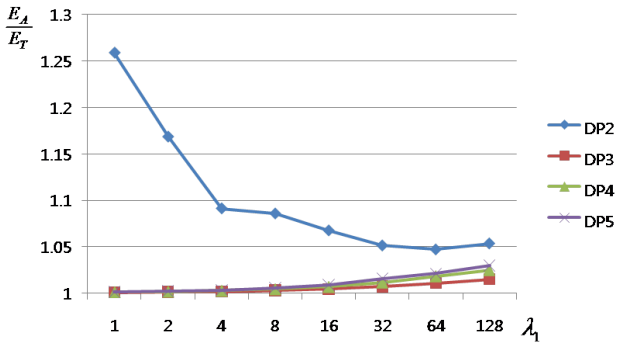

Fig. 2. Comparison of the average energy value for the experiment 1. We computed the average of $\frac{E_A}{E_T}$ over the computed label counts where E_A is the average energy value for 8 images of DP , and E_T is the average energy value of PP for 8 images. μ did not affect the values, so in this graph the values with $\mu = 20$ are shown.

	Time(seconds)				
	PP	DP_2	DP_3	DP_4	DP_5
IM1	8	10	18	24	27
IM2	11	23	31	42	53
IM3	51	22	28	35	42
IM4	4	11	15	18	19
IM5	23	42	48	63	75
IM6	12	38	42	52	62
IM7	26	44	55	76	94
IM8	17	42	46	58	70

Table 2. Table for running time of the algorithms for 8 images.

(a) original (b) ground truth (c) PP (d) DP_3

Fig. 3. (b) is the ground truth for segmentation, (c) is the output of PP for a label count close to that of the ground truth, and (d) is the output of DPs with similar label count.

Experiment 2 The second setup is for comparing the average energy values of DD, DP and PP. This experiment used 8 images each of which was a 200×200 size image, and simulated DP_{20} and DD_{20}. λ_1 and μ were the same as in the setup 1.

Table 3 shows the average ratio of the energy values of DP_{20} and DD_{20} over the optimal solutions obtained by PP. It is similar with Figure 2 to increase the energy value as λ_1 or μ increases. Note that DP_{20} outputs almost the same energy ratio values with DD which is optimal in each decomposed subimages. Sometimes DP is even slightly better. This is because although DD is actually optimal in each subimage, after merging, it is not guaranteed to have lower energy than the output of the DP on the whole image.

	$\mu = 10$		$\mu = 20$		$\mu = 30$	
λ_1	DP_{20}	DD_{20}	DP_{20}	DD_{20}	DP_{20}	DD_{20}
1	1.0061	1.0079	1.0080	1.0104	1.0100	1.0127
2	1.0086	1.0109	1.0111	1.0140	1.0136	1.0172
4	1.0129	1.0157	1.0163	1.0199	1.0194	1.0235
8	1.0197	1.0232	1.0249	1.0295	1.0302	1.0359
16	1.0308	1.0360	1.0388	1.0456	1.0472	1.0558
32	1.0481	1.0492	1.0592	1.0611	1.0711	1.0739
64	1.0706	1.0713	1.0862	1.0877	1.1020	1.1044
128	1.1008	1.1021	1.1228	1.1253	1.1447	1.1484

Table 3. Comparison of the average energy value for the experiment 2. We computed the values in the same way with Figure 2

Table 4 shows the running time of DD_{20} and DD_{25} for each image. Although DD is guaranteed to output approximate solutions, its running time is comparably longer than that of DP.

	Time	
	DD_{20}	DD_{25}
IM1	24m 41s	17m
IM2	32m 33s	29m 21s
IM3	24m 16s	16m 21s
IM4	25m 17s	17m 43s
IM5	27m 39s	21m 14s
IM6	26m 26s	19m 14s
IM7	33m 26s	25m 15s
IM8	27m 12s	22m 6s

Table 4. Table for running time of DD_{20} and DD_{25} for 8 images.

From experiment 1 and 2, we can conclude that for binary labeling with submodular energy function, DP_3 performs best among PP , DP_i, DD when considering the number of label counts, accuracy, and the running time altogether.

Experiment 3 In this experiment, we compare our algorithm with the Werner's max-sum diffusion algorithm [16] on the binary image denoising problem. A binary image corrupted with Gaussian noise of size 150×150 is used. DP_3 with D_3, $\lambda_1 = 8$ and $\lambda_2 = 160$ and Werner's were simulated. Figure 4 shows the original image and two resulting images with the same label count. It can be seen that the DP method produced assignments for many more label count constraints (21568) while still remaining close to the ground truth result. In contrast, Werner's max-sum diffusion algorithm was only able to find solutions for 12 label counts.

Fig. 4. (a) Original image of size 150×150 corrupted with Gaussian noise. (b) Result of DP with D_3 under label count 7058 among 21568 many outputs. (c) Result of Werner's max-sum diffusion algorithm under the same label count among 12 many outputs.

6 Discussion and Future Work

We have proposed novel algorithms for minimizing energy functions under label counting constraints. We first provided an efficient algorithm that outputs ε-approximate solutions for all possible counts of labels for any energy function. We also developed a variant of this algorithm which can be applied to submodular energy functions, that is much faster but misses solutions corresponding to some label counts.

In this paper, we have only considered the counting constraint defined on *vertices*. Another important counting constraint problem is that of *edge counting*, ie. the number of times we see a discontinuity in the labelling which is exactly equal to the boundary length in the case of image segmentation.

Consider the energy minimization with constraint on the number of *boundary edges* (edges having different labels on its two end vertices). This problem corresponds to segmentation with fixed boundary length. Note that this problem can be partially solved by considering the following parametric maxflow:

$$H^{\lambda}(\mathbf{x}) = H(\mathbf{x}) + \lambda \sum_{(v,w)\in E} x_v x_w,$$

where $\lambda \leq 0$ using the same reasoning as vertex counting. However, the algorithm is partial, since it cannot work for $\lambda \geq 0$ where the energy become non-submodular. As a future work, we will work on analysis of this algorithm.

References

1. Chekuri, C., Khanna, S., Naor, J., Zosin, L.: A linear programming formulation and approximation algorithms for the metric labelling problem. SIAM Journal on Discrete Mathematics (2005)
2. Komodakis, N., Tziritas, G., Paragios, N.: Fast, approximately optimal solutions for single and dynamic MRFs. In: CVPR. (2007)
3. Kumar, M.P., Koller, D.: MAP estimation of semi-metric MRFs via hierarchical graph cuts. In: UAI. (2009)
4. Sontag, D., Meltzer, T., Globerson, A., Jaakkola, T., Weiss, Y.: Tightening LP relaxations for MAP using message passing. In: UAI. (2008)
5. Werner, T.: A linear programming approach to max-sum problem: A review. PAMI (2007)
6. Weiss, Y., Yanover, C., Meltzer, T.: MAP estimation, linear programming and belief propagation with convex free energies. In: UAI. (2007)
7. Yedidia, J., Freeman, W., Weiss, Y.: Generalized belief propagation. In: NIPS. (2001)
8. Wainwright, M., Jaakkola, T., Willsky, A.: MAP estimation via agreement on trees: message-passing and linear programming. IEEE Transactions on Information Theory (2005)
9. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. PAMI (2001)
10. Kolev, K., Cremers, D.: Integration of multiview stereo and silhouettes via convex functionals on convex domains. In: ECCV. (2008)
11. Sinha, S., Pollefeys, M.: Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In: ICCV. (2005)
12. Vogiatzis, G., Torr, P., Cipolla, R.: Multi-view stereo via volumetric graph-cuts. In: CVPR. (2005)
13. Boykov, Y., Jolly, M.: Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In: ICCV. (2001)
14. Vicente, S., Kolmogorov, V., Rother, C.: Graph cut based image segmentation with connectivity priors. In: CVPR. (2008)
15. Naor, J., Schwartz, R.: Balanced metric labeling. In: STOC. (2005)
16. Werner, T.: High-arity interactions, polyhedral relaxations, and cutting plane algorithm for soft contraint optimisation (map-mrf). In: CVPR. (2008)
17. Woodford, O., Rother, C., Kolmogorov, V.: A global perspective on MAP inference for low-level vision. In: ICCV. (2009)
18. Kolmogorov, V., Boykov, Y., Rother, C.: Application of parametric maxflow in computer vision. In: ICCV. (2007)
19. Gallo, G., Grigoriadis, M., Tarjan, R.: A fast parametric maximum flow algorithm and applications. SIAM J. on Comput. **18** (1989) 18:30–55
20. Garey, M., Johnson, D.S.: Computers and intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
21. Goldberg, A., Tarjan, R.: A new approach to the maximum-flow problem. Journal of the Association for Computing Machinery (1988)
22. Kohli, P.: Minimizing dynamic and higher order energy functions using graph cuts. (2007)
23. Jung, K., Shah, D.: Local algorithms for approximate inference in minor-excluded graphs. In: NIPS. (2007)
24. Blake, A., Rother, C., Brown, M., Pérez, P., Torr, P.: Interactive image segmentation using an adaptive gmmrf model. In: ECCV. (2004)
25. Rhemann, C., Rother, C., Rav-Acha, A., Sharp, T.: High resolution matting via interactive trimap segmentation. In: CVPR. (2008)

Appendix I

Let $C \in \mathcal{C}(N)$ be fixed, and let

$$\mathbf{x}^* = \mathrm{argmin}_{\mathbf{x} \in \mathcal{R}(C)} H(\mathbf{x}),$$

be the optimal solution, and let $\hat{\mathbf{x}} = \hat{\mathbf{x}}(C)$ be the output of DD for C. For each $D \in \mathcal{D}$, denote $\hat{\mathbf{x}}_D = \hat{\mathbf{x}}_D(C)$. From the positivity of ϕ_{vw}, for all $D \in \mathcal{D}$,

$$\sum_{v \in V} \phi_v(x^*_v) + \sum_{(v,w) \in E_D} \phi_{vw}(x^*_v, x^*_w) \leq H(x^*). \tag{4}$$

From the minimality of $\hat{\mathbf{x}}_D$ in each connected components of D, for all $D \in \mathcal{D}$,

$$\begin{aligned} &\sum_{v \in V} \phi_v((\hat{\mathbf{x}}_D)_v) + \sum_{(v,w) \in E_D} \phi_{vw}((\hat{\mathbf{x}}_D)_v, (\hat{\mathbf{x}}_D)_w) \\ &\leq \sum_{v \in V} \phi_v(x^*_v) + \sum_{(v,w) \in E_D} \phi_{vw}(x^*_v, x^*_w). \end{aligned} \tag{5}$$

From (4), (5) and the definition of $\hat{\mathbf{x}}$,

$$\begin{aligned} &\sum_{D \in \mathcal{D}} \left[\sum_{v \in V} \phi_v(\hat{\mathbf{x}}_v) + \sum_{(v,w) \in E_D} \phi_{vw}(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_w) \right] \\ &\leq \sum_{D \in \mathcal{D}} \left[\sum_{v \in V} \phi_v((\hat{\mathbf{x}}_D)_v) + \sum_{(v,w) \in E_D} \phi_{vw}((\hat{\mathbf{x}}_D)_v, (\hat{\mathbf{x}}_D)_w) \right] \\ &\leq \sum_{D \in \mathcal{D}} \left[\sum_{v \in V} \phi_v(x^*_v) + \sum_{(v,w) \in E_D} \phi_{vw}(x^*_v, x^*_w) \right] \\ &\leq |\mathcal{D}| H(x^*). \end{aligned} \tag{6}$$

By the property (2) of Lemma 2, i.e. from the property that for each edge e of E, the number of decompositions in $\mathcal{D}$ that removes e is at most $\varepsilon|\mathcal{D}|$, we obtain that

$$\begin{aligned} &(1-\varepsilon)|\mathcal{D}| H(\hat{\mathbf{x}}) \\ &\leq \sum_{D \in \mathcal{D}} \left[\sum_{v \in V} \phi_v(\hat{\mathbf{x}}_v) + \sum_{(v,w) \in E_D} \phi_{vw}(\hat{\mathbf{x}}_v, \hat{\mathbf{x}}_w) \right] \end{aligned} \tag{7}$$

From (6) and (7), we have

$$(1-\varepsilon) H(\hat{\mathbf{x}}_D) \leq H(\mathbf{x}^*).$$

Appendix II

Computing g_i

- Let $V_i = \{(a,b) | a,b \in \{1,2,\ldots \frac{1}{\varepsilon}\}\}$ be the set of vertices of R_i.
- Order the elements of V_i by dictionary order, i.e., $(a_1,b_1) < (a_2,b_2)$ if $a_1 < a_2$ or, $a_1 = a_2$ and $b_1 < b_2$. Let $v_1, v_2, \ldots, v_{\frac{1}{\varepsilon^2}}$ be the vertices in that order.
- For $t = 0, 1, \ldots, \left(|V_i| - \frac{1}{\varepsilon}\right) = t^*$, let
  $B_t = \left\{v_{t+1}, \ldots, v_{t+\frac{1}{\varepsilon}}\right\}$.
  Let $V_{it} = \{v \in V_i \mid$ order of v is less than or equal to some vertex in $B_t\}$. Let E_{it} be the set of edges that connect two vertices of V_{it}.
- For each assignment $\hat{\mathbf{x}}^{B_t} \in [k]^{|B_t|}$ over B_t, and each $C_{(t)} = (C_{t1}, C_{t2}, \ldots, C_{tk}) \in \mathcal{C}(|V_t|)$, let
  $$\mathcal{R}(\hat{\mathbf{x}}^{B_t}, C_{(t)}) = \mathcal{R}(C_{(t)}) \bigcap \left\{\mathbf{x} \in [k]^{|V_{it}|} \mid \mathbf{x}_v = \hat{\mathbf{x}}_v^{B_t} \; \forall \; v \in B_t\right\}.$$
  We will compute the following for $t = 0, 1, \ldots, t^*$).
  $$\hat{g}_t(\hat{\mathbf{x}}^{B_t}, C_{(t)}) = \min_{\mathbf{x} \in \mathcal{R}(\hat{\mathbf{x}}^B, C_{(t)})} \left[\sum_{v \in V_{it}} \phi_v(\mathbf{x}_v) + \sum_{(v,w) \in E_{it}} \phi_{vw}(\mathbf{x}_v, \mathbf{x}_w)\right].$$
- For $t = 0$, note that $V_{i0} = B_0$. Hence we directly compute $\hat{g}_0(\hat{\mathbf{x}}^{B_0}, C_{(0)})$ for all $\hat{\mathbf{x}}^{B_0} \in [k]^{|B_0|}$ and $C_{(0)} \in \mathcal{C}(|V_{i0}|)$.
- For $t = 1, 2 \ldots t^*$,
  - For each t, let $B'_t = B_t \bigcup B_{t-1}$. For each $\hat{\mathbf{x}}^{B'_t} \in [k]^{|B'_t|}$, and $C_{(t)} \in \mathcal{C}(|V_t|)$ let
    $$\mathcal{R}(\hat{\mathbf{x}}^{B'_t}, C_{(t)}) = \mathcal{R}(C_{(t)}) \bigcap \left\{\mathbf{x} \in [k]^{|V_{it}|} \mid \mathbf{x}_v = \hat{\mathbf{x}}_v^{B'_t} \; \forall \; v \in B'_t\right\},$$
    and compute
    $$\hat{g}'_t(\hat{\mathbf{x}}^{B'_t}, C_{(t)}) = \min_{\mathbf{x} \in \mathcal{R}(\hat{\mathbf{x}}^{B'_t}, C_{(t)})} \left[\sum_{v \in V_{it}} \phi_v(\mathbf{x}_v) + \sum_{(v,w) \in E_{it}} \phi_{vw}(\mathbf{x}_v, \mathbf{x}_w)\right]$$
    by the relation
    $$\begin{aligned}\hat{g}'_t(\hat{\mathbf{x}}^{B'_t}, C_{(t)}) = \hat{g}_{t-1}&\left(\left(\hat{\mathbf{x}}^{B'_t}\right)_{B_{t-1}}, C'_{(t)}\right) \\ &+ \phi_{v_{t+\frac{1}{\varepsilon}}}\left(\left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}}}\right) \\ &+ \phi_{v_{t+\frac{1}{\varepsilon}}, v_{t+\frac{1}{\varepsilon}-1}}\left(\left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}}}, \left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}-1}}\right) \\ &+ \phi_{v_{t+\frac{1}{\varepsilon}}, v_t}\left(\left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}}}, \left(\hat{\mathbf{x}}^{B'_t}\right)_{v_t}\right),\end{aligned}$$

where $C'_{tj} = C_{tj} - 1$ for $j \in [k]$ such that $\left(\hat{\mathbf{x}}^{B'_t}\right)_{v_{t+\frac{1}{\varepsilon}}} = j$, and $C'_{tj} = C_{tj}$ for other j's. In the above computation, when there is no edge between $v_{t+\frac{1}{\varepsilon}}$ and $v_{t+\frac{1}{\varepsilon}-1}$, the term $\phi_{v_{t+\frac{1}{\varepsilon}}, v_{t+\frac{1}{\varepsilon}-1}}$ is not computed.

- For $\hat{\mathbf{x}}^{B_t} \in [k]^{|B_t|}$ and $C_{(t)} \in \mathcal{C}(|V_t|)$, compute

$$\hat{g}_t(\hat{\mathbf{x}}^{B_t}, C_{(t)}) = \min_{j \in [k]} \hat{g}'_t(\left(j, \hat{\mathbf{x}}^{B_t}\right), C_{(t)}).$$

– For each $C_{(i)} \in \mathcal{C}(|V_i|)$, output

$$g_i(C_{(i)}) = \min_{\hat{\mathbf{x}}^{B_{t^*}} \in [k]^{|B_{t^*}|}} \hat{g}_{t^*}(\hat{\mathbf{x}}^{B_{t^*}}, C_{(i)}).$$