Multi Label Generic Cuts: Optimal Inference in Multi Label Multi Clique MRF-MAP Problems

Chetan Arora*
The Hebrew University of Jerusalem
Jerusalem, Israel

S.N. Maheshwari
Indian Institute of Technology Delhi
New Delhi, India

Abstract

We propose an algorithm called Multi Label Generic Cuts (MLGC) for computing optimal solutions to MRF-MAP problems with submodular multi label multi-clique potentials. A transformation is introduced to convert a m-label k-clique problem to an equivalent 2-label (mk)-clique problem. We show that if the original multi-label problem is submodular then the transformed 2-label multi-clique problem is also submodular. We exploit sparseness in the feasible configurations of the transformed 2-label problem to suggest an improvement to Generic Cuts [3] to solve the 2-label problems efficiently. The algorithm runs in time $O(m^k n^3)$ in the worst case (n is the number of pixels) generalizing $O(2^k n^3)$ running time of Generic Cuts. We show experimentally that MLGC is an order of magnitude faster than the current state of the art [17, 20]. While the result of MLGC is optimal for submodular clique potential it is significantly better than the compared methods even for problems with non-submodular clique potential.

1. Introduction

Many problems in computer vision can be naturally modeled as pixel labeling problems. Formulating such a problem as finding the labeling with Maximum a-posteriori Probability (MAP) in a Markov Random Field (MRF) setting converts the problem to a discrete optimization problem. The MRF-MAP inference problem in general is NP-hard. However, because of its immense applicability, various algorithms have been developed which can perform an optimal or efficient approximate inference on a subset of problems.

It has been known for a long time [10] that the 2-label 2-clique MRF-MAP problems, with submodular clique potential, can be solved optimally by finding a minimum cut in a appropriately constructed flow graph. Boykov and Kolmogorov's Graph Cut algorithm [7] and it's variations [15] are the most widely used algorithms for solving problems which fit into this limited framework. Ishikawa [11] extended the class for which optimal inference can be found in polynomial time to multi-label 2-clique problems with submodular clique potentials. His construction uses notions of convexity and ordering among labels to formulate a flow graph on which minimum cut corresponds to the minimum energy configuration. Both of these algorithms have resisted generalization and techniques for solving multi-label and/or higher order potential energy minimization have been either indirect, involving reduction to pairwise problems, or non combinatorial, involving some kind of iterative or message passing framework.

Indirect schemes transform the 2-label multi-clique problems to equivalent 2-clique problems using reduction techniques [12, 9, 14] and then work with the reduced/transformed problem using Graph Cut [7] or QPBO [19]. In general the transformation process can result in exponential number of additional terms. Also, transformations may convert submodular potentials into non-submodular [6, 3]. For working with multi-label problems, these schemes typically use α-expansion [8] in the outer loop, where solutions to multi-label multi-clique problems are approximated by repeatedly solving a series of 2-label multi-clique problems. While the theoretical convergence is slow, working approximations are available in a few cycles of α-expansion iterations.

Ramalingam et al. [23] showed that it is possible to transform multi label higher order problems to 2-label second order problems. Their transformation is a two stage process. First stage involves, using an encoding similar to that in [11], replacing multi-label variables in the energy function by Boolean variables. Second stage reduces the higher order Boolean energy function to a second order system. This obviates the use of α-expansion iterations. However, the transformation is polynomial time and preserves submodularity only for multi label problems of order three or less.

There have also been attempts based on Generalized Roof Duality to handle higher order 2-label [14] and multi-

[*] Chetan Arora is now with IIIT Delhi.

label [26] functions by working with their submodular relaxations. The objective for such techniques is to output a good approximate solutions with weak persistence as measure of goodness. In general, these techniques cannot handle cliques of size larger than four for 2-label problems [14] and work with only pairwise potentials for multi-label problems [26].

Non combinatorial techniques involve use of gradient descent at some stage in the algorithm. Dual decomposition [20] based methods break a problem into smaller subproblems and work best when techniques exist to solve subproblems optimally. Combining sub solutions involves some form of gradient descent. This needs to be done repeatedly until convergence. Using this framework to multi-label as well as multi-clique problems has been attempted [21, 20, 25]. In general, the decomposition step result in sub problems that can not be solved optimally leading to lack of convergence guarantees. Also, there are no bounds on the number of gradient descent iterations in general.

When clique potentials are submodular (multi-label and/or multi-clique), MRF MAP energy function minimization is essentially minimizing a sum of submodular functions [18]. Submodular function minimization has been studied extensively. It has been known for a decade that submodular set functions can be minimized in strongly polynomial time [13]. However, since the most efficient algorithm for minimizing a submodular function has $O(n^6)$ time complexity [13] direct use of these algorithms to solve computer vision problems where the number of pixels/nodes can reach millions is impractical. Generic Cuts [3] algorithm proposed recently, is a gadget based flow approach for optimal inference on all 2-label submodular functions. The importance of Generic Cuts also lies in showing the effectiveness and efficiency of direct combinatorial algorithm for 2-label multi-clique problems over reduction based or gradient descent based approaches. We show in this paper that the gadget for Generic Cuts can be the basis for a polynomial time transformation that maps multi-label problems to 2-label problems preserving submodularity. Generic Cuts can then be used to perform optimal inference in multi-label multi-clique problems.

The basic idea is to transform the original multi-label multi-clique problem to a 2-label multi-clique problem by encoding the original multi label variables using multiple binary variables. We show that using a simple encoding, any higher order multi-label problem can be transformed into 2 label problem preserving submodular properties of the original problem. The encoding has been used earlier in [11] and [23]. The focus in the earlier work has been to come up with a representation which can be used to transform the original problem to a 2-label 2-clique submodular problem. Our focus, on the other hand, is to transform any higher order multi-label function to a 2 label function which preserves submodularity. In contrast to the earlier approaches the clique size of the transformed problem actually increases in our case. However, we show that though the clique size increases there is an interesting generalization of Generic Cuts which can still solve such problems optimally in polynomial time when label and clique sizes are fixed. This shift in focus from transformations leading to Boolean second order functions to transformations outputting higher order Boolean problems preserving submodularity and are applicable to multi-label problems of arbitrary order is one of the major contributions of this paper.

It is important to note that though the transformation converts submodular multi-clique multi-label problems into submodular 2-label multi-clique problem, by doing so we do not violate any hardness results of the type that there exists no polynomial transformation from submodular multi-label functions of order four or more to submodular Boolean second order functions [23]. This is true since the transformation does not convert the problem to a second order function but a 2-label problem whose order is even higher than the original problem and it is known that not all 2-label higher order submodular problems can be converted to submodular pairwise ones [6, 3].

We have compared performance of our algorithm with algorithms like [17, 20] which work directly on multi-label multi-clique problems. Our algorithm not only runs an order of magnitude faster, it outputs optimal results for submodular potentials and better approximations on nonsubmodular potentials than other methods.

The organization of the paper is as follows. Section 3 introduces the label encoding transformation. Section 4 details out the optimization algorithm. Section 5 and 6 contains details of the experimental investigations and conclusions respectively.

2. Basic Notation and Definitions

We denote the set of pixels in an image by $\mathcal{P}$, the set of higher order cliques by $\mathcal{C}$, and set of labels by $\mathcal{L}$. Number of pixels is denoted by n, size of clique by k and number of labels is denoted by m. We assume an ordering on the label set allowing us to represent $\mathcal{L}$ as $\{1, \dots, m\}$. The ordering on the label set also enables us to define $\max$ and $\min$ operations on two labels. We denote the label of a pixel p as $l_p \in \mathcal{L}$. The labeling configuration of clique $\mathbf{c}$, denoted by $\mathbf{l_c}$, is essentially the set of k labels associated with the pixels in clique $\mathbf{c}$. $\mathbf{l}_\mathbf{c}^p$ denotes the label of pixel p in $\mathbf{l_c}$. Finding a labeling with maximum *a posteriori* probability (MAP) assuming labeling to be a MRF can be shown to be equivalent to minimizing energy of following kind:

$$E(\mathbf{l}_\mathcal{P}) = \sum_{p \in \mathcal{P}} D_p(l_p) + \sum_{\mathbf{c} \in \mathcal{C}} W_\mathbf{c}(\mathbf{l_c}), \qquad (1)$$

where $D_p(l_p)$, called the *unary potential*, is the cost of assigning label l_p to p. $W_{\mathbf{c}}(\mathbf{l_c})$, called the *clique potential*, is the penalty/cost of any labeling configuration $\mathbf{l_c}$ on clique $\mathbf{c}$.

Equation (1) can be reparametrized such that effect of $D_p(l_p)$ is equivalently expressed using only clique potentials. One of the ways to achieve this is to choose a clique $\mathbf{c}$ such that $p \in \mathbf{c}$. Then increase values of $W_{\mathbf{c}}(\mathbf{l_c})$ such that $\mathbf{l}^p_{\mathbf{c}} = l_p$ by amount $D_p(l_p)$ and set $D_p(l_p) = 0$. It can be shown that the original and reparametrized function have identical values for all labeling configurations. In the discussion that follows we will assume that the energy function as given in (1) has been reparametrized if needed to the form given below:

$$E(\mathbf{l}_{\mathcal{P}}) = \sum_{\mathbf{c} \in \mathcal{C}} W_{\mathbf{c}}(\mathbf{l_c})). \qquad (2)$$

2.1. Submodularity

$W_{\mathbf{c}}(\cdot)$ can be looked upon as a k-ary function with each parameter (representing a pixel of the k-clique) taking a value from the label set $\mathcal{L}$. Let X and Y stand for the k-tuples of parameters. Let $\vee$ and $\wedge$ be $\max$ and $\min$ operators and let $(X \vee Y)$, $(X \wedge Y)$ denote k-tuples resulting from element wise application of the $\max$ and $\min$ operators over k-tuples X and Y. $W_{\mathbf{c}}(\cdot)$ is called submodular if:

$$W_{\mathbf{c}}(X) + W_{\mathbf{c}}(Y) \geq W_{\mathbf{c}}(X \vee Y) + W_{\mathbf{c}}(X \wedge Y). \qquad (3)$$

It is easy to see that when X and Y takes value from label set $\mathcal{L} = \{0, 1\}$ the above condition reduces to following:

$$W_{\mathbf{c}}(X) + W_{\mathbf{c}}(Y) \geq W_{\mathbf{c}}(X \cup Y) + W_{\mathbf{c}}(X \cap Y). \qquad (4)$$

Note that a multi-label function which is submodular under one label ordering may not be submodular under some other ordering. It is known that reparametrization preserves submodularity when $|\mathcal{L}| = 2$ [3]. In the supplementary we show it for the case when $|\mathcal{L}| > 2$.

3. Conversion to 2-Label Problem

For the purpose of illustrating transformation introduced in this section we assume that the potential function correspond to m-label 2-clique problem. However, the transformation itself is generalizable to clique potentials of arbitrary order.

3.1. Encoding

Let the label set $\mathcal{L}$ be represented by the sequence $\{l_1, l_2, \ldots, l_m\}$. For each pixel p we introduce an m-tuple $\mathbf{b_p} = (b^p_1, b^p_2, \ldots, b^p_m) \in \mathbb{B}^m$. In any labeling configuration, a pixel p can take any of the m labels. The m possible states of p is equivalently represented by states $\{(0,1,1,\ldots,1), (0,0,1,1,\ldots,1), \ldots, (0,0,0,\ldots,1)\}$ of $\mathbf{b_p}$. Observe that the label at position i in $\mathcal{L}$ is represented by the state of $\mathbf{b_p}$ in which i Boolean variables from left have value 0 and remaining $(m-i)$ variables have value 1. We use $B(l_p)$ to denote the state of $\mathbf{b_p}$ corresponding to label l_p of p.

3.2. Transformation

For the 2-ary function $W_{\{p,q\}}(l_p, l_q)$ defined on labels of pixels in clique $\{p, q\}$, we introduce a $2m$-ary function $W^b_{\{p,q\}}(B(l_p), B(l_q)) : \mathbb{B}^{2m} \to \mathbb{R}$. Note that the size of the domain of function $W^b_{\{p,q\}}$ is 2^{2m} states of the $2m$ variables involved. Of these 2^{2m} states only m^2 corresponds to labels of pixels p and q. These m^2 states which correspond to the labels associated with the two pixels in question are termed *feasible*. We call the remaining $2^{2m} - m^2$ states which do not correspond to labels of pixels, *infeasible*. The value of $W^b_{\{p,q\}}(\cdot)$ for feasible states is set equal to the value of $W_{\{p,q\}}(\cdot)$ for the corresponding labels. The value of the function $W^b_{\{p,q\}}(\cdot)$ for infeasible states is set to a very large number.[1] This is to ensure that $W^b_{\{p,q\}}$ attains its minimum value only on feasible states, i.e., those which correspond to some valid labeling of pixels. Therefore:

$$\begin{aligned} &\min_{\mathbf{l}_{\mathcal{P}}} \sum_{\{p,q\} \in \mathcal{C}} W_{\{p,q\}}(l_p, l_q) \\ = \; &\min_{\mathbf{b}} \sum_{\{p,q\} \in \mathcal{C}} W^b_{\{p,q\}}(B(l_p), B(l_q)), \qquad (5) \end{aligned}$$

where $\mathbf{b}$ is a collection of all binary variables associated with all pixels in $\mathcal{P}$. Note that $W_{\{p,q\}}$ was defined on m-state variables of a 2-clique problem. $W^b_{\{p,q\}}$ can be looked upon as being defined on 2-state variables of a $2m$-clique problem. The transformation essentially embeds a m-label 2-clique energy function onto a 2-label $2m$-clique energy function.

3.3. Submodularity Preservation

Let $\cup$ denote the *'OR'* operator applied element wise over tuples of Boolean variables such that:

$$B(l_p) \cup B(l_q) = \left((b^1_p \cup b^1_q), \ldots, (b^m_p \cup b^m_q)\right). \qquad (6)$$

The proposed transformation has the following crucial property:

$$B(\max(l_p, l_q)) = B(l_p) \cup B(l_q), \qquad (7)$$

where l_p and l_q are two labels from set $\mathcal{L}$ at pixel p and q respectively and $B(\cdot)$ is the Boolean variable representation of a multi label variable. Similarly if $\cap$ refers to the element wise *'AND'* on tuples of Boolean variables then:

$$B(\min(l_p, l_q)) = B(l_p) \cap B(l_q). \qquad (8)$$

[1] we shall come back to the issue of setting values for infeasible states later again

$$\begin{aligned} W_{\{p,q\}}(l_a, l_b) + W_{\{p,q\}}(l_c, l_d) &\geq W_{\{p,q\}}(l_a \vee l_c, l_b \vee l_d)) + W_{\{p,q\}}(l_a \wedge l_c, l_b \wedge l_d), \\ \Rightarrow W^b(B(l_a), B(l_b)) + W^b(B(l_c), B(l_d)) &\geq W^b(B(l_a \vee l_c), B(l_b \vee l_d)) + W^b(B(l_a \wedge l_c), B(l_b \wedge l_d)), \\ \Rightarrow W^b(B(l_a), B(l_b)) + W^b(B(l_c), B(l_d)) &\geq W^b(B(l_a) \cup B(l_c), B(l_b) \cup B(l_d)) + W^b(B(l_a) \cap B(l_c), B(l_b) \cap B(l_d)). \end{aligned}$$

Figure 1: Proposed encoding ensures that if multi label clique potential is submodular than encoded 2-label clique potential is submodular as well

Let $X = \{l_a, l_b\}$ and $Y = \{l_c, l_d\}$ be two labeling configuration on clique $\{p, q\}$. If clique potential $W_{\{p,q\}}$ is submodular and then it can be shown that implications in Figure 1 hold. The first implication in Figure 1 holds because values of $W_{\{p,q\}}$ and $W^b_{\{p,q\}}$ correspond one to one for all feasible states. Second implication follows from equations (6) and (7). Setting $X_{ab} = \{B(l_a), B(l_b)\}$ and $X_{cd} = \{B(l_c), B(l_d)\}$ gives us:

$$W^b(X_{ab}) + W^b(X_{cd}) \geq W^b(X_{ab} \cup X_{cd}) + W^b(X_{ab} \cap X_{cd}),$$

which is the submodularity condition for 2-label clique potentials. This implies that if the given $W_{\{p,q\}}$ is submodular, $W^b_{\{p,q\}}$ satisfies all conditions for submodularity for the feasible Boolean variable states. We would like to point out that the above also follows from Birkhoff's Representation Theorem [5]. Readers are referred to [22] for its varied applications in optimization literature.

Since W^b attains its minimum value only on feasible states, in effect we have an encoding to transform a m-label 2-clique inference problem to 2-label $2m$-clique problem having the same minimum energy configuration. If the original multi-label clique potential is submodular then the derived 2-label potential function is guaranteed to be submodular over the domain of interest.

We do not elaborate, but give a brief glimpse here on how $W^b_{\{p,q\}}$ can be made submodular even for infeasible states. Note that in any feasible binary variable state, binary variable values are 0s followed by some 1s depending on which label they are encoding. All feasible states therefore have exactly one $0 \rightarrow 1$ transition and no $1 \rightarrow 0$ transition whereas all infeasible states have at least one $1 \rightarrow 0$ transition.[2] If we set the value of $W^b_{\{p,q\}}(B(l_p), B(l_q))$ equal to the x times a very large number where x is sum of $1 \rightarrow 0$ transitions in $B(l_p)$ and $B(l_q)$ then it can be shown that $W^b_{\{p,q\}}$ will satisfy submodularity conditions even with terms corresponding to infeasible states.

The proposed transformation can be generalized to embed a k-ary m-label energy function into km-ary 2-label energy function for any value of k and m. As done earlier, for each pixel introduce m binary variables to encode the m possible states of l_p. For each k-ary function $W_{\mathbf{c}}(\cdot)$ introduce a km-ary function $W^b_{\mathbf{c}}(\cdot)$. If $\mathbf{c}$ is $p, q, \ldots, r$ then the form of the new function is $W^b_{\{p,q,\ldots,r\}}(b^p_1, \ldots, b^p_m, b^q_1, \ldots, b^q_m, \ldots, b^r_1, \ldots, b^r_m,)$. As before the $2^{km} - m^k$ states of the km binary variables, which do not correspond to the labels of the pixels, are termed *infeasible* and the value of the function $W^b_{\{p,q,\ldots,r\}}(\cdot)$ for infeasible states is a very large number. The rest of the m^k states which correspond to the labels associated with the pixels in question are termed *feasible* and the value of $W^b_{\{p,q,\ldots,r\}}(\cdot)$ for feasible states is set equal to the value of $W_{\{p,q,\ldots,r\}}(\cdot)$ for those labeling configurations. As shown for the 2-clique case, if $W_{\{p,q,\ldots,r\}}(\cdot)$ is submodular then $W^b_{\{p,q,\ldots,r\}}(\cdot)$ is guaranteed to be submodular .

At this stage it is instructive to compare our transformation with the one introduced by Ishikawa in [11] to develop a polynomial time algorithm for optimal inference of convex/submodular m-label 2-clique problems for pixels arranged in a grid. The approach there involves construction of a flowgraph in which corresponding to each pixel is a set of $m - 1$ edges and m vertices arranged in a path of length m. One end of the path is connected to the source s and the other to the sink t. The interpretation is that a node and the edge emanating from it on the path corresponds to a label associated with a pixel. Weights of these and the other edges introduced by the transformation are such that an (s, t) cut in the flowgraph includes only one edge from such paths, and the label of the pixel corresponds to that edge in the cut. If all the vertices on s side of the cut are labeled 0 and the rest are labeled 1 then the sequence of labels associated with the vertices of a path are the same as the encoding defined here. However, in [11] the encoding and transformation steps are interlinked tightly whereas in our approach encoding of labels by Boolean m-tuples is delinked from the transformation step used to embed a k-ary m-label energy function into km-ary 2-label energy function. The tight interlinking of [11] is perhaps one of the reasons why transformations using this path based encoding does not seem to be possible for cliques larger than 2.

For the benefit of readers we give a worked out example here for a 3-label 3-clique energy function function defined over pixels p, q and r. The 3-label 3-clique energy function with unary energy and clique potential is given in Table 1 and Table 2 respectively. After reparametrization

[2] The case of all 0s and 1s is handled separately as we show later. Clique potential is reparametrized to be zero for state all 0s. State all 1s is explicitly denied by construction

Labeling state	Pot.	Labeling state	Pot.	Labeling state	Pot.
$D_p(a)$	10	$D_q(a)$	30	$D_r(a)$	10
$D_p(b)$	20	$D_q(b)$	20	$D_r(b)$	30
$D_p(c)$	30	$D_q(c)$	10	$D_r(c)$	20

Table 1: Unary potentials for the 3-label 3-clique energy function

Labeling state	Pot.	Labeling state	Pot.	Labeling state	Pot.
$W(a,a,a)$	0	$W(b,a,a)$	40	$W(c,a,a)$	80
$W(a,a,b)$	40	$W(b,a,b)$	40	$W(c,a,b)$	80
$W(a,a,c)$	80	$W(b,a,c)$	80	$W(c,a,c)$	80
$W(a,b,a)$	40	$W(b,b,a)$	40	$W(c,b,a)$	80
$W(a,b,b)$	40	$W(b,b,b)$	0	$W(c,b,b)$	40
$W(a,b,c)$	80	$W(b,b,c)$	40	$W(c,b,c)$	40
$W(a,c,a)$	80	$W(b,c,a)$	80	$W(c,c,a)$	80
$W(a,c,b)$	80	$W(b,c,b)$	40	$W(c,c,b)$	40
$W(a,c,c)$	80	$W(b,c,c)$	40	$W(c,c,c)$	0

Table 2: Clique potentials for clique {p, q, r}

we can convert the problem to one having zero unary costs and clique potential satisfying Eq. 2. We transform this reparametrized problem to a 2-label 9-clique problem. The resultant binary clique potential function W^b is given in Table 3.

4. Multi Label Generic Cuts (MLGC) Algorithm

In Section 3 we described the transformation to convert an arbitrary multi-label multi-clique problem to a 2-label multi-clique problem while preserving submodularity of the original multi-label clique potential. With this transformation, the problem reduces to an optimal inference for the constructed 2-label multi-clique problem.

4.1. Generic Cuts

We propose to solve the transformed problem using Generic Cuts (GC), an algorithm proposed by Arora et al. [3] that guarantees an optimal solution to the energy minimization for 2-label problems of any clique size with submodular clique potentials. It may be noted that any method that guarantees optimal inference of 2-label higher order submodular problems could be chosen in principle. Discussion that follows shows that the choice of GC is appropriate as it enables, regardless of the size of the resulting multi-clique optimization problem, limiting the calculation of capacity of flow edges to a subset of the dual feasibility constraints.

In GC a flowgraph is constructed in which each higher order clique is represented by a gadget of the type given in Figure 2. We assume some familiarity on the reader's part with the algorithmic details and notation of GC. For those who would like to use GC as a black box we summarize here the basic terminology and concepts. The edges emanating out of node n and those incident at node m in a gadget are called conjugate edges (n and m are called auxiliary nodes and the other nodes of a gadget are the pixel nodes). A flow graph is created in which each pixel is represented by a pixel node and corresponding to each clique of the energy minimization problem there is a gadget over the pixel

Labeling state	Pot.	Labeling state	Pot.
$W^b(011011011)$	50	$W^b(001001000)$	10
$W^b(011011001)$	110	$W^b(001000011)$	12
$W^b(011011000)$	140	$W^b(001000001)$	10
$W^b(011001011)$	80	$W^b(001000000)$	90
$W^b(011001001)$	100	$W^b(000011011)$	150
$W^b(011001000)$	130	$W^b(000011001)$	170
$W^b(011000011)$	110	$W^b(000011000)$	160
$W^b(011000001)$	130	$W^b(000001011)$	140
$W^b(011000000)$	120	$W^b(000001001)$	120
$W^b(001011011)$	10	$W^b(000001000)$	110
$W^b(001011001)$	12	$W^b(000000011)$	130
$W^b(001011000)$	15	$W^b(000000001)$	110
$W^b(001001011)$	90	$W^b(000000000)$	60
$W^b(001001001)$	70		

Table 3: Clique potentials for the equivalent 2-label problem corresponding to Table 1 and 2. The table lists values for feasible states only. All other labelings have infinity costs. Note that all unary costs for binary variables are set to zero

Labeling state	Pot.	Labeling state	Pot.
$W^b(011011011)$	50	$W^b(001001000)$	40
$W^b(011011001)$	110	$W^b(001000011)$	120
$W^b(011011000)$	80	$W^b(001000001)$	100
$W^b(011001011)$	80	$W^b(001000000)$	30
$W^b(011001001)$	100	$W^b(000011011)$	150
$W^b(011001000)$	70	$W^b(000011001)$	170
$W^b(011000011)$	110	$W^b(000011000)$	100
$W^b(011000001)$	130	$W^b(000001011)$	140
$W^b(011000000)$	60	$W^b(000001001)$	120
$W^b(001011011)$	100	$W^b(000001000)$	50
$W^b(001011001)$	120	$W^b(000000011)$	130
$W^b(001011000)$	90	$W^b(000000001)$	110
$W^b(001001011)$	90	$W^b(000000000)$	0
$W^b(001001001)$	70		

Table 4: Clique potentials for the 2-label problem corresponding to Table 3 after reparametrizing. $D_{b^3_r}(0)$ is now 60. All other unary costs for binary variables are zero

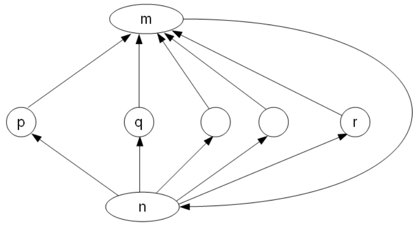

Figure 2: A GC gadget for a 2-label 5-clique problem

nodes of the clique. The dual optimization problem has 2^g dual feasibility constraints corresponding to the 2^g labeling states possible in a 2-label g clique. At any stage flow in the conjugate edges of a gadget corresponds to the value of the dual variables for the nodes of a clique in the Lagrangian dual created for the problem. Slack of the dual feasibility constraint is defined as the difference between the constraint value and the sum of flow in the conjugate edges participating in the constraint. The residual capacity of a conjugate edge is the minimum of the slacks of the dual feasibility constraints in which a conjugate edge participates. Max flow in the flow graph so created corresponds the optimal value of the dual of the energy minimization problem. The time complexity of GC has a multiplicative exponential factor 2^g where g is the size of the gadget used to model a g node 2-label clique. This multiplicative factor of 2^g in the time complexity of GC is there because the number of dual feasibility constraints whose slack needs to be evaluated to calculate the residual capacity of a conjugate edge of a gadget is 2^g.

4.2. Generic Cuts Improvement

Note that using the encoding function of Section 3 a m-label k-clique problem gets transformed to a 2-label km-clique problem. The flow graph for this km-clique problem will have gadgets of size km and the time complexity of GC will have a multiplicative factor of 2^{km}. As explained earlier there are 2^{km} dual feasibility constraints whose slack needs to be evaluated to calculate the residual capacity of a conjugate edge of a gadget. Each of these 2^{km} dual feasibility constraints corresponds to one of the 2^{km} possible states of the km binary variables introduced to model the m labels that can be associated with a pixel. We also know that only m^k of these 2^{km} states are feasible and the cost associated with an infeasible state is so large that the constraint corresponding to an infeasible state can never become tight by the process of pushing flow in the conjugate edges of a gadget. In effect the constraints corresponding to infeasible states cannot contribute to the process of computing residual capacity of conjugate edges. Only m^k constraints corresponding to feasible states have costs small enough for any one of them to potentially become tight. A brute force residual capacity calculation step should therefore need only take into consideration no more than m^k constraints. The complexity of GC modified to take into consideration only m^k constraints works out to $O(m^k.(km)^3.n^3)$, which is the complexity of MLGC. The size of the gadget changes from k to km and cost of residual capacity calculation changes from 2^k to m^k in GC's original complexity expression.

4.3. Gadget Graph Construction

The MLGC algorithm differs from traditional GC algorithm in few other ways. In GC the gadget construction first reparametrizes the 2-label clique potential (W^b in our case) such that the cost of uniform labeling 0 or 1 becomes zero. Given the multi-label to 2-label transformation of MLGC as explained in previous section, it may be noted that the labeling all binary variables of a clique to 1 corresponds to infeasible labeling and the cost of such labeling is therefore set to infinity (or very large number). Reparametrization is therefore done only to bring the value of W^b, corresponding to the labeling in which binary variables are 0, to zero. Table 4 gives the reparametrized binary clique potential corresponding to example problem discussed in Table 3. After reparametrization unary cost $D_{b_r^3}(0)$ is 60 (the earlier value value was 0).

To make sure that on a gadget all nodes can never take label simultaneously 1 (all 1s is an infeasible state), we add additional terminal edge from s to all n type auxiliary nodes in GC gadget with residual capacity of infinity. We then run the standard GC algorithm while tracking slack in constraints corresponding to feasible binary variable states only. Once the maximum flow has been found, we find the (S, T) cut by putting in S all nodes reachable from source s in the residual graph. The nodes in S set are labeled 0. Rest of the nodes are placed in T and are labeled 1. These 0/1 labels on the nodes are the values of the corresponding binary variables. The equivalent labels on the pixels of the original multi label multi clique problem can be recovered using the encoding function described in Section 3. The pseudo-code of MLGC algorithm is as given in Algorithm 1.

Algorithm 1 MLGC Algorithm

```
Reparametrize original problem such that Unary costs are all zero.
Create $m$ binary variables corresponding to each pixel.
Create a 2-label $km$-clique problem with the values of binary clique potential equal to multi-label clique potential at all feasible states and infinity otherwise
Set Unary cost for all binary variables as zero
Reparametrize 2-label problem to make cost of uniformly labeling 1 to zero.
Create a GC gadget graph with additional terminal edge at each $n$ type node from $s$ with infinite capacity.
Run standard GC algorithm tracking slack in constraints corresponding to feasible states only.
Find the min $ST$ cut, once the flow is maximized.
Label all binary variables in $S$ as 0 and in $T$ as 1.
Do a reverse lookup in encoding and find the labels at each pixel given state of the binary variables.
```

5. Experiments and Discussion

The purpose of experiments is threefold.

- To show that the primal and dual values computed by MLGC on real life problems are equal, thus confirming the optimal inference experimentally.

- To show that MLGC can do the inference much faster than the current state of the art.

- Even when the potential is not submodular, MLGC can still be used, though optimality is not guaranteed. We show that in these cases also MLGC gives better output both in terms of energy of the primal as well as visually and in a time which is an order of magnitude faster than the competitive methods.

All experiments were conducted on a computer with 2.5GHz dual core processor, 2GB of RAM running Windows 7 operating system with 64 bit addressability. Our algorithm has been implemented in C++. The comparison has been done with only those methods whose implementations are publicly available. We have used the implementations available in Darwin framework [1] of Iterated Conditional Modes (ICM) [4], Max Product Inference (MPI) [16], TRWS [17] and Dual Decomposition MAP inference (DD) [20]. We consider these as direct methods and similar in spirit with MLGC. Comparison with [23]and [26]could not be performed because of unavailability of their implementations. However, it may be noted that our method will ouput optimal results for all submodular multi-label multi-clique potentials which no other method will do.

We consider the problem of finding disparity from a pair of stereo images. Data energy term along with Sawtooth image are as used by MRF code at the Middlebury test bench [2]. We have run 8-label 3-clique problems with 4 different clique potential functions. If labels on the 3 pixels of a clique are a, b and c, then the cost of the labeling under the 4 clique potential functions is as follows:

1. $ABS\_SUM(a, b, c) = |a - b| + |b - c| + |c - a|$

2. $SQR\_ABS\_SUM(a, b, c) = (|a-b|+|b-c|+|c-a|)^2$

3. $SQRT\_ABS\_SUM(a, b, c) = (|a-b|+|b-c|+|c-a|)^{0.5}$

4. $SEC\_DIFF(a, b, c) = |a - 2b + c|$

Note that ABS_SUM is submodular for 8-label 3-clique potentials while the other three are not. On nonsubmodular potentials MLGC outputs what can be considered to be submodular approximations. Our experiments have focussed on the energy of the primal solution inferred and the time taken for inference. Figure 3 shows the results of our experiments. The first number in the image below MLGC output is the value of primal while the second number is the value of dual. Interestingly MLGC solution is optimal also for SQRT_ABS_SUM potential. Except for SEC_DIFF potential for which DD outputs lower energy, energy outputted by MLGC is significantly less than the other methods. Other than MPI which probably converges to bad local minima relatively quickly, MLGC is an order of magnitude faster than the other methods.

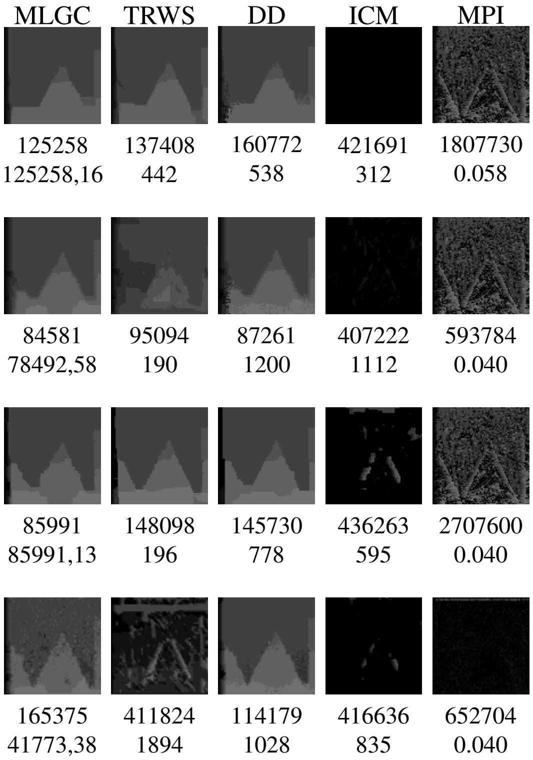

Figure 3: Comparison on the sawtooth dataset [2]. The numbers below each result represents primal energy followed by time taken for inference in seconds. For MLGC however it is primal energy, followed by dual followed by time. The four rows represent results with clique potential ABS_SUM, SQR_ABS_SUM, SQRT_ABS_SUM, SECOND_DIFF respectively.

We know that MLGC will output optimal solutions for submodular potentials and comparative run data reported above shows MLGC is an order of magnitude faster than the other direct methods. Our preliminary first cut use of MLGC on nonsubmodular potential functions has indicated that it is better than the other direct methods and may sometimes even output optimal results. Nevertheless MLGC as is currently formulated is essentially targeted for providing optimal solutions to submodular multi-label multi-clique problems. We believe that it is possible to design variations of MLGC that output good approximate solutions for large nonsubmodular problems and at the same time circumvent the m^k bottleneck. One area to focus on is designing data structures which would enable residual capacity calculation without enumerating slacks of all m^k dual feasibility constraints. The other area of investigation is exploring the possibility of identifying a small subset of the m^k constraints to which residual capacity calculation could be limited. We consider design of practical approximation algorithms based on MLGC as a problem with potentially large pay off.

6. Conclusion

We have presented in this paper an optimal algorithm for submodular multi-label multi-clique MRF-MAP inference problems. If the number of labels and clique size are fixed the algorithm's time complexity is strongly polynomial. The earlier work on optimal inference has essentially handled 2-label 2-clique [10], multi-label 2-clique [11] and 2-label multi-clique problems [3]. With the proposed algorithm for optimal inference in multi-label multi-clique problems we have effectively closed the loop in the sense that optimal inference for submodular clique potentials is now possible for all classes of MRF-MAP problems. Since recent techniques for handling 2-label nonsubmodular clique potentials have essentially involved finding some kind of submodular approximation [14], we believe that the current work can be the basis for solving the multi-label nonsubmodular problems by finding suitable submodular approximation. While MLGC is guaranteed to output an optimal solutions regardless of the size of cliques or the number of labels involved, developing efficient algorithms for residual capacity calculation, both by new algorithmic insights and better implementations on multi-core architectures is an area of research we advocate. Also, we work under the assumption that all m^k configurations on a clique are equally likely. Rother et al. [24] have shown that sparsity in clique in many computer vision problems can be exploited for MRF-MAP problems with large size cliques. MLGC can also potentially take advantage from such an approach.

Acknowledgement: We would like to thank Srikumar Ramalingam for interesting and useful discussions. Chetan Arora is supported by postdoctoral fellowship of the Council for Higher Education in Israel.

References

[1] Darwin Framework, Version 1.1.2. http://drwn.anu.edu.au/.

[2] MRF energy minimization software - Version 2.1. http://vision.middlebury.edu/MRF/code/.

[3] C. Arora, S. Banerjee, P. Kalra, and S. Maheshwari. Generic Cuts: An efficient optimal algorithm for submodular MRF-MAP problems with higher order cliques. In *ECCV*, pages 17–30, 2012.

[4] J. Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, B-48:259–302, 1986.

[5] G. Birkhoff. Rings of sets. *Duke Mathematical Journal*, 3(3):443–454, 1937.

[6] E. Boros and A. Gruber. On quadratization of pseudo-boolean functions. In *ISAIM*, 2012.

[7] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *PAMI*, 26(9):1124–1137, 2004.

[8] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23:2001, 1999.

[9] A. Fix, A. Gruber, E. Boros, and R. Zabih. A graph cut algorithm for higher-order markov random fields. In *ICCV*, pages 1020–1027, 2011.

[10] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc.*, 51:271–279, 1989.

[11] H. Ishikawa. Exact Optimization for Markov Random Fields with Convex Priors. *PAMI*, 25(10):1333–1336, 2003.

[12] H. Ishikawa. Transformation of General Binary MRF Minimization to the First-Order Case. *PAMI*, 33:1234–1249, 2011.

[13] S. Iwata and J. B. Orlin. A simple combinatorial algorithm for submodular function minimization. In *SODA*, pages 1230–1237, 2009.

[14] F. Kahl and P. Strandmark. Generalized roof duality for pseudo-boolean optimization. In *ICCV*, pages 255–262, 2011.

[15] P. Kohli and P. H. S. Torr. Dynamic Graph Cuts for Efficient Inference in Markov Random Fields. *PAMI*, 29:2079–2088, 2007.

[16] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive Computation and Machine Learning. MIT Press, 2009.

[17] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568–1583, 2006.

[18] V. Kolmogorov. Minimizing a sum of submodular functions. *Discrete Appl. Math.*, 160(15):2246–2258, 2012.

[19] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts-a review. 2007.

[20] N. Komodakis and N. Paragios. Beyond pairwise energies: Efficient optimization for higher-order MRFs. In *CVPR*, pages 2985–2992, 2009.

[21] N. Komodakis, N. Paragios, and G. Tziritas. MRF Energy Minimization and Beyond via Dual Decomposition. *PAMI*, 33(3):531 –552, 2011.

[22] K. Murota. *Discrete Convex Analysis*. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 2003.

[23] S. Ramalingam, P. Kohli, K. Alahari, and P. Torr. Exact inference in multi-label crfs with higher order cliques. In *CVPR*, pages 1–8, 2008.

[24] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, pages 1382–1389, 2009.

[25] D. Sontag, T. Meltzer, A. Globerson, Y. Weiss, and T. Jaakkola. Tightening LP relaxations for MAP using message-passing. In *UAI*, pages 503–510, 2008.

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