Supplementary Material

1 Completely Worked Out Example

Labeling	Potential
bbbb	70.000000
abbb	141.000000
babb	141.000000
aabb	71.000000
bbab	141.000000
abab	141.000000
baab	200.000000
aaab	141.000000
bbba	141.000000
abba	200.000000
baba	141.000000
aaba	141.000000
bbaa	141.000000
abaa	141.000000
baaa	141.000000
aaaa	40.000000

Figure 1: Input clique potential

Labeling	Potential
bbbb	0.000000
abbb	101.000000
babb	71.000000
aabb	31.000000
bbab	71.000000
abab	101.000000
baab	130.000000
aaab	101.000000
bbba	71.000000
abba	160.000000
baba	71.000000
aaba	101.000000
bbaa	71.000000
abaa	101.000000
baaa	71.000000
aaaa	0.000000

Figure 2: Reparametrized clique potential

Monomial Number	Coefficient
0	70.000000
1	71.000000
2	71.000000
3	-141.000000
4	71.000000
5	-71.000000
6	-12.000000
7	82.000000
8	71.000000
9	-12.000000
10	-71.000000
11	82.000000
12	-71.000000
13	12.000000
14	12.000000
15	-124.000000

Figure 3: PBF generated from input clique potential

Consider a problem having single clique of size 4. The clique potential penalizing a labeling as per 100 times the square root of number of edges in the labeling in which cost of labeling $aabb$ is reduced to 71 (instead of 141 assigned by square root of number of edges) and the cost of uniform labeling is set arbitrarily at 70 and 40. The energy values for the aforesaid clique potential is listed in Figure 1. It can be verified that the clique potential is submodular. The higher order pseudo Boolean function (PBF) generated from the example clique potential as used by IQ is shown in Figure 3. The positive coefficients in pseudo Boolean function indicates that the potential is not regular. The reparametrization done by GC, first increases the V variable of 0^{th} b ball by 70 and of 0^{th} a ball by 40 (along with associated changes in unary potential) to arrive at reparametrized clique potential as shown in Figure 2. It can be verified that the reparametrized clique potential remains submodular.

node index	$E(a)$	$E(b)$
0	50	0
1	0	200
2	100	0
3	100	0

Figure 4: Unary potential

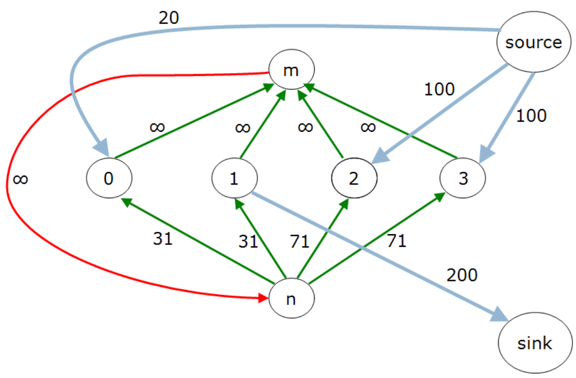

Figure 5: Flow graph for the example

Assume that the unary potential of the 4 nodes before reparametrization is as shown in Table 4. The flow graph for the reparametrized clique potential along with terminal edge capacities and the residual edge capacities is as shown in Figure 5. It may be noted that during reparametrization the b ball at 0^{th} is raised by 70 and a ball by 40. The net height of the a and b are therefore at 90 and 70 respectively. In the flow graph the $0th$ node is therefore connected to source by the capacity equal to difference in heights of the ball that is

$90 - 70 = 20$. The graph of Figure 5 should be considered as the residual graph when flow in all edges is). The labels on the conjugate edges have been set as per "When there is no flow in either of the edges of a conjugate pair, then the residual graph has two edges corresponding to the two conjugate edges. One emanating from the auxiliary node (n type) has capacity equal to the capacity of the conjugate edge pair. Capacity of the other is infinity. In this case the requirement that only one of the conjugate edges has zero flow is ensured by restricting augmenting flow on paths from s to t that include only one of the edges of a conjugate pair in the residual graph.".

Labeling	Potential
bbbb	0.000000
abbb	101.000000
babb	40.000000
aabb	0.000000
bbab	102.000000
abab	132.000000
baab	130.000000
aaab	101.000000
bbba	71.000000
abba	160.000000
baba	40.000000
aaba	70.000000
bbaa	102.000000
abaa	132.000000
baaa	71.000000
aaaa	0.000000

Figure 6: Revised slacks

Flow is first pushed along the path $source \to 2 \to m \to n \to 1 \to sink$. As per the residual capacities of edges on the path its value is 31. This makes the constraint *aabb* tight (i.e. its slack becomes 0). Table 6 contains the revised slacks of all the constraints after this flow push. The new revised residual capacities of the pairs of conjugate edges are given in Table 7 and the residual graph with the residual capacities on individual edges is given in Figure 8. Note that the edges in the residual graph are as per the earlier rule quoted in the previous paragraph and "The conjugate edge with non zero flow emanates from n type auxiliary node. In this case the residual graph has two edges. The edge in direction from an auxiliary node (n type) to a pixel node has capacity equal to the residual capacity of the conjugate edge pair, and the edge in the direction from a pixel node to an auxiliary node has capacity equal to the flow in the conjugate edge pair. In the other case, the capacity of the residual edge from the pixel node to the auxiliary node (m type) is infinity, and the reverse direction edge has capacity equal to the flow towards the auxiliary node (m type) in the conjugate edge pair in question.". As per Lemma 4, this saturation allows the flow through path $source \to 0 \to m \to n \to 1 \to sink$ since both nodes 0 and 1 are present in constraint *aabb* and sending flow from $0 \to 1$ does not affect the slack in the tight constraint. The capacity of flow is contextually constrained at 40 due to constraints *babb* and *baba*. However because of constraints posed by terminal edge capacity 0^{th} node can send flow of only 20 to 1^{st} node saturating edge $source \to 0$. No more flow can be further sent in the flow graph and the maximum flow arrived at is $31 + 20 = 51$. The labeling recovered by putting nodes having unsaturated path from *source* in S set is *aabb*. The cut edges incident at 0^{th} and 1^{st} node are covered by set of constraints $\{\{abbb\}, \{babb\}\}$ or $\{\{aabb\}\}$, minimum of which (that is $\{\{aabb\}\}$) decides the cost of cut which is 31. Edge $source \to 0$ is also on the cut thereby bringing the total value of the cut to 51 which is equal to the flow in the flow graph.

It is interesting to note that the same example graph when run using IQ gives all nodes unlabeled.

Conjugate edge pair incident at	Residual cap
0	0
1	0
2	71
3	71

Figure 7: Revised residual capacities

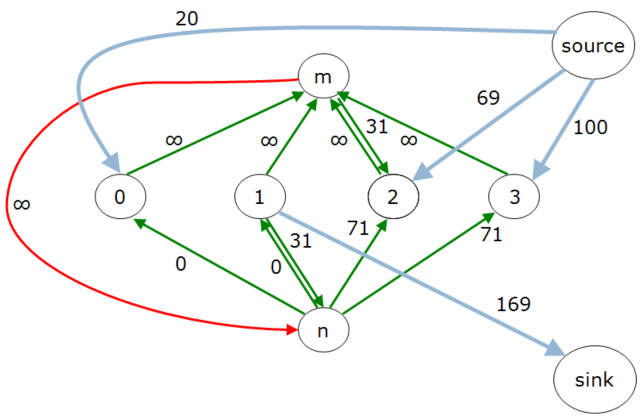

Figure 8: Revised flow graph

2 Formulation of the Dual

We denote the set of pixels in an image by $\mathcal{P}$, and the set of higher order cliques on the pixel set by $\mathcal{C}$. The label of a pixel p is denoted by l^p and the labeling configuration of clique $\mathbf{c}$ by $\mathbf{l_c}$. Finding a labeling with maximum *a posteriori* probability (MAP) in such a MRF can be shown to be equivalent to minimizing energy of following kind:

$$E(\mathbf{l_f}) = \sum_{p \in \mathcal{P}} D_p(l_p) + \lambda \sum_{\mathbf{c} \in \mathcal{C}} W_{\mathbf{c}}(\mathbf{l_c})), \tag{1}$$

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 $\mathbf{c}$.

The LP formulation for MRF-MAP given below follows Kleinberg and Tardos [10]. Any pixel can take a label from the set $\mathcal{L} = a, b$ of possible labels, and $l_{\mathbf{c},p,a}$ is that subset of labeling configurations $\mathbf{l_c}$ of clique $\mathbf{c}$ in which the label of pixel p is a. We introduce a binary variable X_p^a whose value is 1/0 whenever pixel p is assigned label a/b respectively. Similarly the binary variable $Y_{\mathbf{c}}^{\mathbf{l_c}}$ takes value 1 whenever clique $\mathbf{c}$ is assigned label configuration $\mathbf{l_c}$ and is 0 otherwise. Let $W_{\mathbf{c}} : \mathcal{L}^k \to \mathcal{R}$ be the clique potential function giving the penalty of labeling the pixels of clique $\mathbf{c}$ by $\mathbf{l_c}$. The MRF-MAP equation (1) can be equivalently written as following linear program:

$$\min_{X_p^a, Y_{\mathbf{c}}^{\mathbf{l_c}}} \sum_{p \in \mathcal{P}} \sum_{a \in \mathcal{L}} C_p^a X_p^a + \sum_{\mathbf{c} \in \mathcal{C}} \sum_{\mathbf{l_c} \in \mathcal{L}^k} W_{\mathbf{c}}(\mathbf{l_c}) Y_{\mathbf{c}}^{\mathbf{l_c}} \tag{2}$$

subject to

$$\sum_{a \in \mathcal{L}} X_p^a = 1, \qquad p \in \mathcal{P},$$

$$\sum_{\forall l_{\mathbf{c},p,a}} Y_{\mathbf{c}}^{l_{\mathbf{c},p,a}} = X_p^a, \qquad \mathbf{c} \in \mathcal{C},\ p \in \mathbf{c}, ; a \in \mathcal{L},$$

where $l_{\mathbf{c},p,a}$ denotes those $\mathbf{l_c}$ labeling configuration with label a at p, and

$$X_p^a \geq 0 \qquad , \qquad Y_{\mathbf{c}}^{\mathbf{l_c}} \geq 0.$$

Lagrangian of the primal in equation (2) can be written as

$$\mathcal{G} = \underset{U_p, \lambda_{\mathbf{c}}^{\mathbf{l_c}}, V_{\mathbf{c},p,a}}{\arg\max} \ \underset{X_p^a, Y_{\mathbf{c}}^{\mathbf{l_c}}}{\arg\min} \Bigg[\sum_{p \in \mathcal{P}} \sum_{a \in \mathcal{L}} C_p^a X_p^a + \sum_{\mathbf{c} \in \mathcal{C}} \sum_{\mathbf{l_c} \in \mathcal{L}^k} W_{\mathbf{c}}(\mathbf{l_c}) Y_{\mathbf{c}}^{\mathbf{l_c}} \tag{3}$$

$$+ \sum_{p \in \mathcal{P}} U_p \left(1 - \sum_{a \in \mathcal{L}} X_p^a\right) + \sum_{\mathbf{c} \in \mathcal{C}, p \in \mathbf{c}, a \in \mathcal{L}} V_{\mathbf{c},p,a} \left(X_p^a - \sum_{\forall l_{\mathbf{c},p,a}} Y_{\mathbf{c}}^{l_{\mathbf{c},p,a}}\right) \tag{4}$$

$$- \sum_{p \in \mathcal{P}, a \in \mathcal{L}} \lambda_p^a X_p^a - \sum_{\mathbf{c} \in \mathcal{C}, \mathbf{l_c} \in \mathcal{L}^k} \lambda_{\mathbf{c}}^{\mathbf{l_c}} Y_{\mathbf{c}}^{\mathbf{l_c}} \Bigg]. \tag{5}$$

Rearranging terms with variables X_p and $Y_{\mathbf{c}}$

$$\mathcal{G} = \underset{U_p,\lambda_{\mathbf{c}}^{\mathbf{l_c}},V_{\mathbf{c},p,a}}{\arg\max} \ \underset{X_p^a,Y_{\mathbf{c}}^{\mathbf{l_c}}}{\arg\min} \Bigg[\sum_p U_p \tag{6}$$

$$+ \sum_{p\in\mathcal{P},a\in\mathcal{L}} X_p^a \left(C_p^a - \lambda_p^a - U_p + \sum_{\forall \mathbf{c}\ s.t.\ p\in\mathbf{c}} V_{\mathbf{c},p,a} \right) \tag{7}$$

$$+ \sum_{\mathbf{c}\in\mathcal{C},\mathbf{l_c}\in\mathcal{L}^n} Y_{\mathbf{c}}^{\mathbf{l_c}} \left(W_{\mathbf{c}}(\mathbf{l_c}) - \lambda_{\mathbf{c}}^{\mathbf{l_c}} - \sum_{p\in\mathbf{c}} V_{\mathbf{c},p,l_{\mathbf{c}}^p} \right) \Bigg], \tag{8}$$

where ${\mathbf{l_c}}^p$ is the label of point p in label configuration $\mathbf{l_c}$.
The lagrangian will be unbounded if the partial derivative with respect to X_p and $Y_{\mathbf{c}}$ is not zero. In our case these will be coefficients of X_p and $Y_{\mathbf{c}}$. Setting them to zero in lagrangian will give us our dual objective function as:

$$g(U,V,\lambda) = \underset{U,V,\lambda}{\arg\max} \sum_{p\in\mathcal{P}} U_p.$$

The requirement of partial derivative with respect to X_p equal to zero gives us our first dual feasibility constraint

$$C_p^a - \lambda_p^a - U_p + \sum_{\forall \mathbf{c}\ s.t.\ p\in\mathbf{c}} V_{\mathbf{c},p,a} = 0, \qquad p\in\mathcal{P},\ a\in\mathcal{L}.$$

Since $\lambda_p^a \geq 0$

$$U_p \leq h_p^a \qquad p\in\mathcal{P},\ a\in\mathcal{L}. \tag{9}$$

where

$$h_p^a = C_p^a + \sum_{\forall \mathbf{c}\ s.t.\ p\in\mathbf{c}} V_{\mathbf{c},p,a}.$$

Similarly second feasibility constraint can be arrived setting partial derivative of $Y_{\mathbf{c}}$ equal to zero

$$W_{\mathbf{c}}^{\mathbf{l_c}} - \lambda_{\mathbf{c}}^{\mathbf{l_c}} - \sum_{p\in\mathbf{c}} V_{\mathbf{c},p,{\mathbf{l_c}}^p} \qquad \mathbf{c}\in\mathcal{C}\ and\ \mathbf{l_c}\in\mathcal{L}^k$$

Since $\lambda_{\mathbf{c}}^{\mathbf{l_c}} \geq 0$

$$\sum_{p\in\mathbf{c}} V_{\mathbf{c},p,{\mathbf{l_c}}^p} \leq W_{\mathbf{c}}(\mathbf{l_c}), \qquad \mathbf{c}\in\mathcal{C},\ \mathbf{l_c}\in\mathcal{L}^k. \tag{10}$$

Complimentary slackness conditions can be written as

$$X_p^a > 0 \qquad \Rightarrow \qquad U_p = h_p^a, \tag{11}$$

$$Y_{\mathbf{c}}^{\mathbf{l_c}} > 0 \qquad \Rightarrow \qquad \sum_{p\in\mathbf{c}} V_{\mathbf{c},p,{\mathbf{l_c}}^p} = W_{\mathbf{c}}(\mathbf{l_c}). \tag{12}$$

Assuming the cost of assigning uniform labeling (all a's or all b's) to clique as zero gives us the following constraint

$$\sum_{p\in\mathbf{c}} V_{\mathbf{c},p,a} = 0, \qquad \mathbf{c}\in\mathcal{C},\ a\in\mathcal{L}. \tag{13}$$

3 Uniform Labeling Cost and Submodularity Constraints

The dual feasibility constraints as derived in Section 2 are:

$$U_p \leq h_p^a \qquad p \in \mathcal{P},\ a \in \mathcal{L} \tag{14}$$

where

$$h_p^a = C_p^a + \sum_{\forall \mathbf{c}\ s.t.\ p \in \mathbf{c}} V_{\mathbf{c},p,a}$$

, and

$$\sum_{p \in \mathbf{c}} V_{\mathbf{c},p,\mathbf{l_c}^p} \leq W_{\mathbf{c}}(\mathbf{l_c}) \qquad \mathbf{c} \in \mathcal{C},\ \mathbf{l_c} \in \mathcal{L}^k. \tag{15}$$

Suppose we replace the variable $V_{\mathbf{c},p,a}$ of clique $\mathbf{c}$ by $V_{\mathbf{c},p,a} + \delta$ in all dual feasibility constraints in which it occurs and subtract δ from both sides of constraints defined by equation 15 in which $V_{\mathbf{c},p,a}$ occurs then effectively the only change that have occurred in the dual feasibility constraints are the following:

1. The l.h.s of equation 14 for pixel $\mathbf{c}$ which defines the value of dual variable h_p^a has an additional term δ, and
2. In all feasibility constraints defined by equation 15 for clique $\mathbf{c}$ in which $V_{\mathbf{c},p,a}$ occurs, the r.h.s. has become $W_{\mathbf{c}}(\mathbf{l_c}) - \delta$.

In effect we have increased the height of the ball a in well for pixel p, that is we have increased the cost of assigning label a to pixel p in clique $\mathbf{c}$ and compensated for that increase by decreasing the contextual cost of this assignment in clique $\mathbf{c}$. It is easy to establish that from the perspective of the optimal solution to the optimization problem, the value of the optimal solution and the assignment of labels in that optimal solutions in both versions of the problem are the same. The transformation can be seen as re-parametrization of the energy function [19].

If δ is so chosen that one of the contextual costs term becomes zero we can look upon it as tightening of a dual feasibility constraint. We call this process of tightening at least one constraint and raising the height of a ball in a well the normalization step. If the dual feasibility constraint that gets tightened corresponds to the uniform labeling of clique pixels being labeled a then we are done. If not, there must exist another pixel q in clique $\mathbf{c}$ for which label assignment is a in the constraints which still have slack. Limiting attention to just these constraints if the normalization step is repeated in clique $\mathbf{c}$ for pixel q and ball a then at least one more dual feasibility constraint will become tight. Since labeling costs in a clique are submodular, Lemma 2 will apply and the constraint which is union of the two constraints will also be tight. The normalization step can continue to be applied till the constraint corresponding to uniform labeling of all a becomes tight. It is easy to see by symmetry that the above reasoning and normalization step can be now carried out with $V_{\mathbf{c},p,b}$ variables resulting in a system in which both uniform costs are zero.

We can show that if an energy function was submodular before transformation it will remain submodular after transformation also as follows. W.l.g. we interpret constraints/labelings-energy as sets consisting of pixels labeled a in all cliques. The set function assigns a value to such set equal to cost of the corresponding labeling. We are given that such set functions are submodular prior to any normalization transformation.

Consider an increase in one of the V variable corresponding to ball p_a as a part of the normalization process. Prior to this increase since the energy function was submodular, i.e. we had

$$E(\mathbf{X}) + E(\mathbf{Y}) \geq E(\mathbf{X} \cup \mathbf{Y}) + E(\mathbf{X} \cap \mathbf{Y})\ \forall \mathbf{X}, \mathbf{Y}.$$

Now the following cases arise:

1. p_a was not part of either $\mathbf{X}$ or $\mathbf{Y}$
   In this case all the constraints $E(\mathbf{X})$, $E(\mathbf{Y})$, $E(\mathbf{X} \cup \mathbf{Y})$ and $E(\mathbf{X} \cap \mathbf{Y})$ does not change. Both LHS and RHS remain same and equation will continue to remain satisfied.
2. p_a was part of only one of $\mathbf{X}$ or $\mathbf{Y}$
   In this case p_a will be part of set $\mathbf{X} \cup \mathbf{Y}$ but not $\mathbf{X} \cap \mathbf{Y}$. Therefore $E(\mathbf{X})$ or $E(\mathbf{Y})$ decrease by δ and $E(\mathbf{X} \cup \mathbf{Y})$ decrease by δ. The LHS and RHS of the equation decrease by same amount (δ) and equation remains satisfied.

3. p_a was part of both $\mathbf{X}$ and $\mathbf{Y}$
   In this case p_a will be part of set $\mathbf{X} \cup \mathbf{Y}$ as well as $\mathbf{X} \cap \mathbf{Y}$. Therefore all E ($E(\mathbf{X})$, $E(\mathbf{Y})$, $E(\mathbf{X} \cup \mathbf{Y})$ and $E(\mathbf{X} \cap \mathbf{Y})$) decrease by amount δ. The LHS and RHS of the equation decrease by same amount (2δ) and equation remains satisfied

Therefore any change in the V variable is nothing but an equivalent energy transformation which maintains submodularity property.

Since there are 2^k dual feasibility constraints for a clique time taken for normalization per clique will be $O(2^k)$. Under the assumption that the number of cliques is of the same order as n, the number of pixels, the total normalization time is $O(n2^k)$.

4 Maxflow Mincut Theorem

Lemma 1 shows that flow in a gadget based flow graph cannot exceed the capacity of any (S, T) cut.

Lemma 1. *Let (S, T) be the cut in which S is the set of nodes reachable from s in the residual graph when flow is maximal. Value of max flow is equal to the sum of flow in saturated edges out of s, saturated edges into t in the (S, T) cut and flow in conjugate edges from auxiliary nodes in S to nodes in T, and is less than or equal to the capacity of the (S, T) cut.*

Proof. Let $f = (f_{ij})$ be a flow in the flow graph and let (S, T) be any cut. Summing up the flow conservation equations for all nodes $i \in S$ we get

$$v = \sum_{i \in S} (\sum_j f_{ij} - \sum_j f_{ji}) = \sum_{i \in S} \sum_{j \in S} (f_{ij} - f_{ji}) + \sum_{i \in S} \sum_{j \in T} (f_{ij} - f_{ji}) = \sum_{i \in S} \sum_{j \in T} (f_{ij} - f_{ji}),$$

where v is the flow entering t. In effect v is the net flow through any (S, T) cut. Since $f_{ji} \geq 0$

$$v \leq \sum_{i \in S} \sum_{j \in T} f_{ij}.$$

Note that flow in all edges other than conjugate edges in the (S, T) cut is less than their capacities. Also, since the r.h.s. of a dual feasibility constraints is larger than or equal to the sum of flows in all the conjugate edges that occur in it's l.h.s., the sum of flows in all the conjugate edges in the (S, T) cut is less than or equal to the capacity of the smallest cost edge cover. Therefore, it follows that flow in the flow graph is always less than or equal to the capacity of any (S, T) cut.

Consider the scenario when flow equal to the maximum value flows through the edges of the flow graph. Since flow can not be incremented any further, no path can exist between s and t in the residual graph constructed in the presence of this flow. Let S be the set of nodes reachable from s in this residual graph and let T be the rest of the nodes of the flow graph. Effective flow across the cut from S to T is equal to the sum of flow in the flow graph edges from S to T minus the sum of flow in the flow graph edges from T to S. Since only edges with finite capacity constraints in the flow graph are conjugate edges, this (S, T) cut set can have only conjugate edges with zero residual capacity or saturated edges incident at s and t. Other than saturated edges incident at s and saturated edges from S side nodes to t the flow graph edges from S to T will be directed out of only auxiliary nodes. Also, note that an S to T edge emanating out of an auxiliary node implies that the other auxiliary node of that clique is also on the S side. This is because the infinite capacity edge between the two auxiliary nodes has non zero flow and hence there are non zero capacity edges between the two auxiliary nodes in both directions in the residual graph. If the auxiliary node is of n type, then flow in the paired conjugate edge $p \to m$ is zero and the value of effective flow across the cut (S, T) is dependent only on the flow in the cut edge emanating out of the n type auxiliary node in question. If, on the other hand, the auxiliary node is of m type and no flow can be sent in the residual edge $m \to p$, then also the effective flow across the cut (S, T) is dependent only on the flow in the cut edge because if the edge $p \to m$ has non zero flow then the edge $n \to p$ has zero flow. If $p \to m$ has zero flow then the only case which may impact effective flow across the cut is when the edge $n \to p$ has non zero flow. In this situation, n type node on side S is the first case discussed. Therefore, flow in the flow graph is equal to the sum of flow in saturated edges out of s and possibly into t in the (S, T) cut and flow in conjugate edges from auxiliary nodes in S to T.

Note that each of such conjugate edge from S to T has zero residual capacity and therefore at least one dual feasibility constraint in which it participates is tight. In effect there are conjugate edge covers of the edges in the (S, T) cut in which all dual feasibility constraints are tight. Is max flow equal to min capacity cut? Consider the smallest cost cover among the conjugate edge covers of the edges in the (S, T) cut. If each conjugate edge in the cut is covered by only one constraint in smallest cost cover then it follows that max flow is equal to the capacity of the (S, T) cut which implies that max flow is equal to the min capacity cut. This may not hold in general as there can be cases when a conjugate edge in the cut is covered by two or more tight constraints in the smallest cost cover. In such cases flow in the edge gets counted more than once in the cost of the cover, and max flow is less than capacity of the min cost cover in general.

□

We now show that when dual feasibility constraints are submodular then max flow in a flow graph is always equal to the value of the (S, T) cut of Lemma 1.

Since $\mathbf{l_c}$ gets defined by the assignment of one of the two possible labels to pixels in clique $\mathbf{c}$, it can be equivalently looked upon w.l.g. as a subset of pixels of clique $\mathbf{c}$ labeled 'a'. $W_{\mathbf{c}}$, can therefore be considered to be a set function. We assume all functions $W_{\mathbf{c}}$ are submodular, i.e. for all $\mathbf{c} \in \mathcal{C}$, and $X, Y \in \mathbf{c}$

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

Lemma 2. *If $W_{\mathbf{c}}(\mathbf{l_c})$ is submodular for all $\mathbf{c}$ in $\mathcal{C}$, then whenever in the flow graph there exists a conjugate edge that has residual capacity zero and is covered by two tight dual feasibility constraints corresponding to pixel sets X and Y of some clique, the dual feasibility constraint corresponding to pixel set $X \cup Y$ is also tight.*

Proof. Let $\mathbf{X}$ and $\mathbf{Y}$ be the pixel sets in clique $\mathbf{c}$ such that $\mathbf{X} \cap \mathbf{Y}$ is not empty and the dual feasibility constraints corresponding to them are tight in the presence of flow f in the flow graph. Because of submodularity we have

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

Let the conjugate edges with non zero flow corresponding to sets $\mathbf{X}$ and $\mathbf{Y}$ be denoted by $x_1, x_2, ..., x_p$ and $y_1, y_2, ..., y_l$ respectively. Since the dual constraints are tight we have

$$\sum f_{x_i} = W_{\mathbf{c}}(\mathbf{X}), and \sum f_{y_i} = W_{\mathbf{c}}(\mathbf{Y}).$$

Therefore

$$\sum f_{x_i} + \sum f_{y_i} \geq W_{\mathbf{c}}(\mathbf{X} \cup \mathbf{Y}) + W_{\mathbf{c}}(\mathbf{X} \cap \mathbf{Y}).$$

It can be shown that the dual feasibility constraint corresponding to $\mathbf{X} \cap \mathbf{Y}$ is satisfied. Therefore we have

$$W_{\mathbf{c}}(\mathbf{X} \cap \mathbf{Y}) \geq \sum_{e \in (\mathbf{X} \cap \mathbf{Y})} f_e.$$

Therefore

$$\sum f_{x_i} + \sum f_{y_i} - \sum_{e \in (\mathbf{X} \cap \mathbf{Y})} f_e \geq W_{\mathbf{c}}(\mathbf{X} \cup \mathbf{Y}).$$

The l.h.s. of the above is nothing but the sum of flow in the conjugate edges covered by the dual constraint corresponding to $\mathbf{X} \cup \mathbf{Y}$. Since flow is feasible we have

$$W_{\mathbf{c}}(\mathbf{X} \cup \mathbf{Y}) = \sum_{e \in (\mathbf{X} \cup \mathbf{Y})} f_e.$$

□

Lemma 2 allows us to infer that when costs are submodular, the smallest cost cover for conjugate edges in the (S, T) cut corresponding to max flow, consists of a set of dual constraints such that each conjugate edge in the (S, T) cut is covered by only one constraint from that set. Note that our flow formulation is such that all complimentary slackness conditions other than those given by equation (8) are always satisfied. Also, the dual constraints corresponding to the smallest cost cover are tight, i.e. equations (12) for them are satisfied with equality for all cliques on the cut. The cliques not on the cut are labeled uniformly, i.e. all nodes of such cliques in S are labeled a and those in T are labeled b. Uniform labeling constraint, as shown in equation (14) is flow conservation constraint, and is always satisfied. We therefore have that when flow is maximal all complimentary slackness conditions for all cliques are satisfied. Primal and dual solutions consistent with them must be optimal.

We, therefore, have

Theorem 3. *When costs are submodular, in the flow graph corresponding to the dual optimization problem, max flow is equal to min cut and corresponding primal and dual solutions are optimal.*

Proof. Lemma 2 allows us to infer that in the smallest cost cover for conjugate edges in the (S, T) cut corresponding to max flow is each conjugate edge is covered by only one dual feasibility constraint when the r.h.s. of all constraint are submodular. The smallest cost cover is then the min cost cover and max flow is equal to the capacity of the min cost cover.

Note that our flow formulation ensures that all complimentary slackness conditions other than equation (8) are always satisfied. For a maximum flow situation, equation (12) is satisfied with equality for all cliques on the cut which is nothing but another representation of equation (8). For the cliques not on the cut, we give uniform labeling. Uniform labeling constraint as shown in equation (14) is flow conservation constraint and is always satisfied. We can therefore state that when flow is maximum all complimentary slackness conditions are satisfied and primal and dual must be optimal.

□

5 Complexity Analysis

If we use Edmonds and Karp's shortest path augmentation strategy along with ordering of the paths of the same length lexicographically then the number of augmenting path iterations will be $O(|V|.|E|)$ and the time to find the shortest augmenting path and updating the residual capacities of the edges in the residual graph be $O(|E|)$. Note that there is the possibility of pushing flow in a saturated conjugate edge in the forward direction k times before the edge getting dropped. The total number of iterations before the path length increases is still bounded by $O(|E|)$. If the size of a clique is given by k, the set of cliques by $\mathcal{C}$, and the number of pixels by n, then $|V|$ and $|E|$ are $O(n + |\mathcal{C}|)$ and $O(k|\mathcal{C}|)$ respectively. Since the number of dual feasibility constraints per clique can be $O(2^k)$, residual capacity computation of each edge will require $O(2^k)$ steps. Therefore, the complexity of the max flow algorithm implemented Edmonds and Karp way is $O(2^k n(k|\mathcal{C}|)^2)$. Under the assumption that submodular functions for computer vision problems are locally defined with $n \approx |\mathcal{C}|$, the complexity can also be written as $O(2^k k^2 n^3)$.

For IQ algorithm which involves reduction of higher order Boolean functions to second order and using QPBO for optimization, the process involves creation of $O(2^k)$ auxiliary nodes per clique in the reduction phase. Since the resultant flow graph is dense the number of edges can be $O(2^{2k})$ per clique. If $\mathcal{C}$ is the set of cliques, the max flow part of the algorithm's time complexity will be $O(2^{5k} n|\mathcal{C}|^2)$ or $O(2^{5k} n^3)$ if $n \approx |\mathcal{C}|$, assuming that the max flow algorithm is based on Edmonds and Karp. This analysis has been done primarily to estimate the type of speed up one should expect using our algorithm over algorithms based on the reduction technique like IQ.

6 Proof of Various Lemmas

Lemma 4. *A saturated conjugate edge $n \to p$ corresponding to a clique $\mathbf{c}$ in presence of flow f cannot be in an augmenting path if the only tight dual feasibility constraint covering it covers no other edge, or if the intersection of all tight dual feasibility constraints covering it contains no other edge.*

Proof. Suppose the intersection of all tight dual feasibility constraints covering $n \to p$ contains at least one more edge. This edge is either an n type edge $n \to q$ or an m type edge $r \to m$. If the edge is $n \to q$ then both path fragments $q \to n \to p$ and $p \to n \to q$ have residual capacity greater than zero. Since pushing of flow in the path fragment $q \to n \to p$ leaves the tight dual feasibility constraints tight, residual capacity of $q \to n \to p$ is equal to the minimum of the slacks of non tight dual feasible constraints covering $n \to p$ and $n \to q$. Similarly residual capacity of the path fragment $r \to m \to n \to p$ is the minimum of the slacks of non tight dual feasible constraints covering $r \to m$ and $n \to p$. Therefore a path from s to t which contains the path fragment $q \to n \to p$ or $r \to m \to n \to p$ as the case may be with all other edges having residual capacities greater than zero will be an augmenting path.

□

Lemma 5. *Let the shortest augmenting path length from s to t after k shortest augmenting path flow augmentations be denoted by $\delta^k(s,t)$. Then $\delta^{k+1}(s,t) \geq \delta^k(s,t)$.*

Proof. For the purposes of the proof we specify the shortest augmenting path length from s to a node p and from p to t in clique $\mathbf{c}$ after k such iterations by $\delta^k(s, \mathbf{c}, p)$ and $\delta^k(t, \mathbf{c}, p)$ respectively. Consider the flow graph after the k^{th} flow augmentation.It is easy to see that if the path fragments in the shortest augmenting path for the $(k+1)^{th}$ augmentation are not in the cliques through which the k^{th} augmenting path passes then $\delta^{k+1}(s,t) \geq \delta^k(s,t)$. For the possibility that the $(k+1)^{th}$ shortest augmenting path to be smaller than k^{th} shortest augmenting path it is necessary that there be at least one clique in which in which the two paths use different path fragments.

We will consider the case when the two paths share exactly one clique in which the path fragments are different in the two paths. Let that clique be $\mathbf{c}$ and let the path fragment used in the k^{th} augmenting path be from p to q. One possibility is that in the kth augmentation the path fragment from p to q was saturated and in the $(k+1)^{th}$ iteration path fragment from q to p was in the augmenting path. In this case since $\delta^{k+1}(s,\mathbf{c},q) \geq \delta^{k}(s,\mathbf{c},q)$ and $\delta^{k+1}(t,\mathbf{c},p) \geq \delta^{k}(t,\mathbf{c},p)$, implies $\delta^{k+1}(s,t) \geq \delta^{k}(s,t)$. Therefore, assume that the path fragment in $\mathbf{c}$ in the $(k+1)^{th}$ iteration is from r to l. There can be 9 possible cases. The first four cases namely

1. $\delta^{k}(s,\mathbf{c},p) = \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) = \delta^{(k+1)}(t,\mathbf{c},l)$, or
2. $\delta^{k}(s,\mathbf{c},p) = \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) < \delta^{(k+1)}(t,\mathbf{c},l))$, or
3. $\delta^{k}(s,\mathbf{c},p) < \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) < \delta^{(k+1)}(t,\mathbf{c},l))$, or
4. $\delta^{k}(s,\mathbf{c},p) < \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) = \delta^{(k+1)}(t,\mathbf{c},l))$,

are routine. In these cases $\delta^{k}(s,t) \leq \delta^{k+1}(s,t)$. In cases

1. $\delta^{k}(s,\mathbf{c},p) > \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) > \delta^{(k+1)}(t,\mathbf{c},l))$, or
2. $\delta^{k}(s,\mathbf{c},p) < \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) > \delta^{(k+1)}(t,\mathbf{c},l))$, or
3. $\delta^{k}(s,\mathbf{c},p) = \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) > \delta^{(k+1)}(t,\mathbf{c},l)$,

we need to ask why the path fragment from p to l was not chosen in the k^{th} iteration as that would have resulted in a shorter augmenting path from s to t. This implies that the conjugate edge $n \to l$ was saturated earlier. If flow sent through p relaxes any such constraints, then pushing flow in the path fragment from p to l would have been possible in the k^{th} iteration also. But that would contradict the assertion that flow in the k^{th} iteration was pushed on the shortest path. These cases, therefore, can not arise. In cases

1. $\delta^{k}(s,\mathbf{c},p) > \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) = \delta^{(k+1)}(t,\mathbf{c},l))$, or
2. $\delta^{k}(s,\mathbf{c},p) > \delta^{k+1}(s,\mathbf{c},r)$ and $\delta^{k}(t,\mathbf{c},q) < \delta^{(k+1)}(t,\mathbf{c},l)$

when k^{th} augmenting path was used $(k+1)^{th}$ must have been unavailable. Also, the conjugate edge incident at p must have been covered by the constraint making the r to l path fragment saturated because that path fragment got relaxed after the pushing of flow through the conjugate edge incident at p. Also, path through r and q must have also been unavailable otherwise that would have been shorter and chosen. Conjugate edge incident at p must have been covered by this constraint also otherwise it could not send flow to q. From submodularity constraint (Lemma 2) if p and q and p and l are tight then p, q, l must also be tight in the union flow constraint. Therefore any flow sent from p to q can not relax flow constraint of l. The case therefore can not happen.

□