# Supplementary Material

## A. Reparametrization and Submodularity

A label on $l_p$ on a pixel $p$ induces two types of cost in the total energy formulation. One arising from the data energy $D_p(l_p)$ and the other arising from $W_{\mathbf{c}}(\mathbf{l_c})$ corresponding to all $\mathbf{c}$ of which $p$ is part of. Since every $p$ is assigned exactly one label, we can, without loss of generality, count the data energy $D_p(l_p)$ from within the $W_{\mathbf{c}}(\mathbf{l_c})$ by increasing the value corresponding to all $\mathbf{l_c}$ in which $p$ is labeled $l_p$ and setting $D_p(l_p)$ to zero. We call this step repmetrization/normalization using $l_p$. We show below that such reparametrization preserves submodularity, i.e., a function which is submodular remains so even after reparametrization. As described in Section 2, the submodularity condition requires that:

$$W(X) + W(Y) \geq W(X \vee Y) + W(X \wedge Y).$$

Consider a case when when vectors $X$ and $Y$ contains pixel $p$. By definition, if label of a pixel $p$ in $X$ and $Y$ is $i$ and $j$ respectively, then the label of $p$ in $X \vee Y$ and $X \wedge Y$ is $\max(i, j)$ and $\min(i, j)$ respectively. Assuming reparametrization by $\delta$ using $l_p$, three cases arise:

1. Label of $p$ in both $X$ and $Y$ is $l_p$. In this case label of $p$ in both $X \vee Y$ and $X \wedge Y$ is $l_p$. Both r.h.s and l.h.s of the equation increase by $2\delta$ and the equation remains satisfied.

2. Label of $p$ in both $X$ and $Y$ is not $l_p$. In this case label of $p$ in both $X \vee Y$ and $X \wedge Y$ can not be $l_p$. There is no change in r.h.s or l.h.s. and the equation remains satisfied.

3. Label of $p$ in one of $X$ and $Y$ is $l_p$. Let the label of $p$ in other is $l'_p$. If $l_p \geq l'_p$ then label of $p$ in $X \vee Y$ is $l_p$ else label of $p$ in $X \wedge Y$ is $l_p$. Both r.h.s. and l.h.s. of the equation increase by $\delta$ and the equation remains satisfied. The case when $l_p < l'_p$ can be equivalently proved.

## B. Submodularity and Infeasible States

The encoding as defined in Section 3 encodes a label at position $i$ in $\mathcal{L}$ such that it is represented by the state of $\mathbf{b_p}$ in which $i$ Boolean variables from left have value 1 and remaining $(m - i)$ variables have value 0 ($m = |\mathcal{L}|$).

It may be noted that every feasible Boolean encoding has 1 transition of type $1 \rightarrow 0$ at maximum and no transition of type $0 \rightarrow 1$. The Boolean submodularity conditions state that:

$$W(X) + W(Y) \geq W(X \cup Y) + W(X \cap Y)$$

Two cases arise:

1. Both $X$ and $Y$ correspond to feasible states. We already showed in Section 3 that the submodularity conditions are satisfied.

2. One of $X$ or $Y$ or both $X$ and $Y$ correspond to infeasible states. Follows from the observation that neither of the two operators $\cup$ or $\cap$ can create a transition of type $0 \rightarrow 1$ transition. Total number of transitions of type $0 \rightarrow 1$ remain the same or reduce in the the states created by the use of $\cup$ and $\cap$ operators.

## C. Example

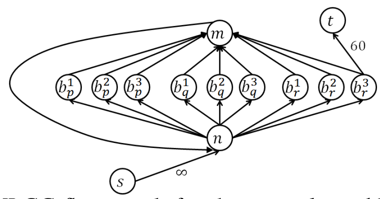


Figure 4: MLGC flow graph for the example problem given in Table 1 and 2

Consider the example problem as given in Tables 1 and 2. As explained the reparametrization followed by encoding and reparametrization again, results in 2-label problem as given in Table 4. The cost of uniform labeling 000000000 is set as 0 and the unary cost of $D_{b_r^3}(0)$ is set to 60. Note that rest of the unary costs are 0. The equivalent flow graph corresponding to the problem is given in Figure 4. Corresponding to the unary cost $D_{b_r^3}(0)$ there is an edge $b_r^3 \rightarrow t$ with capacity 60. The edge $s \rightarrow n$ with cost $\infty$ ensures that node $n$ will never be in the set $T$. This ensures that all the pixel nodes of a gadget can never be in $T$. This is necessary because that corresponds to a labeling of 111111111 of the pixel nodes of the gadget which is ruled out because uniform cost of labeling 111111111 has been set to $\infty$.

The initial residual capacities of the conjugate edges $n \rightarrow b_p^1, n \rightarrow b_p^2, n \rightarrow b_p^3, n \rightarrow b_q^1, n \rightarrow b_q^2, n \rightarrow b_q^3, n \rightarrow b_r^1, n \rightarrow b_r^2$ and $n \rightarrow b_r^1$ can be shown to be $\infty, 50, 30, \infty, 50, 40, \infty, 50$ and 50 respectively (minimum of clique potentials where nodes are labeled 1).

There is one flow augmentation possible along the path $s \rightarrow n \rightarrow b_r^3 \rightarrow t$. Flow of 50 is sent along this path after which constraint corresponding to labeling 011011011 becomes tight. No further flow is possible. The value of dual as determined by the total flow sent is 50.

The $(S, T)$ cut created has $b_p^1, b_q^1$ and $b_r^1$ in $S$ set and remaining nodes in $T$ set. The $b$ nodes in $S$ set are labeled 0 and remaining as 1. Labeling on $b$ nodes is decoded to find the solution to the original multi-label problem, which in the example problem comes out to be $aaa$. The value of the primal for the labeling is 50 which is equal to the dual, certifying the optimality of the solution.