Constrained Discrete Optimization via Dual Space Search

Yongsub Lim
KAIST
yongsub@kaist.ac.kr

Kyomin Jung
KAIST
kyomin@kaist.edu

Pushmeet Kohli
Microsoft Research Cambridge
pkohli@microsoft.com

Abstract

This paper proposes a novel algorithm for solving NP-hard constrained discrete minimization problems whose unconstrained versions are solvable in polynomial time such as constrained submodular function minimization. Applications of our algorithm include constrained MAP inference in Markov Random Fields, and energy minimization in various computer vision problems. Our algorithm assumes the existence of a polynomial time oracle for computing the Lagrangian dual of the constrained optimization problem. One of the key properties of our algorithm is its ability to compute minimizers for several different constraint instances simultaneously. We show that our algorithm isolates all the constraint instances for which strong duality holds, and provides a lower bound for any specific constraint instance. We also developed a variant of the algorithm that is able to efficiently compute a lower bound for a specific constraint instance using a cutting plane scheme. We demonstrated the efficacy of our approach by showing how it can be applied to the image segmentation problem in computer vision.

1 Introduction

Constrained discrete optimization is a key tool in various fields, including machine learning and computer vision [5, 8]. In contrast to problems defined over continuous spaces such as linear programming for which many efficient polynomial time methods have been developed [7, 1], there are few known efficient methods for exact constrained discrete optimization, where even with a simple objective function and constraints the problem is usually NP-hard.

Unconstrained discrete optimization problems such as submodular function minimization (SFM) have been extensively studied in the Operations Research literature. Many important problems in machine learning can be formulated as SFM [3, 12]. For instance, Maximum a Posterior (MAP) inference in many Markov Random Fields (MRF) models in computer vision can be performed by unconstrained minimization of submodular functions [11].

In various real world problems, some prior knowledge about the statistic of the desired solution are available. For instance, in the case of the foreground-background image segmentation problem, we may know the exact shape and size of the object being segmented, and thus want to be able to find the most probable solution that has a particular area (number of foreground pixels) and boundary length (number of discontinuities). In such cases, we need to be able to make solutions from the random field model which are consistent with this prior knowledge about the statistics of the solution.

In this paper, we tackle the problem of minimizing an objective function defined over a discrete space under multiple constraints. Solving this problem provides a way to deal with problems whose unconstrained variants are polynomial time solvable, but their constrained versions are NP-hard. One of the most popular approaches for such constrained discrete optimization is the convex relaxation with rounding [2, 8]. Although it efficiently computes a rounded solution from the relaxed problem for multiple linear constraints, it cannot deal with non-linear equality constraints in general since the feasible region may become non-convex. On the other hand, our algorithm can deal with a larger class of constraints such as non-linear equality, and directly solve the discrete problem rather than continuous relaxation and rounding. Using dual space search, our algorithm computes minimizers for different constraint instances simultaneously, leading to isolating all the constraint instances for which strong duality holds. We also propose a variant for maximizing the dual, providing a lower bound of the constrained problem with any specific constrained instance. The algorithm only requires a polynomial time oracle to compute the Lagrangian dual of the primal problem for any point. Applications of our algorithm include constrained versions of many fundamental problems such as the shortest path, st-mincut, and minimum spanning tree.

Related work A number of methods have been tried to obtain better labeling solutions by inferring the MAP solution from a restricted domain of solutions which satisfy some constraints. Among them, solutions to image labeling problems which have a particular distribution of labels [13] has been widely studied. More specifically, for the problem of foreground/background image segmentation, the most probable segmentation under the 'label count' constraint, *i.e.* a specific number of pixels take the foreground label, have been shown to be closer to the ground truth [10, 12]. Another example is the silhouette constraint which has been used for the problem of 3D reconstruction [9].

The work most closely related to ours is the parametric mincut algorithm for constrained submodular function minimization [12]. This method can deal with the st-mincut problem under a constraint $\sum_i x_i = k$, which means that it finds a mincut in which exact k nodes belong to one part of the cut. It computes several solutions for different k values efficiently, and can partly handle an inequality constraint of the form $\sum_i x_i \leq k$. Further, the parametric mincuts method shows generally how a problem with one constraint can be solved. Although many studies have considered constraints when optimizing discrete functions, there has been no integrated framework to deal with multiple constrained discrete optimization. We develop a novel algorithm for general discrete optimization problems, which can be considered as a generalization of the parametric mincut algorithm for multiple constraints.

2 Setup

Consider a problem to minimize a pseudo-Boolean function $f : \{0,1\}^n \to \mathbb{R}$ under multiple constraints.

$$\min_{x\in\{0,1\}^n} \left\{f(x) : h_i(x) = b_i,\ 1 \leq i \leq m\right\}, \tag{1}$$

where $x \in \{0,1\}^n$, for $1 \leq i \leq m$, $h_i : \{0,1\}^n \to \mathbb{R}$ and $b_i \in \mathbb{R}$, and m is a constant. We denote $(h_1(x), \dots, h_m(x))$ by $H(x)$. We first consider equality constraints, and we show that our algorithm can be generalized to inequality constraints in Section 3.3.

To solve (1), we consider the following Lagrangian dual g of f.

$$g(\lambda) = \min_{x\in\{0,1\}^n} L(x,\lambda), \tag{2}$$

where

$$L(x,\lambda) = f(x) + \lambda^T(H(x) - b). \tag{3}$$

As in the continuous minimization, maximizing g over λ provides a lower bound for (1). First we define the following.

Definition 1 (Characteristic Set)**.** *The Characteristic Set of g is define by*

$$\chi_g = \bigcup_{\lambda\in\mathbb{R}^m} \operatorname*{argmin}_x L(x,\lambda). \tag{4}$$

We abuse the notation $\operatorname{argmin}_x L(x,\lambda)$ to refer to a set of x's including all ties. Note that g is defined over a continuous space while f is defined over a discrete space. Then, the following Lemma holds.

Lemma 1. *Let $x^* \in \chi_g$ and $b^* = H(x^*)$. Then $f(x^*) = \min_{x\in\{0,1\}^n} \{f(x) : H(x) = b^*\}$ [4].*

Our goal is to compute the characteristic set χ_g. Note that χ_g does not depend on the constraint instance b. Therefore, computing the characteristic set χ_g for any fixed b is indeed equivalent to solving (1) for many b values. Since $L(x,\lambda)$ for a fixed x is linear in λ, g can be considered as a polytope which is the intersection of finite number of hyperplanes in $\mathbb{R}^{m+1}$. In Section 3.1, we regard $b = \mathbf{0}$ unless there is explicit specification. An important implication of χ_g is:

$$g(\lambda) = \min_{x \in \{0,1\}^n} L(x, \lambda) = \min_{x \in \chi_g} L(x, \lambda), \tag{5}$$

which means that $\min_{x \in \{0,1\}^n} L(x, \lambda)$ indeed depends on a much smaller set χ_g. In Section 4, we show that $|\chi_g|$ is polynomially bounded in n for image segmentation in computer vision with many interesting constraints.

3 Our Algorithms

3.1 Algorithm for Computing χ_g

In this section, we describe our algorithm which computes the characteristic set χ_g. we assume that for a large enough set $S \subset \mathbb{R}^m$, there is an oracle to compute the Lagrangian dual g of f efficiently for any fixed $\lambda \in S$. For simplicity, we assume $S = [-M, M]^m$ for large M. We denote the oracle call by a function defined by

$$\mathcal{O}(\lambda) = \operatorname*{argmin}_{x \in \{0,1\}^n} L(x, \lambda). \tag{6}$$

For example, such an oracle exists when $L(\lambda, x)$ is submodular over x for any fixed $\lambda \in \mathbb{R}^m$. Basically, our algorithm decides λ's in S for which the oracle is called. Later we prove that the number of oracle calls in our algorithm to compute χ_g is polynomial in $|\chi_g|$. Depending on whether an oracle outputs all ties or not, our algorithm may not compute $x \in \chi_g$ such that $L(x, \lambda) \geq L(x', \lambda)$ for all $x' \in \chi_g$ and $\lambda \in S$. Note that even if such x might not be computed, our algorithm still computes minimizers for all $\lambda \in S$. Before describing the algorithm in detail, first we define an induced dual g_X of a given dual $g : \mathbb{R}^m \to \mathbb{R}$ on $X \subseteq \{0,1\}^n$.

Definition 2 (Induced dual of g on X)**.** *Let $g : \mathbb{R}^m \to \mathbb{R}$ be the Lagrangian dual of f. The induced dual g_X of g is defined by*

$$g_X(\lambda) = \min_{x \in X} L(x, \lambda). \tag{7}$$

From the definition of χ_g, note that $g = g_{\{0,1\}^n} = g_{\chi_g}$. For each $x \in \{0,1\}^n$, the corresponding hyperplane P_x is defined by:

$$P_x = \{(\lambda, z) \in \mathbb{R}^{m+1} : \lambda \in \mathbb{R}^m, z = L(x, \lambda)\}. \tag{8}$$

For convenience, we will denote any $p \in \mathbb{R}^{m+1}$ by (λ_p, z_p) where $\lambda_p \in \mathbb{R}^m$ is the first m coordinates of p and $z_p \in \mathbb{R}$ is the $(m+1)$-th coordinate of p. Since each $x \in \{0,1\}^n$ corresponds to a hyperplane in $(m+1)$-dimension and $\{0,1\}^n$ is finite, g consists of the boundary of the polytope from (2). Then χ_g corresponds to the collection of m-dimensional *facets* of that polytope. Note that when $b \neq \mathbf{0}$ in (3), only a linear term $-\lambda^T b$ is added to $g(\lambda)$. Thus, χ_g corresponding to the boundary m-dimensional facets of g is invariant over b.

To compute χ_g, we use a structure called *skeleton* defined below. Intuitively, the skeleton is a collection of *vertices* and *edges* of the polytope by g.

Definition 3 (Proper convex combination)**.** *Given $x, x_1, \ldots, x_k \in \mathbb{R}^\ell$, x is a proper convex combination of $\{x_i : 1 \leq i \leq k\}$ if $x = \sum_{i=1}^k \alpha_i x_i$ for some $\alpha \in (0,1)^k$ with $\sum_{i=1}^k \alpha_i = 1$.*

Definition 4 (Skeleton of an induced dual g_X over S)**.** *For a given induced dual $g_X : \mathbb{R}^m \to \mathbb{R}$, let $\Gamma_X(S) = \{q \in \mathbb{R}^{m+1} : \lambda_q \in S,\ z_q \leq g_X(\lambda_q)\}$, and for $u, v \in \Gamma_X(S)$, $e(u,v) \in \Gamma_X(S)$ is the line segment connecting u and v. The skeleton of g_X is $\mathcal{G}_{g_X} = (\mathcal{V}_{g_X}, \mathcal{E}_{g_X})$ satisfying the followings.*

- *$\mathcal{V}_{g_X} \subset \Gamma_X(S)$, and $v \in \mathcal{V}_{g_X}$ if and only if v is a proper convex combination of $U \subseteq \Gamma_X(S)$ implies that $U = \{v\}$,*
- *$\mathcal{E}_{g_X} = \{e(u,v) : u, v \in \mathcal{V}_{g_X},\ y \in e(u,v)$ is a proper convex combination of $W(y) \subseteq \Gamma_X(S)$ implies that $W(y) \subseteq e(u,v)\} \cup \{e(u,v) : u \in \mathcal{V}_{g_X},\ \lambda_u \in \{-M, M\}^m,\ v = (\lambda_u, -M)\}$.*

Initially, the algorithm begins with $X = \{x_0\}$ and $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where x_0 is the output of the oracle call for a $\lambda_0 \in \{-M, M\}^m$. Let $\mathcal{V} = \{v_1, \ldots, v_{2^m}\} \subset \mathbb{R}^{m+1}$ where $\{\lambda_{v_i} : 1 \leq i \leq 2^m\} =$

```
$ComputeCharSet$
Input: Oracle $\mathcal{O}$
Output: $X, \mathcal{G} = (\mathcal{V}, \mathcal{E})$
1  $(X, \mathcal{G}) \leftarrow InitSkeleton()$
2  Give $\mathcal{V}$ an arbitrary order
3  foreach $v \in \mathcal{V}$ in the order do
4      $x_v = \mathcal{O}(\lambda_v)$
5      if $P_{x_v}(\lambda_v) < z_v$ then
6          $X = X \cup \{x_v\}$
7          $\mathcal{V}^+ = \mathcal{E} \cap P_{x_v}$ is appended to $\mathcal{V}$ in arbitrary order
8          $\mathcal{V}^- = \{u \in \mathcal{V} : z_u > P_{x_v}(\lambda_u)\}$ is removed from $\mathcal{V}$
9          $\mathcal{E}^- = \{e(u_1, u_2) \in \mathcal{E} : u_1 \in \mathcal{V}^-$ or $u_2 \in \mathcal{V}^-\}$
10         $\mathcal{E}^+ = \{e(u_1, u_3) : \exists\, e(u_1, u_2) \in \mathcal{E}^-,\ u_3 = e(u_1, u_2) \cap P_{x_v}\}$
11         $\mathcal{E} = \mathcal{E} \cup ConvEdge(\mathcal{V}^+) \cup \mathcal{E}^+ - \mathcal{E}^-$
12     end
13 end
```

Figure 1: Pseudocode of the algorithm computing χ_g

$\{-M, M\}^m$, and $z_{v_i} = P_{x_0}(\lambda_{v_i})$ for $1 \leq i \leq 2^m$. Let $\mathcal{E} = \mathcal{E}_{g_X}$. Note that $\mathcal{G} = \mathcal{G}_{g_X}$, the skeleton of g_X. This initialization is denoted by $InitSkeleton()$ and it returns X and $\mathcal{G}$. Figure 1 describes the algorithm. Here, $ConvEdge(\mathcal{V}^+)$ is the set of edges the convex hull of $\mathcal{V}^+$. Then, we obtain the following Theorems whose proofs are provided in [6].

Theorem 1. *When $ComputeCharSet$ terminates with X and $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $X = \chi_g$.*

Theorem 2. *The number of oracle calls in $ComputeCharSet$ is $|\mathcal{V}_g| + |\chi_g|$.*

As we have mentioned above, each $x \in \chi_g$ corresponds to a facet of an $(m+1)$-dimensional convex polytope. Since a vertex is determined by the intersection of $m+1$ facets, at the end of our algorithm, $|\mathcal{V}_g|$ is bounded by $O(|\chi_g|^{m+1})$. Thus, the query complexity becomes $O\left(poly(|\chi_g|)\right)$.

3.2 Algorithm for a Specific Constraint Instance

In this section, we explain how our algorithm is modified to obtain a lower bound of (1) for any specific constraint instance $b \in \mathbb{R}^m$. Note that we have assumed $b = \mathbf{0}$ because we focused on computing χ_g which is invariant over b. For a given b, if a corresponding optimal solution is in χ_g, the computed lower bound is the same as the minimum value of the primal problem. Our modification exploits concavity of the dual and *the weak duality* stating that the dual maximum is a lower bound of the primal minimum. The modification is as follows.

- The initial vertex set is changed to $\mathcal{V}' = \{v\}$ where $z_v \geq z_u$ for all $u \in \mathcal{V}$, and $\mathcal{V}$ is the ordinary initial skeleton vertex.
- Line 7 of Figure 1 is changed to "one of $v \in \mathcal{V}^+$ such that $z_v \geq z_u$ for all $u \in \mathcal{V}^+$ is added to $\mathcal{V}$".

With this modified algorithm, the following Lemma holds, and the proof is provided in [6].

Lemma 2. *When the algorithm terminates, for the last v^*, $z_{v^*} = g(\lambda_{v^*}) = \max_\lambda g(\lambda)$.*

Note that this modified algorithm uses much less number of oracle calls than $|\chi_g|$, which leads to fast computation of the maximum value of g and a corresponding primal solution. The modified algorithm can be considered as an efficient implementation of the cutting plane method for constrained discrete optimization [4]. While the cutting plane method computes the maximum of the dual by linear programming with computed hyperplanes at each time, our algorithm computes it efficiently by keeping and updating the skeleton of the dual.

3.3 Algorithm for Inequality Constraints

Our algorithm can also deal with problems having inequality constraints by inserting a slack variable. Let us consider the following problem.

$$\min_x \{f(x) : b - k \leq H(x) \leq b\}, \tag{9}$$

where $k \in \mathbb{R}^m$. First we transform the problem to a problem with equality constraints using a slack variable $y \in \mathbb{R}^m$ as follows.

$$\min_{x,y} \left\{\hat{f}(x,y) : H(x) + y = b\right\}, \tag{10}$$

where $y \in \prod_{i=1}^m [0, k_i]$, and $\hat{f}(x,y) = f(x)$. Let us consider the following Lagrangian: $\hat{L}(x,y,\lambda) = \hat{f}(x,y) + \lambda^T(H(x)+y-b)$. For a minimizer (x^*, y^*) of $\hat{L}$ for a fixed λ, it always holds that $y_i^* = 0$ for $\lambda_i > 0$, $y_i^* = k_i$ for $\lambda_i < 0$, and y^* can be any number in $[0, k_i]$ for $\lambda_i = 0$. Hence, y^* only depends on λ. Then, we obtain the following.

$$\hat{g}(\lambda) = \min_{x,y^*} \left\{f(x) + \lambda^T(H(x) + y^* - b)\right\}. \tag{11}$$

Then, $\max_\lambda \hat{g}(\lambda)$ is a lower bound of (9). Since y^* is determined only by λ, $\hat{g}(\lambda)$ can be computed by the same oracle for $g(\lambda)$. Note that $\chi_{\hat{g}} = \bigcup_{\lambda \in S} \mathrm{argmin}_{x,y^*} \hat{L}(x, y^*, \lambda)$, which can be computed by our algorithm.

4 Application to Image Processing

In computer vision, discrete minimization has become a key tool for many fundamental problems such as image segmentation, 3D-reconstruction and stereo. In this section, we explain how our algorithm can be applied to energy minimization problems in computer vision by an example of the image segmentation problem.

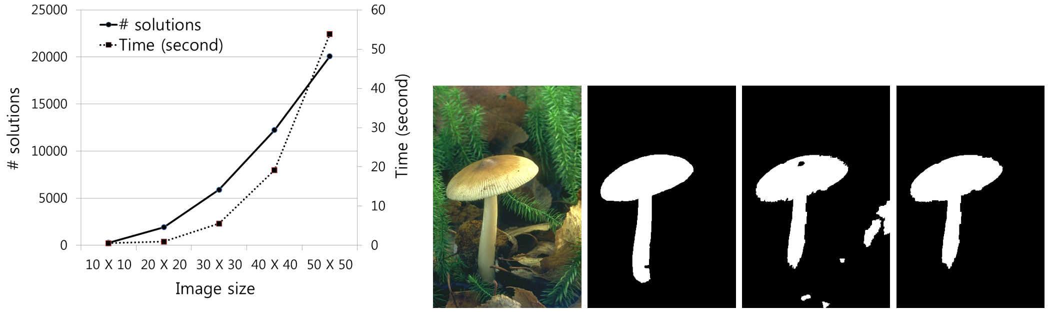
The foreground-background(fg-bg) image segmentation problem is to divide a given image to a foreground(object) and a background. This can be done by labelling all pixels such that 1 is assigned to foreground pixels and 0 is assigned to background pixels. For this problem, one popular approach is to consider an image as a grid graph in which each node has four neighbours, and minimize an *energy function* of the following form.

$$f(x) = \sum_{i \in V} \phi_i(x_i) + \sum_{(i,j) \in E} \phi_{ij}(x_i, x_j), \tag{12}$$

where all ϕ_{ij}'s are submodular. The unary terms of the functions encode how likely each pixel belongs to the foreground or background objects, while the pairwise terms encode the smoothness of the boundary of the objects. It is well known that (12) can be minimized efficiently by reducing it to a st-mincut problem [11].

A natural constraint to enforce in the fg-bg segmentation problem is a particular size for the foreground object in the image, represented by $\sum_{i \in V} x_i = b_1$. This approach has been extensively studied in computer vision when there is prior knowledge about the object size [13, 12]. The segmentation can also be constrained to be consistent with other statistics related to the shape of the object such as the mass center and covariance [8]. The mass center is the means of the vertical and horizontal coordinates of the object. This is one of easiest constraints to obtain from the user, for example, by drawing a circle roughly containing the object. Let v_i and h_i denote the vertical and horizontal coordinates of a pixel i, respectively. Then the mass center can be represented by $\sum_{i \in V} \frac{v_i x_i}{\sum_{i \in V} x_i} = b_2$ and $\sum_{i \in V} \frac{h_i x_i}{\sum_{i \in V} x_i} = b_3$. The covariance constraint represents the "covariance" of the object coordinates, which is represented by $\sum_{i \in V} \frac{x_i(v_i - \mu_v)(h_i - \mu_h)}{\sum_{i \in V} x_i} = b_4$ for some μ_v and μ_h. We can also define the variance constraint for the vertical and horizontal coordinates in the similar way. Note that all of these constraints are linear on variables x_i's, and hence adding them to (12) with any constant factor λ_i does not affect submodularity. Thus, if we consider any combination of these constraints, the search range S for our algorithm can be an arbitrary region of the dual space.

In many scenarios, researchers are also interested in ensuring that the boundary of the object in the segmentation is of a particular specified length. As the object boundary can be measured by counting



(a) (b) (c) (d) (e)

Figure 2: (a) We used one grey scale image while varying the resolution. The energy function in (12) is used with $\phi_i(x_i) = 255 - 2I_i$, and $\phi_{ij}(x_i, x_j) = 65|x_i - x_j|$ where I_i is a color in $[0, 255]$ of a pixel i. We consider the constraints in (13). The graph shows the size of computed characteristic set χ_g and the running time over varying image size. (b) original image (size: 321×481). (c) ground truth. (d) segmentation without constraint. (e) segmentation with specific size and boundary inequality constraints. The running time was 2.60 seconds.

the number of pairs of adjacent variables having different labels, we can encode this constraint as: $\sum_{(i,j)\in E} |x_i - x_j| = b_5$. While using the boundary constraint, the search range S may be restricted to ensure that $L(x, \lambda)$ is submodular over x. Note that convex relaxation based approaches [2, 8] cannot deal with this type of constraint. The following is an example of a Lagrangian containing the size and boundary constraints.

$$L(x, \lambda) = f(x) + \lambda_1 \sum_{i\in V} x_i + \lambda_2 \sum_{(i,j)\in E} |x_i - x_j|. \tag{13}$$

To apply our algorithm to (13), S should be a subset of $\mathbb{R} \times [K, \infty]$, not $\mathbb{R}^2$, where $K < 0$ is the smallest real number making (13) submodular for all $\lambda \in S$. Note that our algorithm can deal with inequality constraints such that $C_1 \le \sum_{i\in V} x_i \le C_2$ and $C_3 \le \sum_{(i,j)\in E} |x_i - x_j| \le C_4$. We performed simulations on image segmentation with (13). Figure 2 shows the simulation results.

References

[1] R. G. Bland, D. Goldfarb, and M. J. Todd. The ellipsoid method: A survey. *Operation Research*, 29, 1981.

[2] T. Chan, S. Esedoḡlu, and M. Nikolova. Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal on Applied Mathematics*, 66, 2006.

[3] U. Feige, V. S. Mirrokni, and J. Vondrák. Maximizing non-monotone submodular functions. In *FOCS*, 2007.

[4] M. Guignard. Lagrangean relaxation. *TOP*, 11, 2003.

[5] S. Iwata and K. Nagano. Submodular function minimization under covering constraints. In *FOCS*, 2009.

[6] K. Jung, P. Kohli, and Y. Lim. Efficient discrete optimization with multiple constraints via dual space search. In *ArXiv*.

[7] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4, 1984.

[8] A. Klodt and D. Cremers. A convex framework for image segmentation with moment constraints. In *ICCV*, 2011.

[9] K. Kolev and D. Cremers. Integration of multiview stereo and silhouettes via convex functionals on convex domains. In *ECCV*, 2008.

[10] V. Kolmogorov, Y. Boykov, and C. Rother. Application of parametric maxflow in computer vision. In *ICCV*, 2007.

[11] V. Kolmogorov and R. Zabih. What energy functions can be minimized using graph cuts? In *ECCV*, 2002.

[12] Y. Lim, K. Jung, and P. Kohli. Energy minimization under constraints on label counts. In *ECCV*, 2010.

[13] O. Woodford, C. Rother, and V. Kolmogorov. A global perspective on MAP inference for low-level vision. In *ICCV*, 2009.