CHAPTER 3

OPTIMIZATION USING GRAPH CUTS

In this chapter, we are going to go through few important mathematical models involving graph cuts applied to computer vision problems. As discussed in chapter 1, the structural constraint plays a pivotal role in the form of objective function. This chapter outlines few of the important structures and corresponding objective functions dealt with graph cuts. The structures involve uniformly smooth structure, segment wise constant structure and segment wise smooth structures. The work presented in this chapter is inspired by / based on the content of **[83], [84]** and **[86].**

3.1 GRAPH CUT MODEL FOR UNIFORMLY SMOOTH STRUCTURE

In this model, the structure constraint is encoded by the sub – function $\psi_{v,w}(x_v, x_w)$ as linear relationship of neighbouring pixels with corresponding weights. It is defined by $\psi_{v,w}(x_v, x_w) = c(v,w)\left|x_v - x_w\right|$. Where *c(v,w)* is the constant corresponding to pair of pixels *v* and *w*. With the expression of structural constraint defined as above, the objective function to be minimized takes the form,

$$O(X) = \sum_{v \in V} \varphi_v(x_v) + \sum_{\{v,w\} \in N} c(v,w)\left|x_v - x_w\right| \tag{3.1}$$

The graph cuts can efficiently minimize the objective function (3.1) globally.

3.1.1 CONSTRUCTION OF NETWORK

The graph cuts technique separates a collection of pixels into two subgroups and hence determines one of the most cost efficient binarization on the given group of pixels in the light of objective function. Let $\{v_1, v_2, ..., v_n\}$ be set of pixels and $\{\sigma_1, \sigma_2, ..., \sigma_p\}$ be set of all possible labels. For the problem under consideration, we construct a graph G with vertex set *V* containing *(np – n + 2)* vertices and edge set *E*. To be more specific, $V = \{v_{11}, v_{12}, ..., v_{1(p-1)}, v_{21}, v_{22}, ..., v_{2(p-1)}, ..., v_{n1}, v_{n2}, ..., v_{n(p-1)}, s, t\}$ with two terminal vertices *s* and *t* called source and sink. Corresponding to each pixel v_i $(1 \le i \le n)$, there are *(p – 1)* vertices v_{i1}, v_{i2}, . . . , $v_{i(p-1)}$ in the graph. The edge set $E = \{e_{ij}^t\ (1 \le i \le n, 1 \le j \le p), e_{ijk}^n (1 \le i \le n, 1 \le j \le n, 1 \le k \le p-1)\}$ consists of two types of edges: e_{ij}^t called terminal edges and e_{ijk}^n called non-terminal edges. e_{ij}^t $(2 \le j \le p-1)$ is an edge connecting $v_{i(j-1)}$ and v_{ij} whereas e_{ijk}^n is an edge connecting v_{ik} and v_{jk} . e_{i1}^t is an edge connecting *s* and v_{i1} whereas e_{ip}^t is an edge connecting $v_{i(p-1)}$ and *t*. Every non-terminal edge e_{ijk}^n has a weight $c(v_i, v_j)$ and terminal edge e_{ij}^t carries a weight $A_i + \varphi_i(\sigma_j)$, where A_i is a constant with

$A_i > (p-1)\left(\sum_{v_j \in N_{v_i}} c(v_i, v_j)\right)$. Figure 3.1 shows a sub-network of *G* representing the structure of the network for two neighbouring pixels v_i and v_j.

TERMINOLOGY

A cut (considered as collection of edges) on the network *(G,V,E)* is said to be an *F-cut* provided it contains exactly one terminal edge for every pixel v_i.

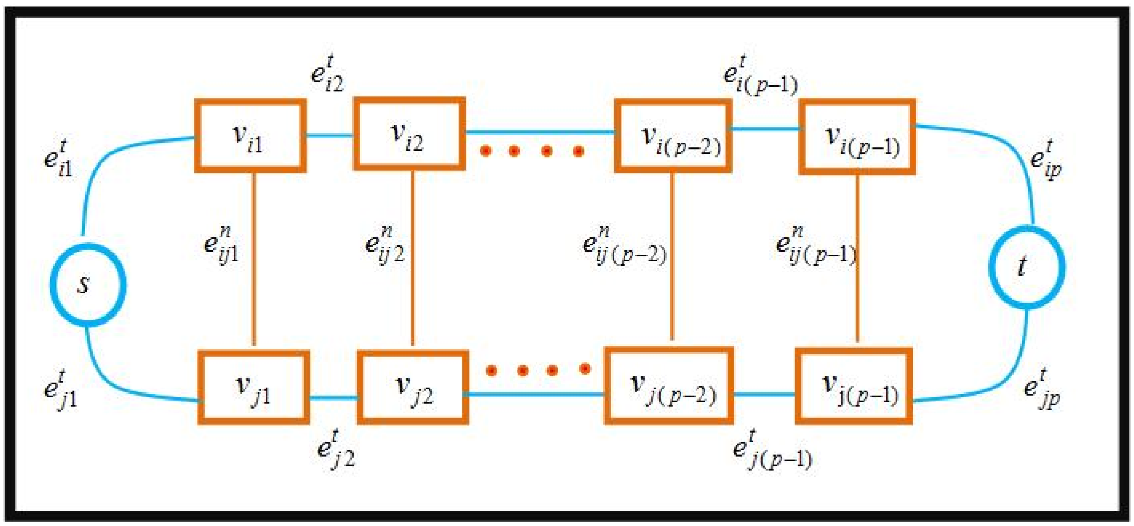

Figure 3.1 Sub-network of G for neighboring pixels v_i and v_j of pixel the pixel set of image

Every F- cut C on the network *G* naturally leads to a labeling X' defined as follows:

$$X'(v_i) = \sigma_k, \text{if } e^t_{ik} \in C \qquad (3.2)$$

LEMMA 3.1.1

There is a one to one correspondence between set of all F – cuts on G and set of all possible labeling on V.

Proof: From (3.2), one part of the result is evident as an arbitrary F – cut C of *G* naturally defines a labeling on *V*. It remains to prove that, given an arbitrary labeling on *V*, there exists an F –cut on *G* corresponding to it. Let's assume that, we are given a labeling $X : V \to \Omega$. This naturally defines a cut defined as follows:

$$e^t_{ik} \in C, \text{if } X(v_i) = \sigma_k$$

It is clear from the definition of *C* that, if $X(v_i) = \sigma_k$, $e^t_{ik} \in C$. As *X* is a labelling and it does not assign more than one label to any pixel, there does not exist any pixel, corresponding to which more than one terminal edge is severed in the cut *C*. This proves that, *C* is an F – cut. This proves the result.

THEOREM 3.1.2

If C is a minimum cut on G, it is an F – cut.

Proof: Let C be a minimum cut on G. To prove that, it is an F – cut, we need to prove that, for every pixel, C contains exactly one terminal edge. If possible, let us assume that, there is a pixel v_i, corresponding to which there are two terminal edges e_{ir}^t and e_{iq}^t $(1 < r, q < p)$ in the cut *C*. Let us consider a cut $C' = \left\{ C \cup \left\{ \bigcup_{\substack{v_k \in N(v_i) \\ 1<q<p}} e_{ikq}^n \right\} \right\} \setminus \left\{ e_{iq}^t \right\}$. As $(p-1)\left(\sum_{v_j \in N_{v_i}} c(v_i, v_j) \right) < A_i$, it is clear that,

$$\text{Cost of } C' = \text{Cost of C} + \left(\sum_{\substack{v_k \in N(v_i) \\ 1<q<p}} \left| e_{ikq}^n \right| \right) - \left| e_{iq}^t \right|$$

$$= \text{Cost of C} + (p-1) \sum_{v_j \in N_{v_i}} c(v_i, v_j) - \left(A_i + \varphi_i(\sigma_j) \right)$$

$$< \text{Cost of C} - \varphi_i(\sigma_j) \qquad \left(\because A_i > (p-1)\left(\sum_{v_j \in N_{v_i}} c(v_i, v_j) \right) \right)$$

It immediately follows that the difference of cost of cuts *C* and C' is $\varphi_i(\sigma_j)$, hence is positive. This deduces that, the cost of C' is less than the cost of *C*, which contradicts with the hypothesis that *C* is a minimum cut. This also proves that, for every pixel, *C* must cut at most one terminal edge. From the construction of the graph G, it is clear that, *C* must contain at least one terminal edge in order to separate terminals *s* and *t*. (If it does not contain any terminal edge corresponding to v_i, there exists a path s-$s - e_{i1}^t - e_{i2}^t - .... - e_{i(p-1)}^t - t$ connecting *s* and *t*, which prevents C from being a cut) This proves that, if *C* is a minimum cut, it must contain exactly one terminal edge corresponding to every pixel. This proves that, *C* is an F-cut.

DEFINITION

Every minimum cut C corresponds to a labeling X_C of G defined by, $X_C(v_i) = q, \text{if } e_{iq}^t \in C$.

THEOREM 3.1.3

If C is a minimum cut on G, the sum of weights of all non-terminal edges of G contained in C is

$$\sum_{(v_i, v_j) \in N} c(v_i, v_j) \left| X_C(v_i) - X_C(v_j) \right|.$$

Proof: We first prove that, if $G \setminus C$ leaves v_i and v_j connected to the same terminal, no non-terminal edge joining v_i and v_j will be part of *C*.

Let us assume that, *C* contains e^t_{iq} and e^t_{jq} . i.e. $X_C(v_i) = q$ and $X_C(v_j) = q$. We want to prove that, $e^n_{ijk} \notin C$ for $1 < k < p$. If possible, assume that, $e^n_{ijl} \in C$ for some *l* with $1 < l < p$. It can be easily checked that, C \ $\left\{e^n_{ijl}\right\}$ is also a cut. But, it is contradiction as no proper subset of a cut can be a cut. Thus, $e^n_{ijk} \notin C$ for $1 < k < p$.

Claim 1 : For v_i and v_j $(i < j)$ with $X_C(v_i) = r$ and $X_C(v_j) = q\,(r < q)$, all non-terminal edges $e^n_{ijl} \in C$ for $1 \le r < l < q < p$.

If possible, assume that, $e^n_{ijl} \notin C$ for some fixed *l* with $r < l < q$. If we can show that, there exists a path from *s* to *t* via e^n_{ijl}, we are through. For that, consider the path $s - e^t_{i1} - e^t_{i2} - .... - e^t_{i(r-1)} - e^n_{ij(r-1)} - e^t_{ij(r-1)} - e^t_{jr} - e^t_{j(r+1)} - ... - e^t_{jl} - e^n_{ijl} - e^t_{il} - e^t_{i(l+1)} - ... - t$, which is in G \ C (Refer to Figure 3.2). This is a contradiction with the fact that, *C* is a cut. This proves that, $e^n_{ijl} \notin C$. As *l* was arbitrary no. with $r < l < q$, it proves the claim.

Thus, a minimum cut C contains $\left|X_C(v_i) - X_C(v_j)\right|$ non-terminal edges for every pair of neighbouring pixels v_i and v_j . The weight of each of the non-terminal edges e^n_{ijl} contributing in the total has individual weight of $c(v_i, v_j)$. Note that, the number $\left|X_C(v_i) - X_C(v_j)\right|$ is zero if the pair of pixels are connected to the same terminal, as in that case, $r = X_C(v_i) = X_C(v_j) = q$. Thus, the sum of weights of all the non-terminal edges severed in the minimum cut *C* is $\sum_{(v_i, v_j) \in N} c(v_i, v_j)\left|X_C(v_i) - X_C(v_j)\right|$, which proves the theorem.

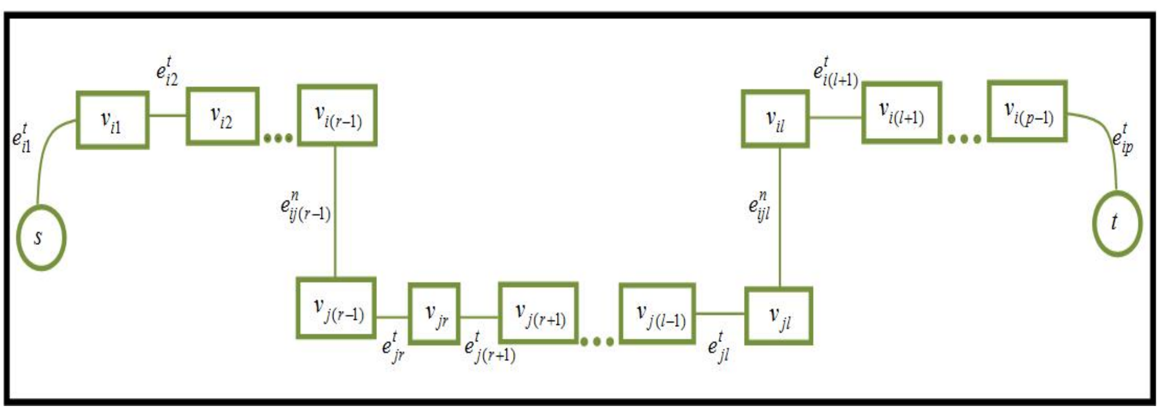

Figure 3.2 Path connecting s and t via e^n_{ijl} in G \C

THEOREM 3.1.4

If C is a minimum cut on G, the labeling X_C *corresponding to C minimizes the objective function O(X) under consideration.*

Proof: Let C be a minimum cut on G. To prove the result, we need to show that, the cost of the cut is $O(X_C)$ plus a constant, where the constant does not depend on the particular cut. From Lemma 3.1.1, C is an F – cut. Hence, corresponding to every pixel of V, C contains unique terminal edge. Thus, the cost of the cut C due to all terminal edges is given by

$$\sum_{v_i \in V}\left(A_i + \varphi_i(\sigma_j)\right) = \sum_{v_i \in V} A_i + \sum_{v_i \in V} \varphi_i(\sigma_j) \tag{3.3}$$

For all the non terminal edges contained in C, by Theorem 3.1.3, the contribution in the cost of C is $\sum_{(v_i,v_j)\in N} c(v_i,v_j)\left|X_C(v_i) - X_C(v_j)\right|$. Thus, the total cost of C is

$$\sum_{v_i \in V} A_i + \sum_{v_i \in V} \varphi_i(\sigma_j) + \sum_{(v_i,v_j)\in N} c(v_i,v_j)\left|X_C(v_i) - X_C(v_j)\right| \tag{3.4}$$

Where, the first term is a constant.

This proves the theorem.

Thus, the objective function defined in (3.1) is efficiently optimized by the model of graph cuts. The structural term best represents the uniformly smooth structure. The model turns out to be unsuitable for other kinds of structures as it does not favor labeling with neighboring pixels with drastically different labels even if it is appropriate. The other major limitation is due to specific type of label sets allowed in the model. The label set must contain integer values. Thus, the model can't be applied to motion problem. However, the model works significantly well and determines the global minimum in case of uniformly smooth structure.

3.2 GRAPH CUT MODELS FOR SEGMENT WISE SMOOTH STRUCTURE

For this model, the function $\psi_{v,w}(x_v, x_w)$ must be a *semi metric*. That is, it needs to obey following properties:

$$\left.\begin{array}{l} 1.\ \psi_{v,w}(\sigma,\sigma) = 0 \\ 2.\ \psi_{v,w}(\sigma_1,\sigma_2) = \psi_{v,w}(\sigma_2,\sigma_1) \\ 3.\ \psi_{v,w}(\sigma_1,\sigma_2) \geq 0 \end{array}\right\} \tag{3.5}$$

If the function $\psi_{v,w}(x_v, x_w)$ satisfies two properties mentioned in (3.6) in addition to those mentioned in (3.5), it is called *metric*.

$$\left.\begin{array}{l} 4.\ \psi_{v,w}(\sigma_1,\sigma_2) > 0, \text{ if } \sigma_1 \neq \sigma_2 \\ 5.\ \psi_{v,w}(\sigma_1,\sigma_3) \leq \psi_{v,w}(\sigma_1,\sigma_2) + \psi_{v,w}(\sigma_2,\sigma_3) \end{array}\right\} \tag{3.6}$$

Hence, the model we are about to describe tries to minimize the objective function

$$O(X) = \sum_{v \in V} \varphi_v(x_v) + \sum_{\{v,w\} \in N} c(v,w)|x_v - x_w|$$

where $\psi_{v,w}(x_v, x_w)$ is either a semi metric or a metric. Optimizing both kinds of objective functions are NP complete problems. Numerous kinds of structures can be dealt with $\psi_{v,w}(x_v, x_w)$ in case it is semi metric, but in this section, we restrict ourselves to segment wise smooth structure. For segment wise smooth structure, $\psi_{v,w}(x_v, x_w)$ must assign larger penalties to labels x_v and x_w with larger value of $|x_v - x_w|$. However, the value of $\psi_{v,w}(x_v, x_w)$ should not grow unboundedly. One needs to formulate the mathematical expression of $\psi_{v,w}(x_v, x_w)$ so that it takes care of this constraint and in addition, also satisfies (3.5). One way to define $\psi_{v,w}(x_v, x_w)$ taking care of these points is to express it linearly as,

$$\psi_{v,w}(x_v, x_w) = \begin{cases} c(v,w)|x_v - x_w|, & \text{if } |x_v - x_w| < a \\ c(v,w)a, & \text{if } |x_v - x_w| \geq a \end{cases} \tag{3.7}$$

This function penalizes the assignment of labels x_v and x_w to pair of neighbouring pixels *v* and *w* with some constant ($c(v,w)$, which is dependent on the pair of pixels *v* and *w*) times the absolute difference between the labels if the difference is within the permissible bound (i.e. *a*) but does not allow it to grow infinitely making such assignment impossible for the model.

The graph of $\psi_{v,w}(x_v, x_w)$ defined as a linear function is shown in figure 3.3.

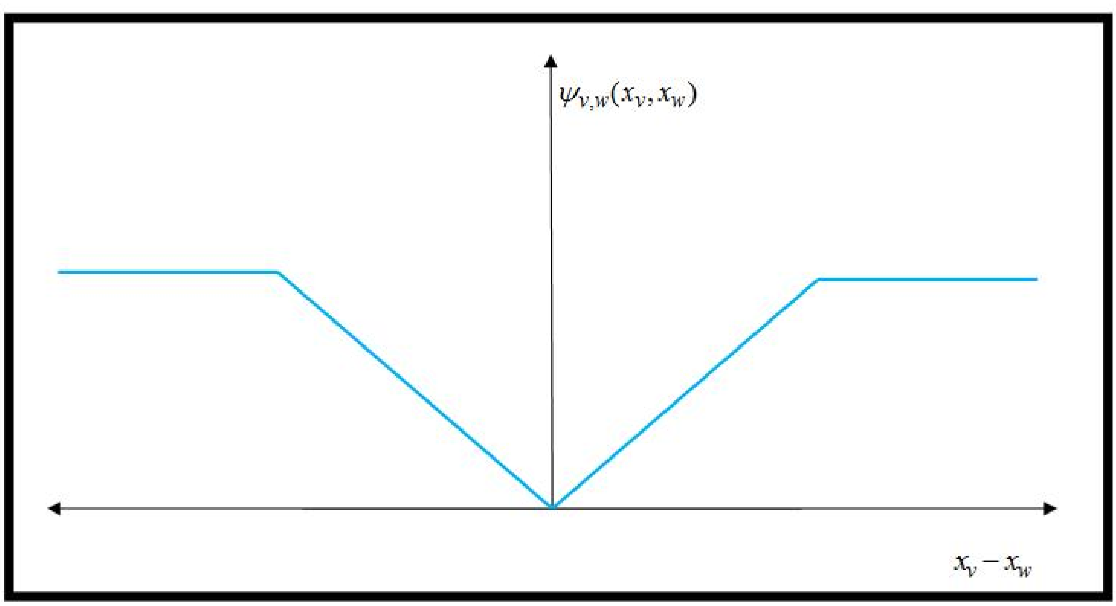

Figure 3.3 Graph of $x_v - x_w$ versus $\psi_{v,w}(x_v, x_w)$ for linearly defined $\psi_{v,w}(x_v, x_w)$

There is an alternate way we can defined it. In (3.7) $\psi_{v,w}(x_v, x_w)$ was a linear function of $|x_v - x_w|$. It is possible to encode more complex type of neighbourhood interrelation by defining $\psi_{v,w}(x_v, x_w)$ as a quadratic function of $|x_v - x_w|$ as defined in (3.8).

$$\psi_{v,w}(x_v, x_w) = \begin{cases} c(v,w)|x_v - x_w|^2, & \text{if } |x_v - x_w| < a \\ c(v,w)a^2, & \text{if } |x_v - x_w| \geq a \end{cases} \tag{3.8}$$

The graph of $\psi_{v,w}(x_v, x_w)$ defined as a quadratic function is shown in figure 3.4.

From the graph it is clear that, this definition of $\psi_{v,w}(x_v, x_w)$ provide a good scope for assignment of any pair of labels to neighbouring pixels, whenever such assignment is essential, by not assigning it infinite cost or penalty. However, it also takes care that such assignments should only be made when it is essential by discriminating such assignments through assigning them relatively higher value compared to those which has less absolute difference in labels. The upper bound of the penalty can be appropriately varied by varying the value of *a*.

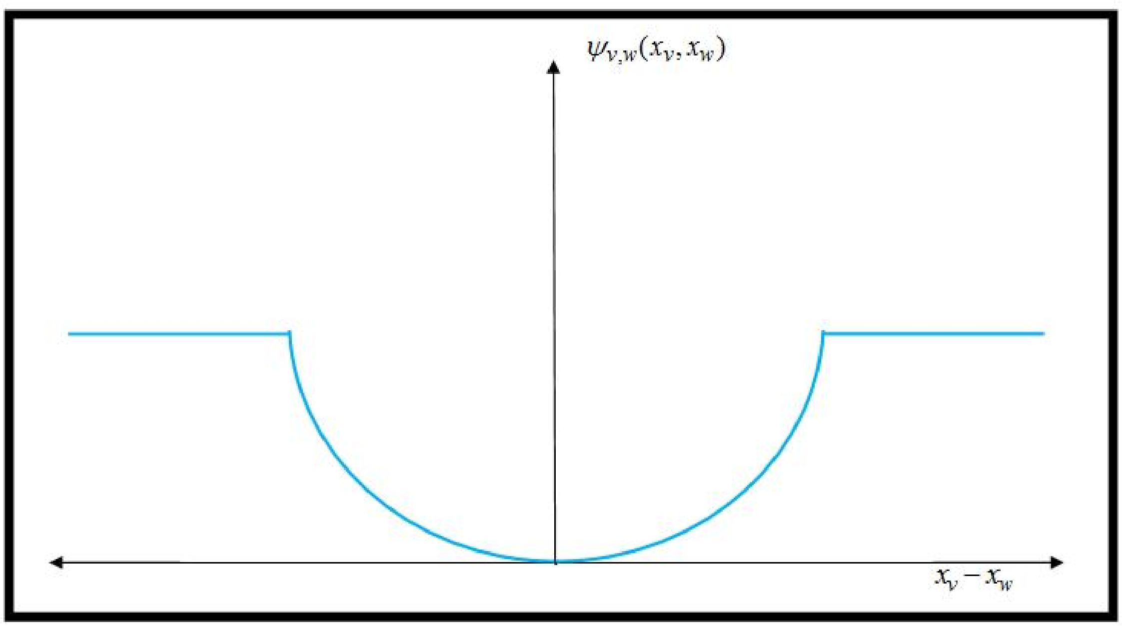

Figure 3.4 Graph of $x_v - x_w$ versus $\psi_{v,w}(x_v, x_w)$ for $\psi_{v,w}(x_v, x_w)$ defined in quadratic expression

It is easy to observe that, the objective function with $\psi_{v,w}(x_v, x_w)$ which is a semi-metric can encrypt vast range of structures which can't be addressed efficiently by the objective function with $\psi_{v,w}(x_v, x_w)$ expressed as metric. Note that, the quadratic expression (3.8) of $\psi_{v,w}(x_v, x_w)$ is a semi – metric but is not a metric as it does not satisfy (3.6). To justify the statement, we can consider ψ defined for the following vertex set and label set.

Vertex set $V = \{u, v, w, x, y\ z\}$

label set $\Omega = \{1, 2, 3, 4, 5\}$

$c(i, j) = 2 \ \forall i, j \in V$ and $a = 20$, Then,

$$\psi_{v,w}(1,4) = 20|1-4|^2 = 20(9) = 180$$

$$\psi_{v,w}(4,5) = 20|4-5|^2 = 20(1) = 20$$

$$\psi_{v,w}(1,5) = 20|1-5|^2 = 20(16) = 320$$

Clearly, $320 = \psi_{v,w}(1,5) > 200 = 180 + 20 = \psi_{v,w}(1,4) + \psi_{v,w}(4,5)$

which is violation of second property of (3.6). Thus, ψ defined in (3.8) is a semi- metric but not a metric.

However, for the same vertex set and label set, ψ defined in (3.7) is a metric because

$$\psi_{v,w}(1,5) = 80 = 60 + 20 = \psi_{v,w}(1,4) + \psi_{v,w}(4,5)$$

The famous *interchange* (or *swap* move) and *growth* (or *expansion* move) models of graph cuts presented in **[139]** are discussed briefly in this subsection. Both the algorithms use the same analogy in different move spaces. The algorithm laying foundation of the models is as below:

Step 1: Begin with arbitrary labelling X

Step 2 : Set s to be zero

Step 3: For every k in the index set, choose refined labelling X' from all labelling which can be achieved from X with single k-move, such that, O(X') - O(X) is maximum. If such X' exists, make updates X = X' and s = 1

Step 4: If s = 1, go to step 2.

Step 5: Return X.

Figure 3.5 Optimized labelling algorithm for move space S

The algorithm mentioned in figure 3.5 iteratively leads to labelling which is optimal with respect to the move space under consideration. The algorithm begins with an arbitrary labelling *X* and at every stage it searches for the labelling *X'* which can be attained with *k* – moves from *X* in the move space and has a lowest value/ penalty under the objective function among all the labelling which are *k* – move far from *X*. If at any stage, such labelling is found, it immediately replaces *X*. The process continues until no further cost effective labelling exists. The algorithm greedily achieves the locally optimal labelling.

Due to the gluttonous nature of the algorithm, there are few noteworthy benefits.

1. At every stage, the algorithm searches for the most cost effective labelling that is k –move far from the existing labelling and the search for the best labelling traverses the entire move space with rapid speed and these results into fast convergence of the algorithm.
2. In general, to check whether the labelling is local minimum over all possible labelling is computationally expensive task but it can be easily assured that the labelling returned by the

algorithm is really a local minimum. The logical justification behind such assurance is that the algorithm terminates only if the labelling in the last iteration does not find any cost efficient replacement in the move space. This proves that, the labelling returned by the algorithm is local minimum over all the labelling those are k–move far from the final labelling.

The prime reason for the success of the algorithm lies in its traverse through the entire move space and best possible moves made during the traverse. In the process of coding of the algorithm, the task could be accomplished by random order traverse along the entire move space or it could be through some systematic ordered moves covering the entire move space.

3.2.1 GRAPH CUT MODEL FOR INTERCHANGE MOVES

As discussed in Chapter 2, interchange move space for pair of labels (σ_1, σ_2) with reference to initial labelling X consists of all labelling X' those can be achieved by interchanging the labels of selected pixels of two subsets $V_1 = \{v \in V | X_v = \sigma_1\}$ and $V_2 = \{v \in V | X_v = \sigma_2\}$. In this model, in each cycle, the most cost efficient labelling over the interchange move space specified by pair of labels (σ_1, σ_2) and initial labelling X is determined. In the next cycle, the interchange move for new pair of labels are considered with the latest labelling of the last cycle as an initial labelling for the current cycle and the optimization in the move space is carried out. The algorithm terminates when there is no pair of labels which gives the cost efficient labelling in the respective interchange move space. The solution arrived at in this way turns out to be the local minimum over the space of all the labelling of the move space under the light of objective function.

At every cycle, minimum labelling over current interchange move space is evaluated. That is a crucial step, which is addressed by graph cuts terminology. During any cycle with pair of labels (σ_1, σ_2) and initial labelling X, the undirected network flow is constructed. The vertex set of the network flow G consists of two terminals σ_1 and σ_2 along with other non-terminal vertices $V' = V_1 \cup V_2 = \{v \in V \mid X_v = \sigma_1 \text{ or } X_v = \sigma_2\}$, where $V_1 = \{v \in V \mid X_v = \sigma_1\}$ and $V_2 = \{v \in V \mid X_v = \sigma_2\}$. For every pixel v of V', there are two terminal edges $e_v^{\sigma_1}$ and $e_v^{\sigma_2}$ connecting it (i.e. to v) to σ_1 and σ_2 respectively. In addition, for every pair of pixels u and v, which are neighbours, there is a non-terminal edge e_{uv}^n connecting them. Thus, the network G has vertex set $\boldsymbol{V} = \{\sigma_1, \sigma_2\} \cup V_1 \cup V_2$ and edge set $\boldsymbol{E} = \{e_v^{\sigma_1} \mid v \in V_1 \cup V_2\} \cup \{e_v^{\sigma_2} \mid v \in V_1 \cup V_2\} \cup \{e_{uv}^n \mid u \in N(v)\}$ as shown in Fig 3.6. The assignment of weights to the edges is as per follows:

For terminal edges $e_v^{\sigma_1}$ and $e_v^{\sigma_2}$,

$$\left.\begin{aligned} \left|e_v^{\sigma_1}\right| &= \varphi_v(\sigma_1) + \sum_{\substack{\{v,\, w\} \in N \\ w \notin V_1 \cup V_2}} \psi_{v,w}(\sigma_1, x_w) \\ \left|e_v^{\sigma_2}\right| &= \varphi_v(\sigma_2) + \sum_{\substack{\{v,\, w\} \in N \\ w \notin V_1 \cup V_2}} \psi_{v,w}(\sigma_2, x_w) \end{aligned}\right\} \tag{3.9}$$

For the non-terminal edge e_{uv}^{n},

$$\left|e_{uv}^{n}\right| = \psi_{u,v}(\sigma_1, \sigma_2) \tag{3.10}$$

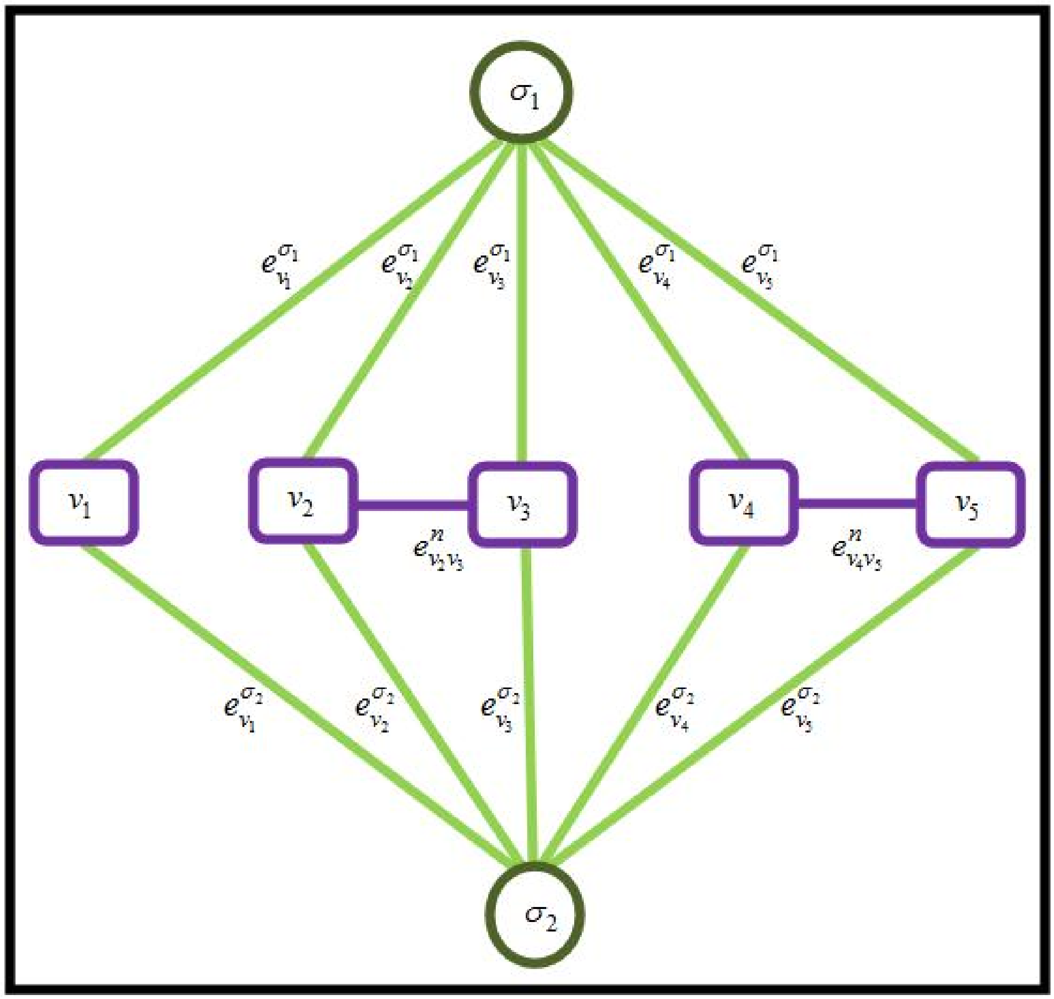

Figure 3.6: Network for interchange move with labels σ_1 and σ_2, and neighbouring pixels $\{v_2, v_3\}$ and $\{v_4, v_5\}$

Thus, *G(**V**,**E**,w)* is a network. The aim is to maximize the value of the flow in the network. By Ford Fulkerson theorem, the value of the maximum flow is the same as the cost of the minimum cut. Hence, the problem reduces to determination of minimum cut on the network *G(**V**,**E**,w)*. Alike earlier model, the new labelling is obtained naturally by the minimum cut *C* on *G*. For every vertex *v* of $V_1 \cup V_2$, the terminal edge contained in the minimum cut *C* decides the label of *v* under the new labelling. Before relating the minimum cut with new labelling, let's go through a result.

LEMMA 3.2.1.1

If C is a cut on the network flow G, it must contain exactly one terminal edge for each vertex v of G.

Proof: To prove the result, we need to prove two parts:

1. There does not exist any cut containing no terminal edge corresponding to some vertex v of $\boldsymbol{V}$.

2. There does not exist any cut containing both the terminal edges corresponding to some vertex v of $\boldsymbol{V}$.

First let us prove 1. If possible, assume that, there's a cut C of G, which contains no terminal edge corresponding to some vertex v of $\boldsymbol{V}$. Then, we can prove, by showing the existence of a path connecting both the terminals in $G \setminus C$ that C is not a cut. Consider the path $\sigma_1 - (e_v^{\sigma_1}) - v - (e_v^{\sigma_2}) - \sigma_2$. As none of the terminal edges $e_v^{\sigma_1}$ and $e_v^{\sigma_2}$ are contained in the cut C, the entire path is in $G \setminus C$. Thus, $G \setminus C$ does not separate the terminals and hence C can't be a cut on G.

To prove the second part, if possible, assume that, there's a cut C of G, which contains both the terminal edges corresponding to some vertex v of V. We will prove that C cannot be a cut by showing that there exists a proper subset C_1 of C which is a cut.

Now, if there is a path P in G \ C connecting v and σ_1, there must not exist any path connecting v and σ_2 as C is a cut. If we add back a terminal edge $e_v^{\sigma_1}$ in $G \setminus C$, it certainly doesn't give rise to any path connecting to both the terminals. Thus, $C_1 = C \setminus \{e_v^{\sigma_1}\}$, which is subset of C, is also be a cut, which is a contradiction as C is a cut.

In case of existence of path P in G \ C connecting v and σ_2, we can consider $C_1 = C \setminus \{e_v^{\sigma_2}\}$ and can yield contradiction.

This proves that, every cut C of the network G must contain exactly one terminal edge corresponding to every vertex v of V.

Lemma 3.2.1.1 naturally gives rise to new labelling X_C corresponding to cut C given by,

$$\left.\begin{array}{l} X_C(v) = \begin{cases} \sigma_1, \text{if } e_v^{\sigma_1} \in C \\ \sigma_2, \text{if } e_v^{\sigma_2} \in C \end{cases} \forall v \in V_1 \cup V_2 \\ X_C(v) = X(v), \forall v \in V \setminus (V_1 \cup V_2) \end{array}\right\} \tag{3.11}$$

In simple words, the cut C changes the labels of only those pixels v of V which have existing labels σ_1 or σ_2 (i.e. which belongs to $V_1 \cup V_2$) by flipping the labels between σ_1 and σ_2 in revised labelling X_C if necessary, but doesn't change the labels of those pixels which have existing labels other than σ_1 and σ_2. Thus, the new labelling returned by cut C is only one interchange move far from the existing labelling as per our requirement. This leads to the next lemma.

LEMMA 3.2.1.2

The labeling X_C corresponding to cut C of the network flow G is one interchange move far from the existing labeling.

Proof: Note that, the only difference between labeling X and X_C is that few pixels with labels σ_1 and σ_2 in labelling X have interchanged their labels in new labelling X_C and no other pixels have changed their labels. This proves that, X_C is single interchange move away from the labeling X.

LEMMA 3.2.1.3

If v_1 and v_2 are two pixels belonging to $(V_1 \cup V_2)$ with $X_C(v_1) \neq X_C(v_2)$, then $e^n_{v_1v_2} \in C$.

Proof: To prove the result, assume that, $X_C(v_1) = \sigma_1 \neq \sigma_2 = X_C(v_2)$ and $e^n_{v_1v_2} \notin C$. $X_C(v_1) = \sigma_1 \neq \sigma_2 = X_C(v_2) \Rightarrow e^{\sigma_1}_{v_1}, e^{\sigma_2}_{v_2} \in C$ but $e^{\sigma_2}_{v_1}, e^{\sigma_1}_{v_2} \notin C$. We will show that, there is a path P from σ_1 to σ_2 via $e^n_{v_1v_2}$ in $G \setminus C$. Consider the path P is σ_1 - ($e^{\sigma_1}_{v_2}$) - v_2 - ($e^n_{v_1v_2}$) - v_1 - ($e^{\sigma_2}_{v_1}$) - σ_2 which entirely lies in $G \setminus C$ (Refer to Fig. 3.7). This implies that, removal of C from G does not separate terminals σ_1 and σ_2, which is a contradiction with the fact that, C is a cut. Thus, $e^n_{v_1v_2} \in C$. Similarly, we can handle the other case, where $X_C(v_1) = \sigma_2 \neq \sigma_1 = X_C(v_2)$ and prove that $e^n_{v_1v_2} \in C$ (Refer to Fig. 3.7). This proves the result.

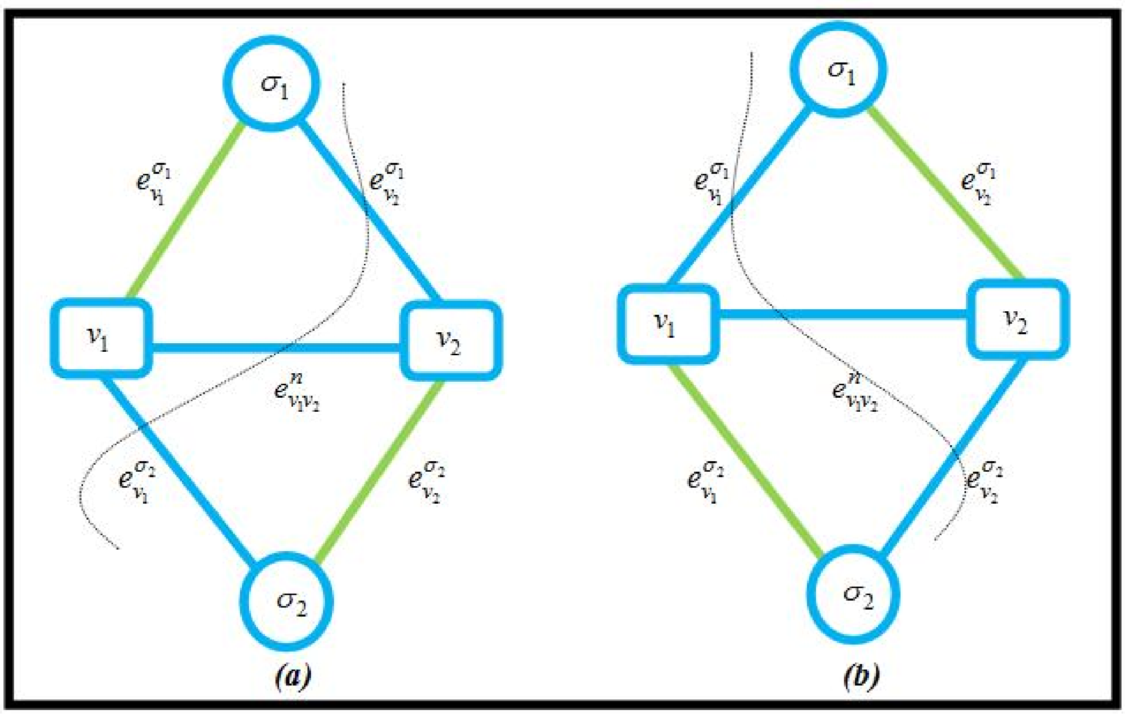

Figure 3.7: Cut C with reference to pixel v when $X_C(v_1) \neq X_C(v_2)$

COROLLARY 3.2.1.4

If v_1 and v_2 are two pixels belonging to $(V_1 \cup V_2)$ with $X_C(v_1) = X_C(v_2)$, then $e^n_{v_1v_2} \notin C$.

Proof: Without loss of generality, let's assume that, $X_C(v_1) = X_C(v_2) = \sigma_1$ (Refer to Fig. 3.8(*a*)). This implies that, $e_{v_1}^{\sigma_1}, e_{v_2}^{\sigma_1} \in C$. Now, if possible, assume that, $e_{v_1v_2}^{n} \in C$. Then, *C* will have a subset C_1=$C \setminus \{e_{v_1v_2}^{n}\}$. We want to show that, C_1 is a cut. For that, if possible, assume that, it is not. Then, there exists a path *P* from σ_1 to σ_2 via $e_{v_1v_2}^{n}$ in $G \setminus C_1$. Let us construct a new path P_1 from *P* by replacing the sub-path of *P* joining v_1 and σ_2 via $e_{v_1v_2}^{n}$ with a new path v_1 - $(e_{v_1}^{\sigma_2})$ - σ_2. Thus, P_1 is a new path joining σ_1 and σ_2, which entirely lies in G \ C (as C_1=$C \setminus \{e_{v_1v_2}^{n}\}$). This contradicts with the fact that, *C* is a cut. This proves that, C_1 is a subset of *C* and is also a cut. This proves that our assumption that, $e_{v_1v_2}^{n} \in C$ is false. Thus, $e_{v_1v_2}^{n} \notin C$.

The other case, $X_C(v_1) = X_C(v_2) = \sigma_2$ can be analogously proved (Refer to Fig. 3.8(*b*)).

This proves the result.

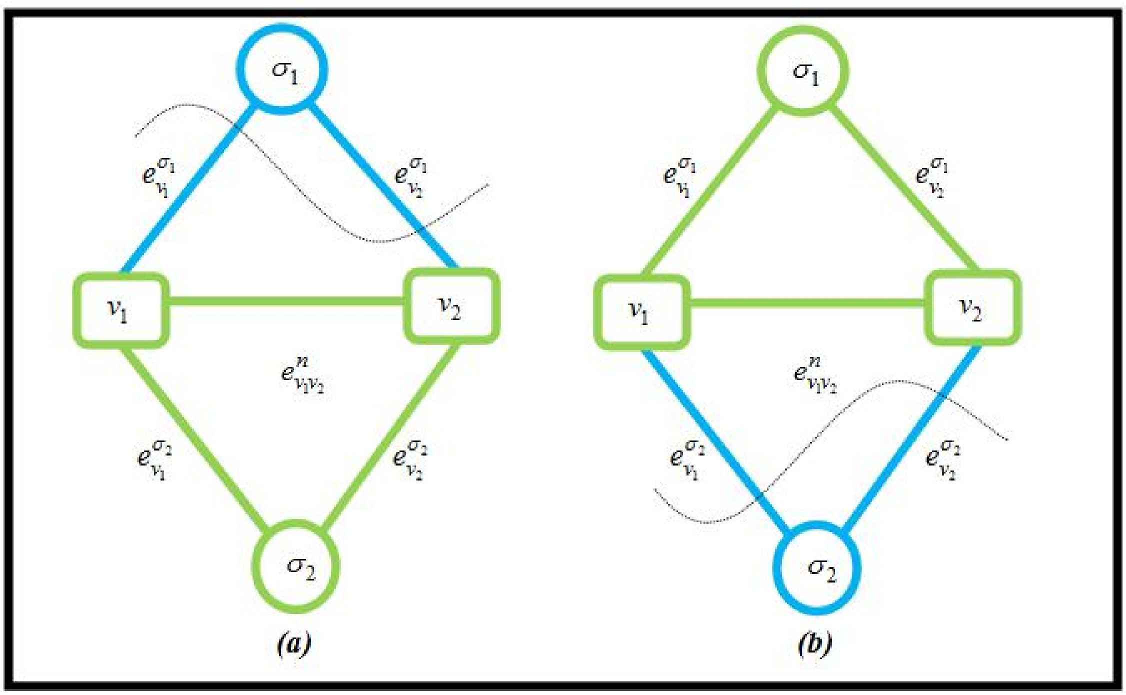

Figure 3.8: Cut C with reference to pixel v when $X_C(v_1) = X_C(v_2)$

LEMMA 3.2.1.5

For any non – terminal link $e_{v_1v_2}^{n}$ *joining* v_1 *and* v_2 *and cut C, the weight of the set of all edges common in* $e_{v_1v_2}^{n}$ *and C is* $\psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$.

Proof: We want to prove that, $\left|e^{n}_{v_1v_2} \cap C\right|$ is $\psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$

There are four possible cases:

If $X_C(v_1) = X_C(v_2) = \sigma_1$ then, $e^{\sigma_1}_{v_1}, e^{\sigma_1}_{v_2} \in C$, and by corollary 3.2.1.4, $e^{n}_{v_1v_2} \notin C$. Thus, the set of edges common to both $e^{n}_{v_1v_2}$ and C is an empty set. Thus, the weight of set of all the edges common to both $e^{n}_{v_1v_2}$ and C is simply weight of an empty set, which is zero. Thus, $\left|e^{n}_{v_1v_2} \cap C\right| = 0$. Also, $\psi_{v_1,v_2}(X_C(v_1), X_C(v_2)) = \psi_{v_1,v_2}(\sigma_1, \sigma_1) = 0$ ($\because$ ψ_{v_1,v_2} is a semi-metric).

Thus, $\left|e^{n}_{v_1v_2} \cap C\right| = 0 = \psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$.

The second case : $X_C(v_1) = X_C(v_2) = \sigma_2$ can be handled analogously. In this case too,

$\left|e^{n}_{v_1v_2} \cap C\right| = 0 = \psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$.

Let's consider the third case: $X_C(v_1) = \sigma_1 \neq \sigma_2 = X_C(v_2)$. Then, $e^{\sigma_1}_{v_1}, e^{\sigma_2}_{v_2} \in C$, and by Lemma 3.2.1.3, $e^{n}_{v_1v_2} \in C$. Thus, the intersection of $e^{n}_{v_1v_2}$ and C is $e^{n}_{v_1v_2}$.

Thus, $\left|e^{n}_{v_1v_2} \cap C\right| = \left|e^{n}_{v_1v_2}\right| = \psi_{v_1,v_2}(\sigma_1, \sigma_2)$ ($\because$ Using (3.10))

$= \psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$ ($\because$ $X_C(v_1) = \sigma_1$, $X_C(v_2) = \sigma_2$)

Thus, $\left|e^{n}_{v_1v_2} \cap C\right| = \psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$.

The forth case $X_C(v_1) = \sigma_2 \neq \sigma_1 = X_C(v_2)$ can be handled similar to the third case. In this case also,

$\left|e^{n}_{v_1v_2} \cap C\right| = \left|e^{n}_{v_1v_2}\right| = \psi_{v_1,v_2}(\sigma_1, \sigma_2)$ ($\because$ Using (3.10))

$= \psi_{v_1,v_2}(\sigma_2, \sigma_1)$ ($\because$ ψ_{v_1,v_2} is a semi-metric)

$= \psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$ ($\because$ $X_C(v_1) = \sigma_2$, $X_C(v_2) = \sigma_1$).

Thus, in all the cases,

$\left|e^{n}_{v_1v_2} \cap C\right| = \psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$. This proves the result.

Note that, the proof of the result uses the fact that $\psi_{u,v}$ is a semi – metric.

THEOREM 3.2.1.6

The set of all cuts C on G and the set of all labeling which are single interchange move (involving labels σ_1 and σ_2) away from the labeling X are in one to one correspondence.

Proof: Lemma 3.2.1.1 implies that, every cut *C* of G leads to a labeling X_C defined by (3.11) which is one interchange move with respect to labels σ_1 and σ_2 away from *X*.

It remains to prove that, every labeling X_C, which is one interchange move with respect to labels σ_1 and σ_2 away from X corresponds to a cut on *G*. Let X_C be a labeling that is single interchange move away from X with respect to labels σ_1 and σ_2. We define cut *C* corresponding to X_C as follows:

$$C = \bigcup_{\substack{v\in V \\ X_C(v)=\sigma_1}} \{e_v^{\sigma_1}\} \cup \bigcup_{\substack{v\in V \\ X_C(v)=\sigma_2}} \{e_v^{\sigma_2}\} \cup \bigcup_{\{u,v\}\in N} \{e_{uv}^n \mid X_C(u) \neq X_C(v)\}\,.$$

It is easy to observe that, *C* is a cut on *G* as it does not leave any path joining the terminals in *G* \ *C*. Thus, every labelling X_C that is single interchange move away from *X* with respect to labels σ_1 and σ_2 leads to a cut *C*.

This proves the theorem.

THEOREM 3.2.1.7

The value of labeling X_C (corresponding to the cut C) under the objective function and the cost of cut C differ by some constant independent of cut C.

Proof: We are to prove that, $O(X_C) - |C|$ is a constant independent of the cut *C*. i.e. for every cut *C* of *G*, the value of the difference is a fixed constant. To prove the result, we will prove that, $|C|$ is $O(X_C)$ plus some constant, where the constant is independent of cut *C*.

Note that, $|C|$ is the sum of weight of all the edges contained in the cut *C*. *C* has two types of edges: terminal edges and non – terminal edges. To be more specific,

$$C = \bigcup_{v\in V_1\cup V_2} \{e_v^{\sigma_1} \text{ or } e_v^{\sigma_2}\} \cup \bigcup_{\substack{\{u,v\}\in N \\ u,v\in V_1\cup V_2}} \{e_{uv}^n \mid X_C(u) \neq X_C(v)\} \qquad (3.12)$$

The first term in (3.12) contains exactly one terminal edge for every vertex *v* of *G* as proved in Lemma 3.2.1.1. The second term in (3.12) contains non – terminal edges for only those pair of neighbouring vertices *u* and *v* of *G*, for which *C* contains terminal edges corresponding to *u* and *v* associated with different terminals.

Thus,

$$\left|C\right| = \sum_{v\in V_1\cup V_2} \left|e_v^{\sigma_1} \text{ or } e_v^{\sigma_2}\right| + \sum_{\substack{\{u,v\}\in N \\ u,v\in V_1\cup V_2 \\ X_C(u)\neq X_C(v)}} \left|e_{uv}^{n}\right| \tag{3.13}$$

Note that, the term $\left|e_v^{\sigma_1} \text{ or } e_v^{\sigma_2}\right|$ refers to the weight of the terminal edge corresponding to *v* contained in the cut *C*.

By (3.9), we have,

$$\left.\begin{aligned} \left|e_v^{\sigma_1}\right| &= \varphi_v(\sigma_1) + \sum_{\substack{\{v,\, w\}\in N \\ w\notin V_1\cup V_2}} \psi_{v,w}(\sigma_1, X(w)) \\ \left|e_v^{\sigma_2}\right| &= \varphi_v(\sigma_2) + \sum_{\substack{\{v,\, w\}\in N \\ w\notin V_1\cup V_2}} \psi_{v,w}(\sigma_2, X(w)) \end{aligned}\right\} = \varphi_v(X_C(v)) + \sum_{\substack{\{v,\, w\}\in N \\ w\notin V_1\cup V_2}} \psi_{v,w}(X_C(v), X(w))$$

Thus,

$$\left|e_v^{\sigma_1} \text{ or } e_v^{\sigma_2}\right| = \varphi_v(X_C(v)) + \sum_{\substack{\{v,\, w\}\in N \\ w\notin V_1\cup V_2}} \psi_{v,w}(X_C(v), X(w))\text{. Thus,}$$

$$\sum_{v\in V} \left|e_v^{\sigma_1} \text{ or } e_v^{\sigma_2}\right| = \sum_{v\in V_1\cup V_2} \varphi_v(X_C(v)) + \sum_{v\in V_1\cup V_2} \sum_{\substack{\{v,\, w\}\in N \\ w\notin V_1\cup V_2}} \psi_{v,w}(X_C(v), X(w)) \tag{3.14}$$

Also, $\left|e_{v_1v_2}^{n} \cap C\right| = \psi_{v_1,v_2}(X_C(v_1), X_C(v_2))$ by Lemma 3.2.1. 5. Therefore, the second term of (3.13) becomes,

$$\sum_{\substack{\{v_1,v_2\}\in N \\ v_1,v_2\in V_1\cup V_2 \\ X_C(v_1)\neq X_C(v_2)}} \left|e_{v_1v_2}^{n}\right| = \sum_{\substack{\{v_1,v_2\}\in N \\ v_1,v_2\in V_1\cup V_2 \\ X_C(v_1)\neq X_C(v_2)}} \psi_{v_1,v_2}(X_C(v_1), X_C(v_2)) \tag{3.15}$$

Now, (3.13) to (3.15) together imply that,

$$\begin{aligned} \left|C\right| &= \sum_{v\in V_1\cup V_2} \varphi_v(X_C(v)) + \sum_{v_1\in V_1\cup V_2} \sum_{\substack{\{v_1,\, v_2\}\in N \\ v_2\notin V_1\cup V_2}} \psi_{v_1,v_2}(X_C(v_1), X_C(v_2)) \\ &\quad + \sum_{\substack{\{v_1,\, v_2\}\in N \\ v_1,\, v_2\in V_1\cup V_2 \\ X_C(v_1)\neq X_C(v_2)}} \psi_{v_1,v_2}(X_C(v_1), X_C(v_2)) \\ &= \sum_{v\in V_1\cup V_2} \varphi_v(X_C(v)) + \sum_{\substack{\{v_1,\, v_2\}\in N \\ v_1 \text{ or } v_2\in V_1\cup V_2}} \psi_{v_1,v_2}(X_C(v_1), X_C(v_2)) \\ &\qquad\qquad \left(\because X_C(w) = X(w) \text{ for all } w\notin V_1\cup V_2\right) \end{aligned}$$

But, $O(\mathrm{X}_C)=\sum_{v\in V}\varphi_v(X_C(v))+\sum_{\substack{\{v_1,\,v_2\}\in N\\ v_1,\,v_2\in V_1\cup V_2}}\psi_{v_1,v_2}(X_C(v_1),X_C(v_2))$

Thus,

$$|C|=\left(\sum_{v\in V}\varphi_v(X_C(v))+\sum_{\substack{\{v_1,\,v_2\}\in N\\ v_1,\,v_2\in V_1\cup V_2}}\psi_{v_1,v_2}(X_C(v_1),X_C(v_2))\right)+\left(-\sum_{v\notin V_1\cup V_2}\varphi_v(X(v))-\sum_{\substack{\{v_1,\,v_2\}\in N\\ v_1,\,v_2\notin V_1\cup V_2}}\psi_{v_1,v_2}(X(v_1),X(v_2))\right)$$

Thus, $|C|=O(\mathrm{X}_C)+k$ (3.16)

where

$$k=\left(-\sum_{v\notin V_1\cup V_2}\varphi_v(X(v))-\sum_{\substack{\{v_1,\,v_2\}\in N\\ v_1,\,v_2\notin V_1\cup V}}\psi_{v_1,v_2}(X(v_1),X(v_2))\right)$$ is a constant independent of the cut C.

This proves the result.

COROLLARY 3.2.1.7

A minimum cut C on graph G leads to a labeling X_C *that optimizes the objective function on the space of all labeling those are one interchange move (involving labels* σ_1 *and* σ_2 *) away from the given labeling.*

Proof: Using the fact that, *C* is a minimum cut, we know that, the value of $|C|$ is minimum over all the labeling those are one interchange move far from the given labeling. Thus, by (3.16), the value of $O(\mathrm{X}_C)+k$ is minimum for minimum cut *C*. As *k* is a constant independent of the cut *C*, it proves that, $O(\mathrm{X}_C)$ is minimum for minimum cut *C*. Thus, the value of the objective function is minimum for the labeling X_C (where, *C* is minimum cut on *G*) on the space of all labeling those are one interchange move away from the given labeling.

3.2.2 GRAPH CUT MODEL FOR GROWTH MOVES

In this sub-section, one of the most popular graph cut model is presented. The model deals with optimization of labeling over growth or expansion move space. The growth move space involving label σ allows the pixels (with labels other than σ) to switch their labels to σ .

Given a labeling *X*, we implement graph cut to find the labeling which is single growth move (with reference to label σ) away from *X* and it is the best labeling among all labeling those are single σ -growth move far from the labeling *X*. The term *best* is used with reference to the value of the labeling under objective function. i.e. We want to find a labeling that is single σ -growth move far from the given labeling and has minimum value under objective function among all candidate labeling (i.e. among all labeling which are one σ -growth move far from the given labeling). Paraphrasing the objective, the goal is to minimize the objective function over the σ - growth space of labeling *X*.

The network has a vertex set V consisting vertex v_i corresponding to every pixel v_i of pixel set, terminals σ and σ' , and in addition, reserve vertices a_{ij} for every pair of vertices (v_i, v_j) with $\mathrm{X}(v_i) \neq \mathrm{X}(v_j)$ as shown in Figure 3.9. For every vertex v_i corresponding to pixel v_i, there are two terminal edges $e_{v_i}^{\sigma}$ and $e_{v_i}^{\sigma'}$. For every pair of neighboring vertices (v_i, v_j) with same label under labeling X (i.e. $X(v_i) = X(v_j)$), there is a non-terminal edge $e_{v_i v_j}^{n}$. For every pair of neighboring vertices (v_i, v_j) with $X(v_i) \neq X(v_j)$ and corresponding reserve vertex a_{ij}, there are pair of non-terminal edges $e_{v_i a_{ij}}^{n}$ and $e_{a_{ij} v_j}^{n}$ and a terminal edge $e_{a_{ij}}^{\sigma'}$. Thus, the network *G(**V**,**E**,w)* corresponding to σ - growth move of given labelling X will have vertex and edge sets ,

$$\boldsymbol{V} = \bigcup_{v_i \in V} \{v_i\} \bigcup_{\substack{v_i, v_j \in V \\ (v_i, v_j) \in N \\ X(v_i) \neq X(v_j)}} \{a_{ij}\} \cup \{\sigma, \sigma'\},$$

$$\boldsymbol{E} = = \left(\bigcup_{v_i \in V} \{e_{v_i}^{\sigma}, e_{v_i}^{\sigma'}\} \right) \cup \left(\bigcup_{\substack{v_i, v_j \in V \\ (v_i, v_j) \in N \\ X(v_i) \neq X(v_j)}} \{e_{v_i a_{ij}}^{n}, e_{a_{ij} v_j}^{n}, e_{a_{ij}}^{\sigma'}\} \right) \cup \left(\bigcup_{\substack{v_i, v_j \in V \\ (v_i, v_j) \in N \\ X(v_i) = X(v_j)}} \{e_{v_i v_j}^{n}\} \right)$$

The weights of edges of $\boldsymbol{E}$ is defined as,

$\left|e_{v_i}^{\sigma}\right| = \varphi_{v_i}(\sigma)$, for all $v_i \in V$

$\left|e_{v_i}^{\sigma'}\right| = \infty$, for all $v_i \in V$ with $X(v_i) = \sigma$

$\left|e_{v_i}^{\sigma'}\right| = \varphi_{v_i}(X(v_i))$, for all $v_i \in V$ with $X(v_i) \neq \sigma$

$$\left.\begin{aligned} \left|e_{v_i a_{ij}}^{n}\right| &= \psi_{v_i, v_j}(X(v_i), \sigma) \\ \left|e_{a_{ij} v_j}^{n}\right| &= \psi_{v_i, v_j}(\sigma, X(v_j)) \\ \left|e_{a_{ij}}^{\sigma'}\right| &= \psi_{v_i, v_j}(X(v_i), X(v_j)) \end{aligned}\right\} \text{for all } (v_i, v_j) \in N \text{ with } X(v_i) \neq X(v_j)$$

$\left|e_{v_i v_j}^{n}\right| = \psi_{v_i, v_j}(\sigma, X(v_j))$ for all pixels $(v_i, v_j) \in N$ with $X(v_i) \neq X(v_j)$

First of all, it should be observed that, every cut C on G leads to a labeling that is one σ - growth move far from the labeling X in natural way. More precisely,

LEMMA 3.2.2.1

Every cut C on G must contain exactly one terminal edge corresponding to each vertex v_i from V.

Lemma 3.2.2.1can be proved with the argument similar to that of Lemma 3.2.1.1.

Thus, every cut C on G gives rise to labeling X_C given by,

$$X_C(v_i) = \begin{cases} \sigma, & \text{if } e_{v_i}^{\sigma} \in C \\ X(v_i), & \text{if } e_{v_i}^{\sigma'} \in C \end{cases} \tag{3.17}$$

This implies the next result.

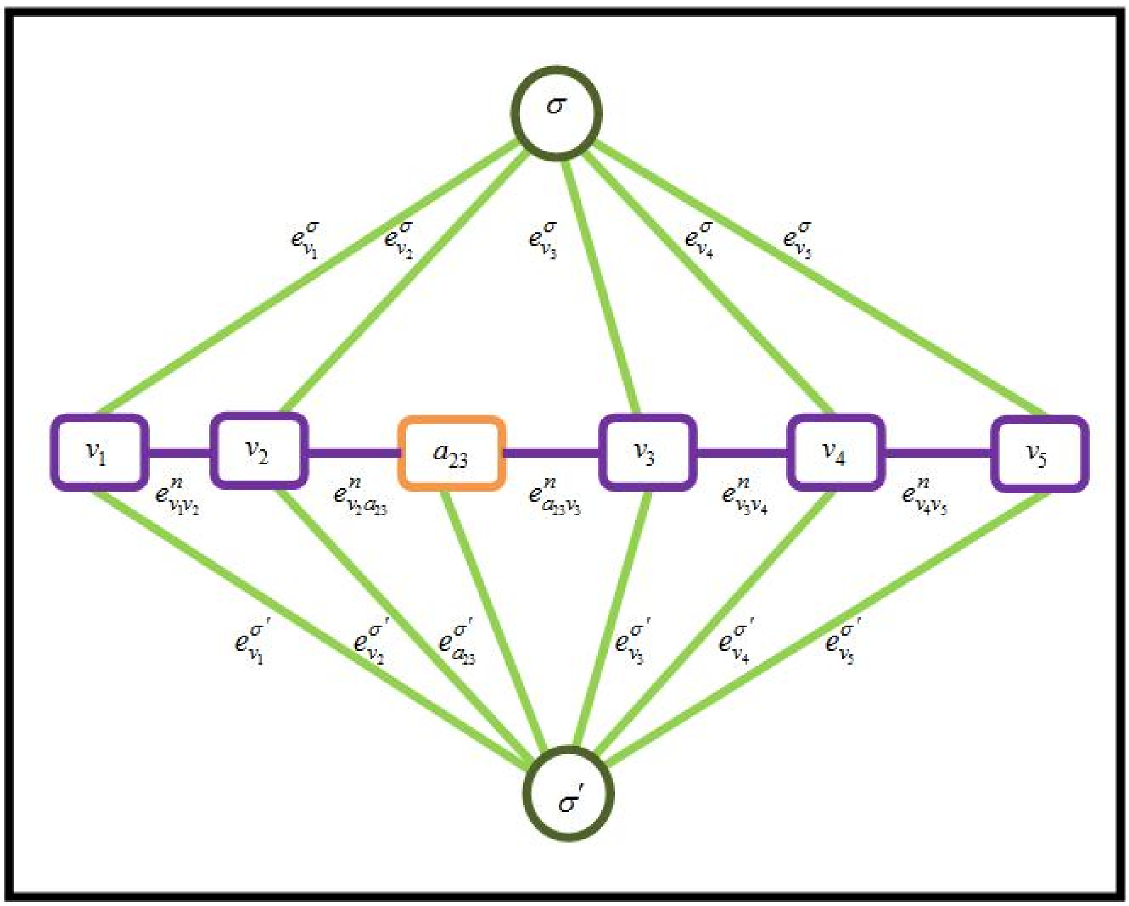

Figure 3.9: Graph for σ - growth move for labeling X for V = $\{v_1,..,v_5\}$ with $X(v_2) \neq X(v_3)$

LEMMA 3.2.2.2

Every cut C on the network G leads to a labeling X_C that is single σ - growth move far from the initial labeling X.

Proof: From (3.17), it is clear that, the only difference between X and X_C is that, few pixels which were assigned labels other than σ in labeling X are assigned with label σ in X_C and no pixel with label σ in labeling X has changed its label in X_C, which proves that, X_C is single σ - growth move far from the initial labeling X.

LEMMA 3.2.2.3

Let C be a cut on the network G. For each pair of neighboring pixels (v_i, v_j) with $X(v_i) = X(v_j)$, if terminal edges corresponding to same terminals are included in C, the corresponding non-terminal edge $e^n_{v_iv_j}$ must not be part of cut C.

Proof: There are two cases: *(i)* $e^{\sigma}_{v_i}, e^{\sigma}_{v_j} \in C$ *(ii)* $e^{\sigma'}_{v_i}, e^{\sigma'}_{v_j} \in C$

Let's first consider the first case. Without loss of generality, let's assume that, $X(v_i) = X(v_j) = \sigma$. Now, if possible, assume that, $e^n_{v_iv_j} \in C$. We will prove that, $C_1 = C \setminus \{e^n_{v_iv_j}\}$ is also a cut. If possible, assume that, there is a path P in $G \setminus C_1$ via $e^n_{v_iv_j}$ connecting both the terminals. As $e^{\sigma}_{v_i}$ and $e^{\sigma}_{v_j}$ does not belong to $G \setminus C_1$, P must have a sub-path P_1 joining σ and v_i which does not involve $e^{\sigma}_{v_i}$ and $e^{\sigma}_{v_j}$ (Refer to Fig 3. 10(a)).

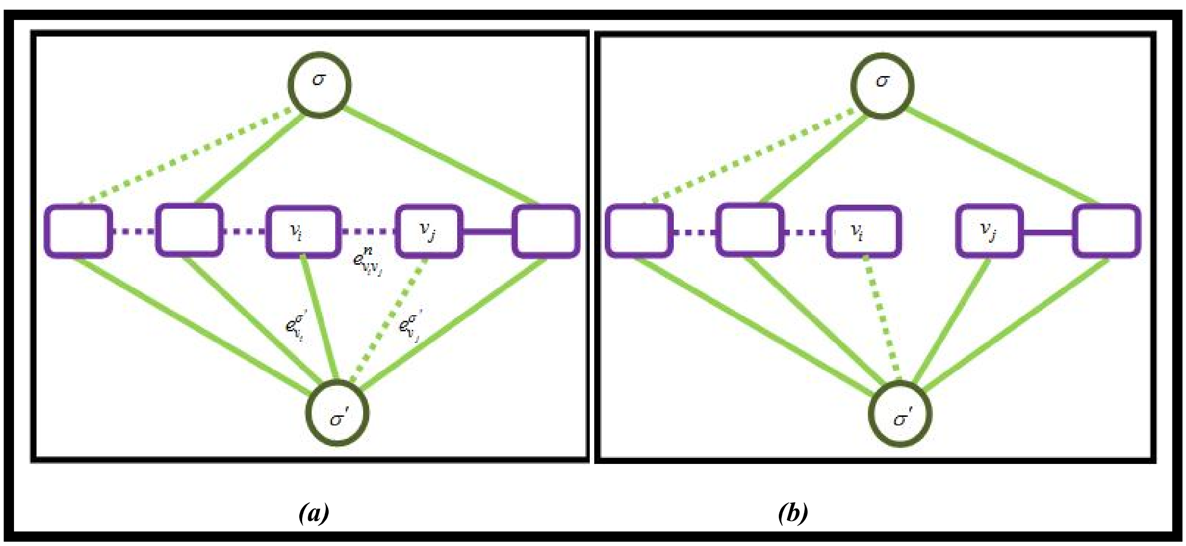

Figure 3.10: (a) path P in G \ C_1 (b) Path $P_1 \cup e^{\sigma}_{v_i}$ in G \ C.

As $e^{\sigma}_{v_i}$ and $e^{\sigma}_{v_j}$ are in C_1, $e^{\sigma'}_{v_i}$ and $e^{\sigma'}_{v_j}$ are not part of C_1 as per Lemma 3.2.2.1. Thus, $e^{\sigma'}_{v_i}$ and $e^{\sigma'}_{v_j}$ are in G \ C_1. Hence, we can create a new path $P_1 \cup e^{\sigma}_{v_i}$ connecting σ and σ' in G \ C_1. Note that, this path does not involve $e^n_{v_i v_j}$ and hence it also lies in $G \setminus C$ (Refer to Fig 3. 10(b)), which contradicts with the fact that, C is a cut. Thus, there does not exist any path in $G \setminus C_1$ joining the terminals. This proves that, C_1 is a cut. As C_1 is a subset of C and no proper subset of a cut is a cut, this proves that, C does not contain $e^n_{v_i v_j}$. The second part can be addressed with similar argument. (Refer figure 3.11)

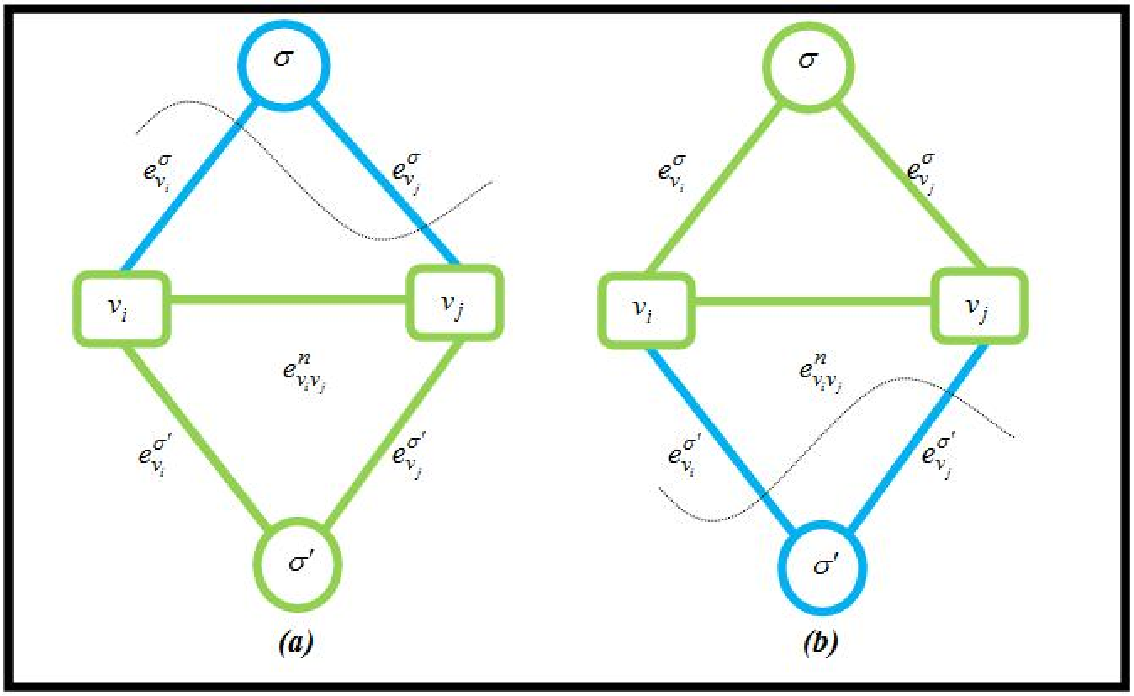

Figure 3.11: (a) Cut C with $e^{\sigma}_{v_i}, e^{\sigma}_{v_j} \in C$ *(b) Cut C with* $e^{\sigma'}_{v_i}, e^{\sigma'}_{v_j} \in C$.

COROLLARY 3.2.2.4

Let C be a cut on the network G. For each pair of neighboring pixels (v_i, v_j) *with* $X(v_i) = X(v_j)$, *if terminal edges corresponding to different terminals are in C, the corresponding non-terminal edge* $e^n_{v_i v_j}$ *must belong to cut C.*

Proof: There are two cases: *(i)* $e^{\sigma'}_{v_i}, e^{\sigma}_{v_j} \in C$ *(ii)* $e^{\sigma}_{v_i}, e^{\sigma'}_{v_j} \in C$

Let's consider Case *(i)*. If possible, assume that, $e^n_{v_i v_j}$ doesn't belong to C. Then, it can be observed that, there is a path connecting both the terminals (σ-$e^{\sigma}_{v_i}$-$e^n_{v_i v_j}$-$e^{\sigma'}_{v_j}$-σ') in $G \setminus C$, which is a contradiction with the fact that, C is cut on G. Thus, every cut containing $e_{v_i}^{\sigma'}$ and $e_{v_j}^{\sigma}$ must contain $e_{v_iv_j}^{n}$. Refer to Figure 3.12 *(a).*

The case *(ii)* can be proved by similar argument. Refer to Figure 3.12 *(b).*

Thus, every cut containing terminal edges corresponding to same terminals for neighboring vertices (with same initial labeling) must not contain the corresponding non-terminal edge (joining the both neighboring vertices), whereas every cut containing terminal edges corresponding to different terminals for neighboring vertices (with same initial labeling) must contain the corresponding non-terminal edge

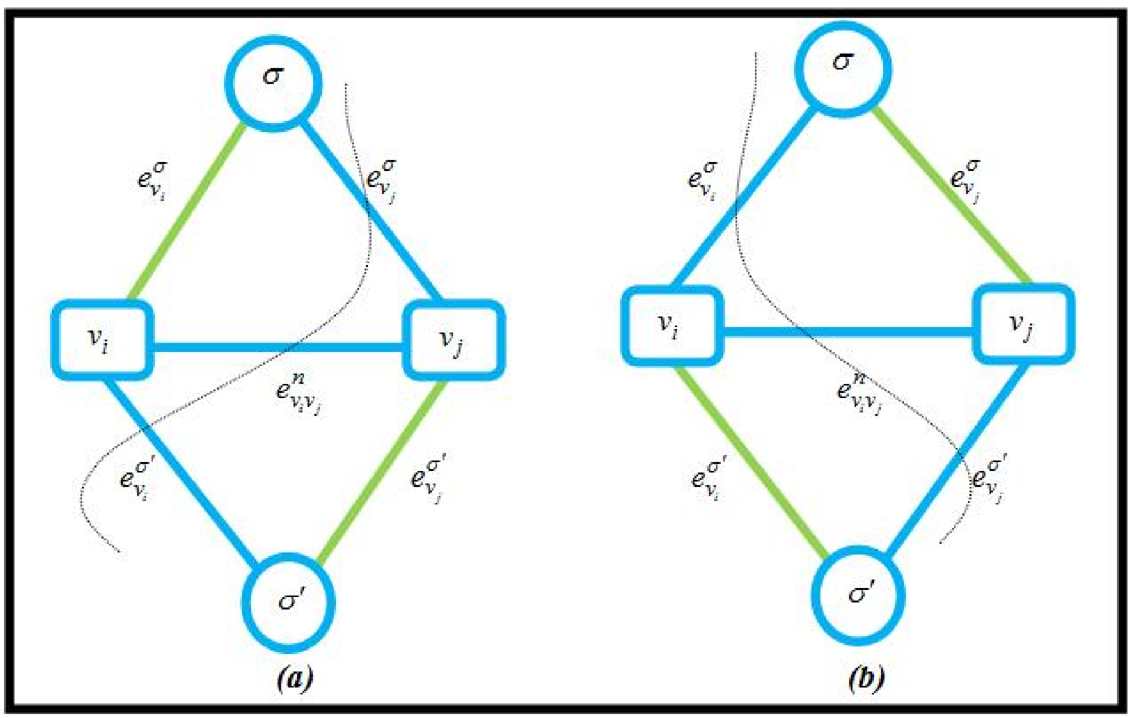

Figure 3.12: (a) Cut C with $e_{v_i}^{\sigma'}, e_{v_j}^{\sigma} \in C$ *(b) Cut C with* $e_{v_i}^{\sigma}, e_{v_j}^{\sigma'} \in C$.

LEMMA 3.2.2.5

Let C be a cut on the network G. For each pair of neighboring pixels (v_i, v_j) *with* $X(v_i) = X(v_j)$*, the total weight of edges common to non-terminal edge* $e_{v_iv_j}^{n}$ *and C is* $\psi_{v_i,v_j}(X_C(v_i), X_C(v_j))$.

Proof: We want to prove that, for $(v_i, v_j) \in N$ *with* $X(v_i) = X(v_j)$, the weight of $e_{v_iv_j}^{n} \cap C$ is $\psi_{v_i,v_j}(X_C(v_i), X_C(v_j))$.

There are four cases:

Case *(i)*: $e_{v_i}^{\sigma}, e_{v_j}^{\sigma} \in C$. Then, by Lemma 3.2.2.3, $e_{v_i v_j}^{n} \notin C$, Thus, there is no edge common to $e_{v_i v_j}^{n}$ and C. Thus,

$$\left|e_{v_i v_j}^{n} \cap C\right| = \left|\{\ \}\right| = 0.$$

Also, $\psi_{v_i, v_j}(\mathrm{X}_C(v_i), \mathrm{X}_C(v_j)) = \psi_{v_i, v_j}(\sigma, \sigma) = 0$

Thus, $\left|e_{v_i v_j}^{n} \cap C\right| = \psi_{v_i, v_j}(\mathrm{X}_C(v_i), \mathrm{X}_C(v_j))$.

Case *(ii)*: $e_{v_i}^{\sigma'}, e_{v_j}^{\sigma'} \in C$ Then, by Lemma 3.2.2.3, $e_{v_i v_j}^{n} \notin C$, Thus, there is no edge common to $e_{v_i v_j}^{n}$ and C. Thus,

$$\left|e_{v_i v_j}^{n} \cap C\right| = \left|\{\ \}\right| = 0.$$

Also, $\psi_{v_i, v_j}(\mathrm{X}_C(v_i), \mathrm{X}_C(v_j)) = \psi_{v_i, v_j}(\sigma', \sigma') = 0$

Thus, $\left|e_{v_i v_j}^{n} \cap C\right| = \psi_{v_i, v_j}(\mathrm{X}_C(v_i), \mathrm{X}_C(v_j))$.

Case *(iii)*: $e_{v_i}^{\sigma'}, e_{v_j}^{\sigma} \in C$. Then, by Corollary 3.2.2.4, $e_{v_i v_j}^{n} \in C$. Thus, the edge $e_{v_i v_j}^{n}$ is common to both $e_{v_i v_j}^{n}$ and C.

$$\left|e_{v_i v_j}^{n} \cap C\right| = \left|e_{v_i v_j}^{n}\right| = \psi_{v_i, v_j}(\sigma, \mathrm{X}(v_j)).$$

Also, $\psi_{v_i, v_j}(X_C(v_i), X_C(v_j)) = \psi_{v_i, v_j}(X(v_i), \sigma)\left(\because X_C(v_i) = X(v_i)\right)$

Thus, $\left|e_{v_i v_j}^{n} \cap C\right| = \psi_{v_i, v_j}(\mathrm{X}_C(v_i), \mathrm{X}_C(v_j))$. $\left(\because X(v_i) = X(v_j) \text{ and } \psi \text{ is metric}\right)$

Case *(iv)*: $e_{v_i}^{\sigma}, e_{v_j}^{\sigma'} \in C$. Then, by Corollary 3.2.2.4, $e_{v_i v_j}^{n} \in C$. Thus, the edge $e_{v_i v_j}^{n}$ is common to both $e_{v_i v_j}^{n}$ and C.

$$\left|e_{v_i v_j}^{n} \cap C\right| = \left|e_{v_i v_j}^{n}\right| = \psi_{v_i, v_j}(\sigma, \mathrm{X}(v_j)).$$

Also, $\psi_{v_i, v_j}(\mathrm{X}_C(v_i), \mathrm{X}_C(v_j)) = \psi_{v_i, v_j}(\sigma, \mathrm{X}(v_j))\left(\because \mathrm{X}_C(v_j) = \mathrm{X}(v_j)\right)$

Thus, $\left|e_{v_i v_j}^{n} \cap C\right| = \psi_{v_i, v_j}(\mathrm{X}_C(v_i), \mathrm{X}_C(v_j))$.

This proves the Lemma.

For pair of neighboring pixels (v_i, v_j) with $X(v_i) \neq X(v_j)$, there is a reserve vertex a_{ij} and a triplet of non-terminal edges $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$. There are numerous ways that a cut can conain these non-terminal edges with a fixed pair of terminal edges corresponding to (v_i, v_j). But, for minimum cut C, if a particular pair of terminal edges corresponding to (v_i, v_j) is known to be included in *C*, one can be assured that, a particular combination of edges from the triplet of non-terminal edges $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$ would be contained in *C*. The Lemma 3.2.2.6 and Corollary 3.2.2.7 precisely characterizes the combination of edges of the triplet:

LEMMA 3.2.2.6

For pair of neighboring pixels (v_i, v_j) *with* $X(v_i) \neq X(v_j)$, *if both the terminal edges corresponding to terminal* σ *are in minimum cut C, no edge from the triplet of edges* $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$ *can be in C. If both the terminal edges corresponding to terminal* σ' *are in minimum cut C, only single edge* $e^{\sigma'}_{a_{ij}}$ *from the triplet of edges* $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$ *will be in C.*

Proof: First, let's consider the case wherein both terminal edges corresponding to terminal σ are in *C,* i.e. $e^{\sigma}_{v_i}, e^{\sigma}_{v_j} \in C$. To prove that, $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C$ is an empty set. Refer to the figure 3.13 (*a*). If possible, assume that, *C* contains at least one edge from $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$, say $e^n_{v_i a_{ij}}$. Then, we will prove that, $C_1 = \mathrm{C} \setminus \{e^n_{v_i a_{ij}}\}$ is also a cut. For that, if possible, assume that, there is a path *P* connecting both the terminals via $e^n_{v_i a_{ij}}$ in $G \setminus C_1$. Then, *P* must contain a sub-path P_1 from σ to v_i, not containing $e^n_{v_i a_{ij}}$. We can create a new path $P_1 \cup e^{\sigma'}_{v_i}$ connecting both the terminals, which entirely lies in $G \setminus C$, (as *C* and C_1 differs only by an edge $e^n_{v_i a_{ij}}$) which is a contradiction with the fact that, *C* is a cut. Thus, C_1 is a subset of cut *C* and it is also a cut. This contradicts with the fact that, *C* is a cut. Thus, *C* can not contain any edge from the triplet of edges $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$.

Now, let's assume that, $e^{\sigma'}_{v_i}, e^{\sigma'}_{v_j} \in C$. Our aim is to prove that, the cut will only contain $e^{\sigma'}_{a_{ij}}$ from the triplet of edges $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$. As shown in Figure 3.13 (*b*), *C* must contain $e^{\sigma'}_{a_{ij}}$ in order to separate both the terminals, if $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}\right\} \cap C = \phi$. For if *C* doesn't contain $e^{\sigma'}_{a_{ij}}$, both the terminals stay connected by a path $\sigma - e^{\sigma}_{v_i} - v_i - e^n_{v_i a_{ij}} - a_{ij} - e^{\sigma'}_{a_{ij}} - \sigma'$ in $G \setminus C$.

In order to prove that, none of the remaining two edges can be part of the cut C, we should note that, C is a minimum cut and by triangular inequality, the weight of $e_{a_{ij}}^{\sigma'}$ is lesser than the sum of weights of other two edges as shown below:

$$\left|e_{a_{ij}}^{\sigma'}\right| = \psi_{v_i,v_j}(\mathrm{X}(v_i),\mathrm{X}(v_j)) \le \psi_{v_i,v_j}(\mathrm{X}(v_i),\sigma) + \psi_{v_i,v_j}(\sigma,\mathrm{X}(v_j)) (\because \psi_{v_i,v_j} \text{ is a metric})$$
$$= \left|e_{v_i a_{ij}}^{n}\right| + \left|e_{a_{ij} v_j}^{n}\right|$$

Thus, if $e_{v_i}^{\sigma'}, e_{v_j}^{\sigma'} \in C$, C only contains edge $e_{a_{ij}}^{\sigma'}$ from the triplet of edges $\left\{e_{v_i a_{ij}}^{n}, e_{a_{ij} v_j}^{n}, e_{a_{ij}}^{\sigma'}\right\}$.

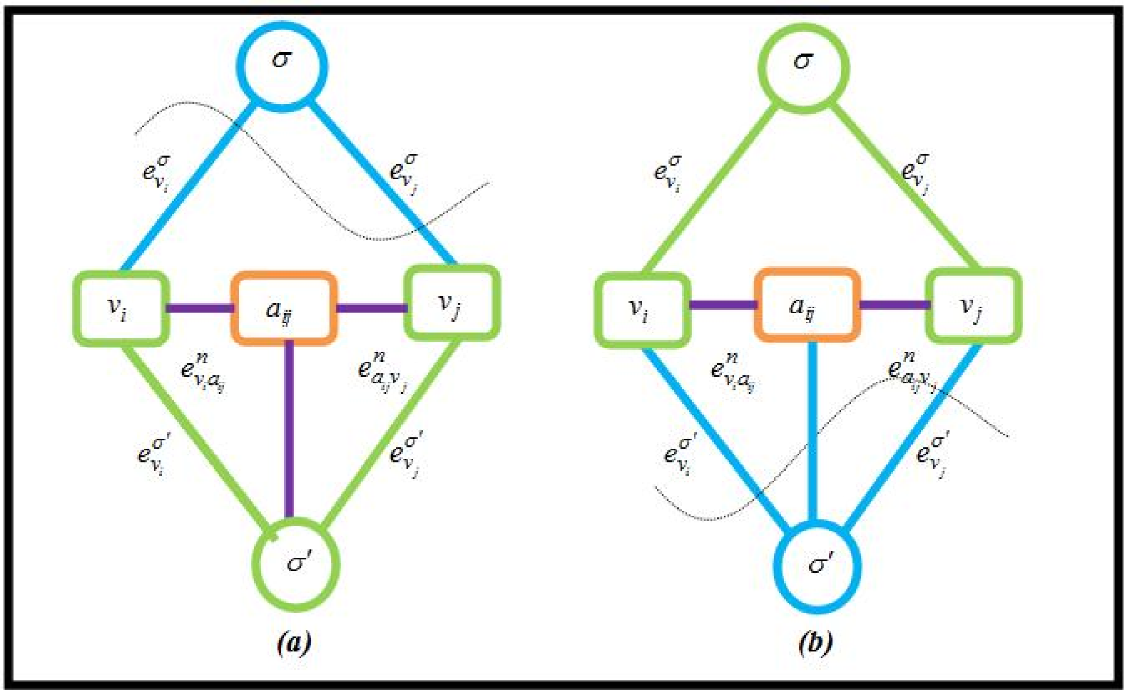

Figure 3.13 (a) $\left\{e_{v_i a_{ij}}^{n}, e_{a_{ij} v_j}^{n}, e_{a_{ij}}^{\sigma'}\right\} \cap C$ *for* $e_{v_i}^{\sigma}, e_{v_j}^{\sigma} \in C$ *(b)* $\left\{e_{v_i a_{ij}}^{n}, e_{a_{ij} v_j}^{n}, e_{a_{ij}}^{\sigma'}\right\} \cap C$ *for* $e_{v_i}^{\sigma'}, e_{v_j}^{\sigma'} \in C$

LEMMA 3.2.2.7

For pair of neighboring pixels (v_i, v_j) *with* $X(v_i) \neq X(v_j)$, *if terminal edges corresponding to terminal* σ *and* σ' *corresponding to vertices* v_i *and* v_j *respectively are in minimum cut C, only edge* $e_{a_{ij} v_j}^{n}$ *from the triplet of edges* $\left\{e_{v_i a_{ij}}^{n}, e_{a_{ij} v_j}^{n}, e_{a_{ij}}^{\sigma'}\right\}$ *will be in C. If terminal edges corresponding to terminal* σ' *and* σ *corresponding to vertices* v_i *and* v_j *respectively are in minimum cut C, only edge* $e_{a_{ij} v_j}^{n}$ *from the triplet of edges* $\left\{e_{v_i a_{ij}}^{n}, e_{a_{ij} v_j}^{n}, e_{a_{ij}}^{\sigma'}\right\}$ *will be in C.*

Proof: Consider the first part: $e^{\sigma}_{v_i}, e^{\sigma'}_{v_j} \in C$, where C is a minimum cut on G. To prove that, $\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \bigcap C = e^{n}_{a_{ij} v_j}$, we observe that, C cannot be a cut if none of the three *edges* are part of C, because in that case, there exists a path $\sigma - e^{\sigma}_{v_j} - v_j - e^{n}_{a_{ij} v_j} - a_{ij} - e^{\sigma'}_{a_{ij}} - \sigma'$ in $G \setminus C$. Thus, C must contain at least one of the three edges. Refer to Figure 3.14 *(b)*. It is clear that,

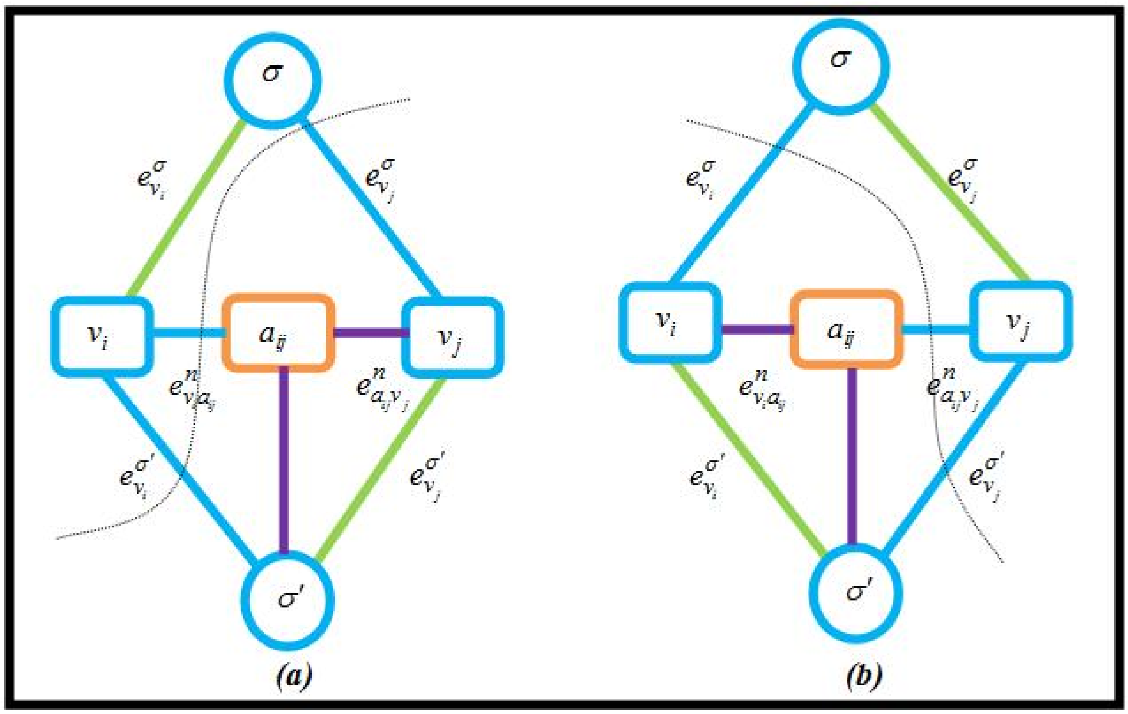

Figure 3.14 (a) $\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \bigcap C$ *for* $e^{\sigma'}_{v_i}, e^{\sigma}_{v_j} \in C$ *(b)* $\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \bigcap C$ *for* $e^{\sigma}_{v_i}, e^{\sigma'}_{v_j} \in C$

if C is a cut it must contain the edge $e^{n}_{v_i a_{ij}}$. It remains to prove that, no edge other than $e^{n}_{v_i a_{ij}}$ from the triplet should be contained in C. For that, note that,

$$\begin{aligned}
\left|e^{n}_{v_i a_{ij}}\right| &= \psi_{v_i, v_j}(\mathrm{X}(v_i), \sigma) \le \psi_{v_i, v_j}(\mathrm{X}(v_i), \mathrm{X}(v_j)) + \psi_{v_i, v_j}(\mathrm{X}(v_j), \sigma) (\because \psi_{v_i, v_j} \text{ is a metric}) \\
&= \psi_{v_i, v_j}(\mathrm{X}(v_i), \mathrm{X}(v_j)) + + \psi_{v_i, v_j}(\sigma, \mathrm{X}(v_j)) (\because \psi_{v_i, v_j} \text{ is a metric}) \\
&= \left|e^{\sigma'}_{a_{ij}}\right| + \left|e^{n}_{a_{ij} v_j}\right|
\end{aligned}$$

As C is a minimum cut, it will contain edge with minimum weight. Thus, it only contains $e^{n}_{v_i a_{ij}}$.

With the similar argument, we can prove the second case: if $e^{\sigma'}_{v_i}, e^{\sigma}_{v_j} \in C$, where C is a minimum cut on G, $\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \bigcap C = e^{n}_{v_i a_{ij}}$. Refer to Figure 3.14 (*a*).

COROLLARY 3.2.2.8

If C is a minimum cut and (v_i, v_j) *with* $X(v_i) \neq X(v_j)$ *are neighboring vertices, the total weight of edges common to* $\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$ *and C is* $\psi_{v_i,v_j}(X_C(v_i), X_C(v_j))$.

Proof: To prove that, $\left|\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \psi_{v_i,v_j}(X_C(v_i), X_C(v_j))$.

There are four cases:

Case *(i)*: $e^{\sigma}_{v_i}, e^{\sigma}_{v_j} \in C$. Then, by Lemma 3.2.2.6, there is no edge common to $\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$ and *C*. Thus,

$$\left|\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \left|\{\ \}\right| = 0.$$

Also, $\psi_{v_i,v_j}(X_C(v_i), X_C(v_j)) = \psi_{v_i,v_j}(\sigma, \sigma) = 0$ as ψ_{v_i,v_j} is a metric

Thus, $\left|\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \psi_{v_i,v_j}(X_C(v_i), X_C(v_j))$.

Case *(ii)*: $e^{\sigma'}_{v_i}, e^{\sigma'}_{v_j} \in C$ Then, by Lemma 3.2.2.6, the only edge common to $\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$ *and C* is $e^{\sigma'}_{a_{ij}}$. Thus,

$$\left|\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \left|e^{\sigma'}_{a_{ij}}\right| = \psi_{v_i,v_j}(X(v_i), X(v_j)).$$

Also, $\psi_{v_i,v_j}(X_C(v_i), X_C(v_j)) = \psi_{v_i,v_j}(X(v_i), X(v_j))$ $(\because e^{\sigma'}_{v} \in C \Rightarrow X_C(v) = X(v))$

Thus, $\left|\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \psi_{v_i,v_j}(X(v_i), X(v_j)) = \psi_{v_i,v_j}(X_C(v_i), X_C(v_j))$.

Case *(iii)*: $e^{\sigma'}_{v_i}, e^{\sigma}_{v_j} \in C$. Then, by Lemma 3.2.2.7, the only edge common to both $\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$ *and C* is $e^{n}_{v_i a_{ij}}$.

$$\left|\left\{e^{n}_{v_i a_{ij}}, e^{n}_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \left|e^{n}_{v_i a_{ij}}\right| = \psi_{v_i,v_j}(X(v_i), \sigma).$$

Also, $\psi_{v_i,v_j}(X_C(v_i), X_C(v_j)) = \psi_{v_i,v_j}(X(v_i), \sigma)\left(\because X_C(v_i) = X(v_i)\right)$

Thus, $\left|\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \psi_{v_i, v_j}(X_C(v_i), X_C(v_j))$.

Case ***(iv):*** $e^{\sigma}_{v_i}, e^{\sigma'}_{v_j} \in C$. Then, by Lemma 3.2.2.7, the only edge common to both $\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\}$ *and* C is $e^n_{a_{ij} v_j}$.

$\left|\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \left|e^n_{a_{ij} v_j}\right| = \psi_{v_i, v_j}(\sigma, X(v_j))$.

Also, $\psi_{v_i, v_j}(X_C(v_i), X_C(v_j)) = \psi_{v_i, v_j}(\sigma, X(v_j))\left(\because X_C(v_j) = X(v_j)\right)$

Thus, $\left|\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C\right| = \psi_{v_i, v_j}(X_C(v_i), X_C(v_j))$.

This proves the corollary.

DEFINITION

A cut C on the network G is said to be a ***basic cut (or b – cut)***, if it satisfies the following properties:

(i) For every pair of vertices (v_i, v_j) *with* $X(v_i) = X(v_j)$

$$e^n_{v_i v_j} \cap C = \begin{cases} \{\ \}, \text{if } e^{\sigma}_{v_i}, e^{\sigma}_{v_j} \in C \\ \{\ \}, \text{if } e^{\sigma'}_{v_i}, e^{\sigma'}_{v_j} \in C \\ e^n_{v_i v_j}, \text{if } e^{\sigma'}_{v_i}, e^{\sigma}_{v_j} \in C \\ e^n_{v_i v_j}, \text{if } e^{\sigma}_{v_i}, e^{\sigma'}_{v_j} \in C \end{cases} \tag{3.18}$$

(ii) For every pair of vertices (v_i, v_j) *with* $X(v_i) \neq X(v_j)$

$$\left\{e^n_{v_i a_{ij}}, e^n_{a_{ij} v_j}, e^{\sigma'}_{a_{ij}}\right\} \cap C = \begin{cases} \{\ \}, \text{if } e^{\sigma}_{v_i}, e^{\sigma}_{v_j} \in C \\ e^{\sigma'}_{a_{ij}}, \text{if } e^{\sigma'}_{v_i}, e^{\sigma'}_{v_j} \in C \\ e^n_{v_i a_{ij}}, \text{if } e^{\sigma'}_{v_i}, e^{\sigma}_{v_j} \in C \\ e^n_{a_{ij} v_j}, \text{if } e^{\sigma}_{v_i}, e^{\sigma'}_{v_j} \in C \end{cases} \tag{3.19}$$

NOTE

Note that, every minimum cut must is b – cut, but not every b – cut is a minimum cut. For example, consider G with $V = \{v_1, v_2, a_{v_1 v_2}\}$ with $\{v_1, v_2\} \in N$ satisfying $X(v_1) \neq X(v_2)$, and

$E = \{e_{v_1}^{\sigma}, e_{v_1}^{\sigma'}, e_{v_2}^{\sigma}, e_{v_2}^{\sigma'}, e_{a_{v_1v_2}v_2}^{n}, e_{v_1a_{v_1v_2}}^{n}, e_{a_{v_1v_2}}^{\sigma'}\}$, where, $\left|e_{v_1}^{\sigma}\right| = 50, \left|e_{v_1}^{\sigma'}\right| = 2,$ $\left|e_{v_2}^{\sigma}\right| = 50,$ $\left|e_{v_2}^{\sigma'}\right| = 2,$ $\left|e_{a_{v_1v_2}v_2}^{n}\right| = 50,$ $\left|e_{v_1a_{v_1v_2}}^{n}\right| = 50,$ $\left|e_{a_{v_1v_2}}^{\sigma'}\right| = 2$ .Consider cuts $C = \{e_{v_1}^{\sigma'}, e_{v_1a_{v_1v_2}}^{n}, e_{v_2}^{\sigma}\}$ and $C' = \{e_{v_1}^{\sigma'}, e_{a_{v_1v_2}}^{\sigma'}, e_{v_2}^{\sigma'}\}$. These cuts have weights or costs given by,

$$|C| = \left|e_{v_1}^{\sigma'}\right| + \left|e_{v_1a_{v_1v_2}}^{n}\right| + \left|e_{v_2}^{\sigma}\right| = 2 + 50 + 50 = 102$$

$$|C'| = \left|e_{v_1}^{\sigma'}\right| + \left|e_{a_{v_1v_2}}^{\sigma'}\right| + \left|e_{v_2}^{\sigma'}\right| = 2 + 2 + 2 = 6 .$$

It can be easily checked that, no cut on *G* can have cost less than 6. Thus, C' is a minimum cut on *G*.

The cut $C = \{e_{v_1}^{\sigma'}, e_{v_1a_{v_1v_2}}^{n}, e_{v_2}^{\sigma}\}$ is a b – cut as it satisfies both (3.18) and (3.19). But, it is not a minimum cut. On the contrary, C' is a minimum cut and also satisfies both (3.18) and (3.19) and hence, is a b – cut.

THEOREM 3.2.2.9

There is a one to one correspondence between the set of all b – cuts C on G and the set of all labeling those are single σ-growth move away from the labeling X.

Proof: First, we attempt to show that, every labeling uniquely corresponds to a b – cut. For that, consider a labeling *X* that is single σ- growth move far from *X*.

We define cut *C* as follows:

$e_v^{\sigma} \in C$ and $e_{v_j}^{\sigma'} \notin C$, if $X(v) = \sigma$

$e_v^{\sigma} \notin C$ and $e_{v_j}^{\sigma'} \in C$, if $X(v) \neq \sigma$

For non-terminal edges, we follow (3.18) and (3.19).

C defined in this way is obviously a b – cut as it satisfies both (3.18) and (3.19).

Now, we have to prove that, each b – cut leads to a labeling that is single σ- growth move far from X. For that, let *C* be a b - cut.

Then, we can define a labeling X_C as follows:

$$X_C(v) = \begin{cases} \sigma, & \text{if } e_v^{\sigma} \in C \text{ and } e_v^{\sigma'} \notin C \\ X_C(v), & \text{if } e_v^{\sigma'} \in C \text{ and } e_v^{\sigma} \notin C \end{cases}$$

This proves the one to one correspondence between set of all b – cuts *C* on *G* and the set of all labeling those are single σ-growth move away from the labeling X.

THEOREM 3.2.2.10

Let C be an b – cut. Then, the value of labeling X_C corresponding to a C and the cost of C are equal.

Proof: We are asked to prove that, $O(X_C)$ and $|C|$ are equal, for every b – cut *C*.

Note that, $|C|$ is the sum of weight of all the edges contained in the cut *C*. *C* has two types of edges: terminal edges and non – terminal edges. To be more specific, terminal edges and two types of non-terminal edges as shown below:

$$C = \bigcup_{v\in V}\{e_v^{\sigma} \text{ or } e_v^{\sigma'}\} \cup \bigcup_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u)=X(v)}} \{e_{uv}^n / X_C(u) \neq X_C(v)\} + \bigcup_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u)\neq X(v)}} \{e_{ua_{uv}}^n \text{ or } e_{a_{uv}v}^n \text{ or } e_{a_{uv}}^{\sigma'}\} \quad (3.20)$$

The first term in (3.20) contains exactly one terminal link for every vertex *v* of *G* as proved in Lemma 3.2.2.1. The second term in (3.20) contains non – terminal links for only those pair of neighbouring vertices *u* and *v* of *G*, for which *C* contains terminal edges corresponding to *u* and *v* associated with different terminals.

Thus,

$$|C| = \sum_{v\in V}\left|e_v^{\sigma} \text{ or } e_v^{\sigma'}\right| + \sum_{\substack{\{u,v\}\in N \\ u,v\in V_1\cup V \\ X(u)=X(v)}} \left|\left\{e_{uv}^n\right\}\cap C\right| + \sum_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u)\neq X(v)}} \left|\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^{\sigma'}\right\}\cap C\right| \quad (3.21)$$

Note that, $\left|e_v^{\sigma_1} \text{ or } e_v^{\sigma_2}\right|$ refers to the weight of the terminal edge corresponding to *v* contained in the cut *C*. But,

$$\left.\begin{aligned} \left|e_v^{\sigma}\right| &= \varphi_v(\sigma) \\ \left|e_v^{\sigma'}\right| &= \varphi_v(X(v)) \end{aligned}\right\} = \varphi_v(X_C(v))$$

Thus,

$$\left|\mathrm{e}_v^{\sigma_1} \text{ or } \mathrm{e}_v^{\sigma_2}\right| = \varphi_v(X_C(v))$$

$$\sum_{v\in V}\left|\mathrm{e}_v^{\sigma_1} \text{ or } \mathrm{e}_v^{\sigma_2}\right| = \sum_{v\in V}\varphi_v(X_C(v)) \quad (3.22)$$

Also, $\left|e_{uv}^n \cap C\right| = \psi_{u,v}(X_C(u), X_C(v))$ by Lemma 3.2.2. 5. Thus, the second term of (3.21) becomes,

$$\sum_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u)=X(v)}} \left|e_{uv}^n \cap C\right| = \sum_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u)=X(v)}} \psi_{u,v}(X_C(u), X_C(v)) \quad (3.23)$$

From corollary 3.2.2.8,

$\left|\left\{e^{n}_{ua_{uv}}, e^{n}_{a_{uv}v}, e^{\sigma'}_{a_{uv}}\right\} \cap C\right| = \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v))$, which implies that,

$$\sum_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u)\neq X(v)}} \left|\left\{e^{n}_{ua_{uv}}, e^{n}_{a_{uv}v}, e^{\sigma'}_{a_{uv}}\right\} \cap C\right| = \sum_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u)\neq X(v)}} \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v)) \tag{3.24}$$

Now, (3.21) to (3.24) together imply that,

$$\begin{aligned}
|C| &= \sum_{v\in V} \varphi_v(X_C(v)) + \sum_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u) = X(v)}} \psi_{u,v}(X_C(u), X_C(v)) + \sum_{\substack{\{u,v\}\in N \\ u,v\in V \\ X(u)\neq X(v)}} \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v)) \\
&= \sum_{v\in V} \varphi_v(X_C(v)) + \sum_{\substack{\{u,v\}\in N \\ u,v\in V}} \psi_{u,v}(X_C(u), X_C(v)) \\
&= O(\mathrm{X}_C)
\end{aligned}$$

This proves the theorem.

COROLLARY 3.2.2.11

The labeling X_C corresponding to minimum cut C on G minimizes the objective function.

Proof: As minimum cut *C* is a b – cut, it satisfies Theorem 3.2.2.10. Thus, for minimum cut *C*,

$|C| = O(X_C)$. As *C* is a minimum cut on *G*, its cost is the least among all possible cuts on *G*.

Thus, X_C minimizes the objective function.

Till now, the focus of the discussion was on to justify the usage of the algorithm with different move spaces (in particular, interchange and growth moves) and to show that, at the end of each iteration, the labeling returned by the minimum cut on the network flow is local minimum of objective function on the space of all the labeling those are single move far from the initial labeling. Now, shifting the focus towards the performance of the algorithm, we move towards the discussion of running time of the algorithm and how far is the solution produced by the algorithm from the global minimum solution.

THOREM 3.2.2.12

The cost of the labeling X_C corresponding to minimum cut C on G (which is σ- growth move far from the initial labeling) is less than some constant times the cost of the labeling $\tilde{X}$ that is global optimum solution of the objective function. The value of the constant depends on the form of $\psi_{u,v}$.

Proof: Let $\sigma \in \Omega$ be an arbitrary label. Define $V_\sigma = \left\{v \in V \mid \tilde{X}(v) = \sigma\right\}$. We can define a labeling X' that is one σ- growth move far from the labeling X_C as follows:

$$X'(v) = \begin{cases} \sigma, & \text{if } v \in V_\sigma \\ X_C(v), \forall v \in V \setminus V_\sigma \end{cases}$$

As X_C is a local minimum over σ - growth move space, $O(X_C) \le O(X')$ (3.25)

Note that, there are two mutually exclusive and exhaustive subsets V_σ and $V - V_\sigma$ of V, on which we can define restriction of the objective function as follows:

$$O_{V_\sigma}(X) = \sum_{v \in V_\sigma} \varphi_v(x_v) + \sum_{\substack{\{v,w\} \in N \\ v,w \in V_\sigma}} \psi_{v,w}(x_v, x_w) \text{ and}$$

$$O_{V \setminus V_\sigma}(X) = \sum_{v \in V \setminus V_\sigma} \varphi_v(x_v) + \sum_{\substack{\{v,w\} \in N \\ v,w \in V \setminus V_\sigma}} \psi_{v,w}(x_v, x_w) \text{ where } X \text{ is any labeling.}$$

We define a set B_σ of neighboring pixels as, $B_\sigma = \{(v,w) \in N \mid v \in V_\sigma \text{ and } w \in V \setminus V_\sigma\}$ and define the restriction of objective function on B_σ as

$$O_{B_\sigma}(X) = \sum_{\{v,w\} \in B_\sigma} \psi_{v,w}(x_v, x_w)$$

Clearly,

$$O(X_C) = O_{V_\sigma}(X_C) + O_{V \setminus V_\sigma}(X_C) + O_{B_\sigma}(X_C) \quad (3.26)$$

$$O(\tilde{X}) = O_{V_\sigma}(\tilde{X}) + O_{V \setminus V_\sigma}(\tilde{X}) + O_{B_\sigma}(\tilde{X}) \quad (3.27)$$

$$O(X') = O_{V_\sigma}(X') + O_{V \setminus V_\sigma}(X') + O_{B_\sigma}(X') \quad (3.28)$$

For pair of neighbouring pixels (v,w), we can define

$$l_{v,w} = \min_{\sigma,\sigma'} \psi_{v,w}(\sigma, \sigma') \text{ and } h_{v,w} = \max_{\sigma,\sigma'} \psi_{v,w}(\sigma, \sigma') \quad (3.29)$$

Let $t = \min_{v,w} \left(\frac{h_{v,w}}{l_{v,w}} \right)$ (3.30)

Then, $O_{B_\sigma}(X') \le \max_{X:V \to \Omega} O_{B_\sigma}(X) \le \sum_{(v,w) \in B_\sigma} h_{v,w} \le \sum_{(v,w) \in B_\sigma} \left\{ \left(\frac{h_{v,w}}{l_{v,w}} \right) l_{v,w} \right\}$

$$\le \sum_{(v,w) \in B_\sigma} t l_{v,w} \le \sum_{(v,w) \in B_\sigma} t \psi_{v,w}(\sigma, \sigma') = t O_{B_\sigma}(\tilde{X})$$

Thus, $O_{B_\sigma}\left(X'\right) \le tO_{B_\sigma}\left(\tilde{X}\right)$ (3.31)

Now, it is easy to observe that,

$$O_{V_\sigma}\left(X'\right) = O_{V_\sigma}\left(\tilde{X}\right) \tag{3.32}$$

$$O_{V \setminus V_\sigma}\left(X'\right) = O_{V \setminus V_\sigma}\left(X_C\right) \tag{3.33}$$

Equations (3.26) and (3.28) together with (3.25) implies that,

$$O_{V_\sigma}(X_C) + O_{V \setminus V_\sigma}(X_C) + O_{B_\sigma}(X_C) \le O_{V_\sigma}(X') + O_{V \setminus V_\sigma}(X') + O_{B_\sigma}(X')$$

Equation (3.32) implies that,

$$O_{V_\sigma}(X_C) + O_{B_\sigma}(X_C) \le O_{V_\sigma}(X') + O_{B_\sigma}(X')$$

Equations (3.32) and (3.33) imply that,

$$O_{V_\sigma}(X_C) + O_{B_\sigma}(X_C) \le O_{V_\sigma}(\tilde{X}) + tO_{B_\sigma}\left(\tilde{X}\right) \tag{3.34}$$

$$\sum_{\sigma \in \Omega}\left(O_{V_\sigma}(X_C) + O_{B_\sigma}(X_C)\right) \le \sum_{\sigma \in \Omega}\left(O_{V_\sigma}(\tilde{X}) + tO_{B_\sigma}\left(\tilde{X}\right)\right) \tag{3.35}$$

Define $B = \bigcup_{\sigma \in \Omega} B_\sigma$. For every pair of neighbor $(v, w) \in B$ and every pair of labels $\left(\sigma, \sigma'\right)$, the term $\psi_{v,w}\left(\sigma, \sigma'\right)$ of $O_{B_\sigma}(X_C)$ is counted twice in the LHS of (3.34), as $\psi_{v,w}\left(X_C(v) = \sigma, \sigma'\right)$ and $\psi_{v,w}\left(\sigma', X_C(v) = \sigma\right)$ for B_σ and $B_{\sigma'}$ respectively. In similar manner, the term $\psi_{v,w}\left(\sigma, \sigma'\right)$ of $O_{B_\sigma}\left(\tilde{X}\right)$ appears 2t times on the R.H.S. of (3.34). Hence, (3.35) becomes,

$$O(X_C) + O_B(X_C) \le O(\tilde{X}) + (2t-1)O_B\left(\tilde{X}\right) \le 2tO_B\left(\tilde{X}\right).$$

This proves that, $O(X_C) + O_B(X_C) \le 2tO_B\left(\tilde{X}\right)$ and thus, $O(X_C) \le 2tO_B\left(\tilde{X}\right)$.

Where, 2*t* is a constant dependant on the form of $\psi_{u,v}$. This proves the result.

THOREM 3.2.2.13

The algorithm 3.5 has a linear running time.

Proof: We need to prove that, if there are n pixels, the algorithm will take no more than some constant times n seconds to return the labeling X_C. For that, an assumption is made to facilitate the reasoning for the bound of running time: φ_v and $\psi_{u,v}$ are independent of *n*, the total number of pixels the image is composed of.

As the algorithm is to be provided with the initial labeling, we choose the initial labeling X as,

$$X(v) = \left\{ \sigma \mid \varphi_v(\sigma) = \min_{\sigma' \in \Omega} \varphi_v(\sigma') \right\}.$$

We may assume that, $\varphi_v(\sigma) = \min_{\sigma' \in \Omega} \varphi_v(\sigma') = 0, \forall v \in V$

We want to minimize the objective function O defined as,

$$O(X) = \sum_{v \in V} \varphi_v(x_v) + \sum_{\{v,w\} \in N} \psi_{v,w}(x_v, x_w). \tag{3.25}$$

But, the initial labeling X is chosen so that, the first term of O(X) is zero, for the initial labeling, O(X) is simply

$$O(X) = \sum_{\{v,w\} \in N} \psi_{v,w}(x_v, x_w). \tag{3.26}$$

Note that, our special construction of initial labeling doesn't give rise to any ambiguity as minimizing (3.26) is equivalent to minimizing the objective function

$$O(X) = \sum_{v \in V} \left(\varphi_v(x_v) - \left(\min_{\sigma' \in \Omega} \varphi_v(\sigma') \right) \right) + \sum_{\{v,w\} \in N} \psi_{v,w}(x_v, x_w) \ .$$

$$\left(\because \varphi_v(x_v) - \left(\min_{\sigma' \in \Omega} \varphi_v(\sigma') \right) = \varphi_v(x_v) \text{ as } \min_{\sigma' \in \Omega} \varphi_v(\sigma') = 0 \right)$$

Let $t = \max_{\substack{v,w \in V \\ \sigma,\sigma' \in \Omega}} \psi_{v,w}(\sigma, \sigma')$ and $l = \min_{\substack{u,v,w \in V \\ \sigma_1,\sigma_2,\sigma_3 \in \Omega}} \left| \psi_{v,w}(\sigma_1, \sigma_2) - \varphi_u(\sigma_3) \right|$,

Then, both t and l are independent of n.

Thus, from 3.26,

$$O(X) = \sum_{\{v,w\} \in N} \psi_{v,w}(x_v, x_w) \leq \sum_{\{v,w\} \in N} t$$

If there are p labels, or in other words, if $|\Omega| = p$, then there will be p and p^2 iterations in one cycle of the algorithm if it involves growth and interchange move space respectively. Algorithm with growth move takes p iterations as corresponding to every label σ, the algorithm has to check for possible cost efficient labelling corresponding to σ growth move. Similarly, for every pair of labelling (σ, σ'), the algorithm has to perform iteration for possible cost efficient labelling corresponding to interchange move involving these two labels (and there are such p^2 pairs of labels).

At the end of every cycle, the revised labelling improves in terms of penalty assigned by objective function at least by l. i.e. If at the end of cycle 1 and 2, the labelling gets updated by X_C and $X_{C'}$ respectively, $O(X_{C'}) \leq O(X_C) - l$. Thus, maximum no. of cycles needed by the algorithm to converge is $\sum_{\{v,w\}\in N} \psi_{v,w}(x_v, x_w) = \sum_{\{v,w\}\in N} \binom{t}{l}$, which is a degree 1 polynomial in *n*. Thus, the algorithm takes *O (n)* time and hence the running time of the algorithm is linear.

3.3 SEGMENT WISE CONSTANT STRUCTURE

In this sub-section, we attempt to study some models which are appropriate for segment wise smooth structure. Segment wise constant structure, being a special case of segment wise smooth structure, can be addressed by expansion and growth move models presented in the earlier sub-section. In this sub-section, we begin with a model that deals with multiple terminals. In earlier sub section we considered network flow with two terminals, called source and sink. That's why, only two labels at a time can be dealt with in the single iteration. In this model, we will consider a graph with multiple terminals. The graph will have a terminal corresponding to each label.

It should be noted that, segment wise constant structure preserves the discontinuity whenever it is essential and it is the simplest structure with this property. One possible way of defining $\psi_{v,w}(x_v, x_w)$ is as follows:

$$\psi_{v,w}(x_v, x_w) = c_{vw}\ I(x_v, x_w)$$

where,

$$I(x_v, x_w) = \begin{cases} 0, & \text{if } |x_v - x_w| = 0 \\ 1, & \text{otherwise} \end{cases}$$

The value of $I(x_v, x_w)$ is independent of the labels in case if the labels under consideration are not the same. i.e., even if the labels differ drastically, the penalty incurred on assignment of such labels to neighbouring pixels *v* and *w* is constant (i.e., c_{uv}).

Thus, the objective to be optimized takes the form,

$$O(X) = \sum_{v\in V} \varphi_v(x_v) + \sum_{\{v,w\}\in N} c_{vw}\ I(x_v, x_w) \tag{3.27}$$

3.3.1 GRAPH CUT MODEL WITH MULTIPLE TERMINAL VERTICES

The graph G will have *p* terminals $\{\sigma_1, \sigma_2, ..., \sigma_p\}$ and *n* non-terminal vertices $\{v_1, v_2, ..., v_n\}$. For every non-terminal vertex $v_i (1 \le i \le n)$, there are *p* terminal edges $\left\{e_{v_i}^{\sigma_1}, e_{v_i}^{\sigma_2}, ..., e_{v_i}^{\sigma_p}\right\}$, one corresponding to each terminal. For every pair of neighboring vertex (v_i, v_j), graph G has a non-terminal edge $e_{v_i v_j}^{n}$. The weights to these edges are assigned as follows:

$$\left|e_{v_i}^{\sigma_j}\right| = k_{v_i} - \varphi_{v_i}(\sigma_j) \tag{3.28}$$

$$\left|e^{n}_{v_i v_j}\right| = c_{v_i v_j} \tag{3.29}$$

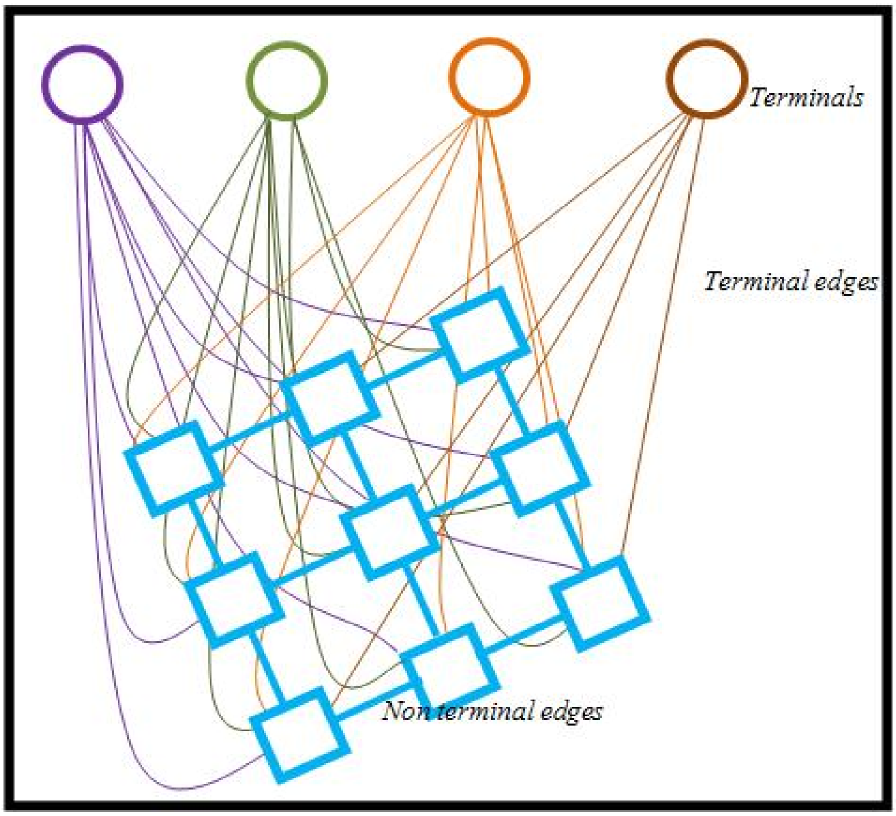

Figure 3.15 Graph for multiple terminal vertices

Any cut on G must separate all the terminals. In other words, If C is a cut on G, there should be no path connecting any two terminals in G \ C. It can be easily proved that, if C is a cut, it must leave every vertex (non-terminal) connected to unique terminal. Thus, the cut naturally gives rise to labeling X_C defined as follows:

$$X_C(v) = \sigma, \text{if } e^{\sigma}_{v} \notin C \tag{3.30}$$

Note that, the cut cannot contain all terminal edges corresponding to some non-terminal vertex *v*, as in that case removing one of the terminal edges from the cut *C* will lead to a subset of *C* which is a cut and thus leads to a contradiction. On the contrary, the cut cannot leave more than one terminal edges corresponding to some vertex *v* in *G* \ *C*, because in that case, there will be a direct path connecting both the terminals via vertex *v*, which contradicts with the fact that *C* is a cut. (Refer to Figure 3.16) This leads to

LEMMA 3.3.1.1

If C is a cut on the graph G with multiple terminals, it must contain all terminal edges except one corresponding to each non-terminal vertex.

It can be observed that, if neighboring vertices v and w are connected to different terminals in $G \setminus C$, C must contain the non-terminal edge e_{vw}^{n}, because if C does not contain the edge, there will be a direct path joining vertices v and w in $G \setminus C$. Similarly, if neighboring vertices v and w are connected to same terminal in $G \setminus C$, C must not contain the non-terminal edge e_{vw}^{n}, because if C contain the edge, C \ { e_{vw}^{n} } will also be a cut. This proves the following result:

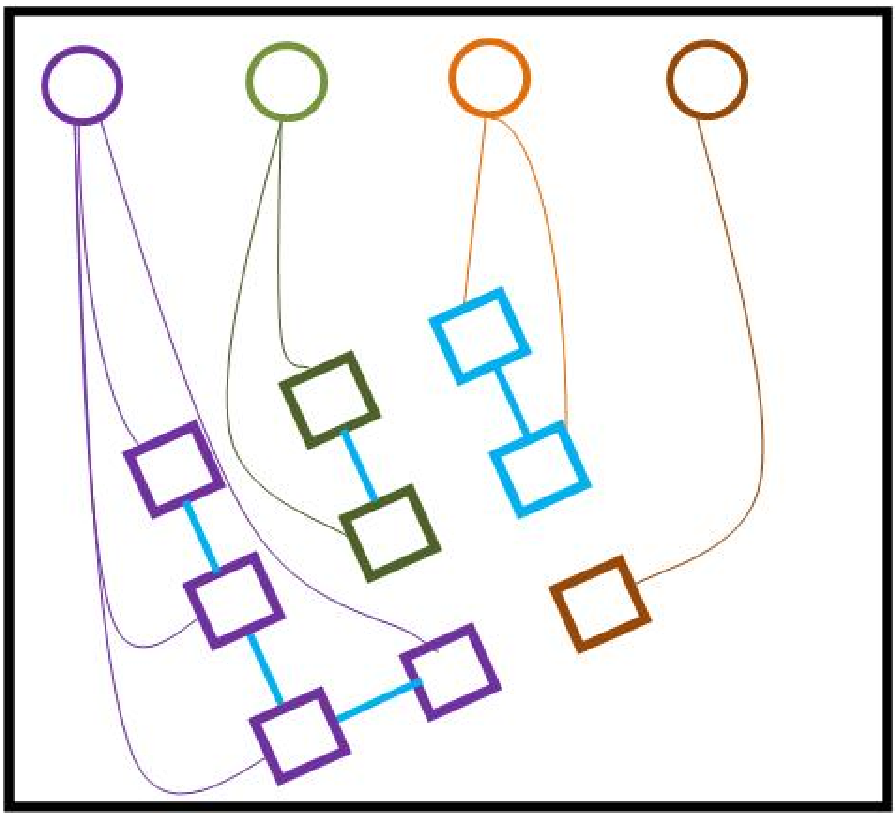

Figure 3.16 Graph G \ C for cut C, which leaves each vertex connected to only single terminal

LEMMA 3.3.1.2

For neighboring non-terminal vertices v and w,

(i) *If* $X_C(v) = X_C(w)$, $e_{vw}^{n} \notin C$.

(ii) *If* $X_C(v) \neq X_C(w)$, $e_{vw}^{n} \in C$.

THEOREM 3.3.3

If C is a cut on graph G with multiple terminals, then the cost of the labeling X_C *and* $O(X_C)$ *(mentioned in (3.27)) differ by a constant.*

Proof: As C contains non-terminal and terminal edges, the cost of C is sum of cost of both types of edges present in C.

The sum of cost of all terminal edges of C will be, $\sum_{v\in V}\sum_{\substack{\sigma\in\Omega \\ e_v^\sigma\in C}}\left|e_v^\sigma\right| = \sum_{v\in V}\sum_{\substack{\sigma\in\Omega \\ e_v^\sigma\in C}}\left(k_v-\varphi_v(\sigma)\right)$

$$\therefore \sum_{v\in V}\sum_{\substack{\sigma\in\Omega \\ e_v^\sigma\in C}}\left|e_v^\sigma\right| = \sum_{v\in V}\sum_{\substack{\sigma\in\Omega \\ e_v^\sigma\in C}} k_v - \sum_{v\in V}\sum_{\substack{\sigma\in\Omega \\ e_v^\sigma\in C}}\varphi_v(\sigma)$$

$$= (\mathrm{p}-1)\sum_{v\in V} k_v - \sum_{v\in V}\sum_{\sigma\in\Omega}\varphi_v(\sigma) + \sum_{v\in V}\varphi_v(X_C(v))$$

Thus, the sum of all terminal edges of cut *C* is,

$$\sum_{v\in V}\sum_{\substack{\sigma\in\Omega \\ e_v^\sigma\in C}}\left|e_v^\sigma\right| = (\mathrm{p}-1)\sum_{v\in V} k_v - \sum_{v\in V}\sum_{\sigma\in\Omega}\varphi_v(\sigma) + \sum_{v\in V}\varphi_v(X_C(v)) \qquad (3.31)$$

By lemma 3.3.1.2, only those non-terminal edges e_{vw}^n will be contained in C, which are connected to different terminal vertices in G \ C. Thus, the sum of all non-terminal edges of C is,

$$\sum_{\substack{\{v,w\}\in N \\ X_C(v)\neq X_C(w)}}\left|e_{vw}^n\right| = \sum_{\substack{\{v,w\}\in N \\ v,w\in V}} c_{vw}\ \ I(X_C(v),X_C(w))\left(\because I(X_C(v),X_C(w))=0,\ \text{if}\ \ X_C(v)=X_C(w)\right)$$

Thus,

$$\sum_{\substack{\{v,w\}\in N \\ X_C(v)\neq X_C(w)}}\left|e_{vw}^n\right| = \sum_{\substack{\{v,w\}\in N \\ v,w\in V}} c_{vw}\ \ I(X_C(v),X_C(w)) \qquad (3.32)$$

Summing up (3.31) and (3.32), we get the sum of weights of all edges of C, which is,

$$\left|C\right| = (\mathrm{p}-1)\sum_{v\in V} k_v - \sum_{v\in V}\sum_{\sigma\in\Omega}\varphi_v(\sigma) + \sum_{v\in V}\varphi_v(X_C(v)) + \sum_{\substack{\{v,w\}\in N \\ v,w\in V}} c_{vw}\ \ I(X_C(v),X_C(w))$$

$$= \left((\mathrm{p}-1)\sum_{v\in V} k_v - \sum_{v\in V}\sum_{\sigma\in\Omega}\varphi_v(\sigma)\right) + \left(\sum_{v\in V}\varphi_v(X_C(v)) + \sum_{\substack{\{v,w\}\in N \\ v,w\in V}} c_{vw}\ \ I(X_C(v),X_C(w))\right)$$

$$=\left((\mathrm{p}-1)\sum_{v\in V}k_v-\sum_{v\in V}\sum_{\sigma\in\Omega}\varphi_v(\sigma)\right)+O\left(X_C\right) \tag{3.33}$$

This proves the result as the term in the first bracket is a constant independent of the cut C.

The theorem lays a foundation for corollary 3.3.1.4 presented below.

COROLLARY 3.3.1.4

If C is a minimum cut on graph G with multiple terminals, then objective function (3.27) has a local minimum at X_C.

Proof: By theorem 3.3.1.3,

$O\left(X_C\right)=|C|+\text{constant}$, which is minimum when $|C|$ is minimum. It is obvious that, $|C|$ is minimum when C is minimum cut.

The problem can be solved efficiently, when the total no. of terminals are two. When number of terminals is higher than two, the problem is NP – complete. Thus, the exact minimum cannot be targeted. However, approximate solution can be evaluated efficiently. One way to approach the problem, when terminals are more than two, is to find a cut C(σ), which separates terminal σ from remaining all terminals. Define C = $\bigcup_{\substack{\sigma\in\Omega\\ \sigma\neq\sigma_{\max}}} C(\sigma)$, where $\sigma_{\max}=\left\{\sigma'\in\Omega\ /\ C(\sigma')=\max_{\sigma\in\Omega} C(\sigma)\right\}$.

The algorithm working on this approach produces a solution that is within factor of $2\left(1-\frac{1}{p}\right)$. However, this algorithm has two major limitations: (1) there might be some of the vertices of V, which are assigned no label by C. The algorithm assigns the label $\sigma_{\max}$ to such vertices (2) If the cut C is near to the exact minimum, it does not guarantee that, the labeling X_C is also close to exact minimum due to the constant term in (3.33).

3.3.2 GRAPH CUT MODEL WITH SHIFT MOVE SPACE

Any labelling partitions the image and is defined by $V=\{V_\sigma\mid\sigma\in\Omega\}$ where, $V_\sigma=\{v\in V\mid X_v=\sigma\}$. As labelling and partitions are in one to one correspondence, both notions can be used interchangeably.

Consider a one to one function $s:\Omega\rightarrow\{0,1,....,n-1\}$, where L denotes set of labels and n is total no. of labels. Shifts are denoted by an integer $k\in\{0,1,....,n-1\}$. A *k*-shift move changes a labelling X to X' if there exists a set S$\subset V$ such that,

$$s(\mathrm{X}'(v))=\begin{cases}s(X(v))+k, \text{ if } v\in S\\ s(X(v)), \text{ if } v\notin S\end{cases} \tag{3.34}$$

In short, *k*-shift increases a label of some pixels v by *k*. Without loss of generality, we may take *s* to be an identity map.

For given labeling *X* and value of *k*, the aim is to find a labeling which is one *k* – shift move away from *X* and has the minimum value of objective function among all moves satisfying this property. The crucial step of the model is to find the cut of graph *G* constructed with the . For this, we will find minimum cut on graph $G=(V,E)$. The structure of *G* is as follows: The graph has terminals *i, j* and all vertices belonging to *V*. Apart from this, for every pair of neighboring pixels *{u,v}* satisfying $|x_u - x_v| = |k|$, a supplementary vertex a_{uv} is created. Hence, the graph contains three types of vertices i.e. terminal vertices (*i* and *j*), set of all non-terminal vertices ($v \in V$) and supplementary vertices (of the form a_{uv}).

Now, we construct sets S_i and S_j such that $\Omega = (S_i - \alpha) \cup (S_j - \alpha)$ and both $(S_i - \alpha)$ and $(S_j - \alpha)$ gives partition of Ω. Note that, α is a dummy label.

For given label σ and integer *k*, by Unique Factorization Theorem, there exists integers *p* and *r* such that, $\sigma = kp + r$ with $r < k$.

Define binary variable *f*: $\Omega \rightarrow \{i, j\}$ by,

$$f(\sigma) = \begin{cases} i, \text{ if } p \text{ is odd} \\ j, \text{otherwise} \end{cases} \tag{3.35}$$

Now, let us define S_i and S_j as follows:

$$S_i = \{\sigma / f(\sigma) = i\} \cup \{\alpha\}$$

$$S_j = \{\sigma / f(\sigma) = j\} \cup \{\alpha\}$$

If vertex *v* is separated from terminal *i*, *v* is assigned label $X_i'(x_v)$ and if it is separated from *j*, the label to be assigned is $X_j'(x_v)$ where, $X_i'(x_v)$ and $X_j'(x_v)$ are given by,

$$X_i'(x_v) = \begin{cases} x_v, \text{ if } x_v \in S_i \\ x_v + k, \text{ if } x_v + k \in S_i \\ \alpha, \text{ otherwise} \end{cases} \tag{3.36}$$

and

$$X_j'(x_v) = \begin{cases} x_v, \text{ if } x_v \in S_j \\ x_v + k, \text{ if } x_v + k \in S_j \\ \alpha, \text{ otherwise} \end{cases} \tag{3.37}$$

LEMMA 3.3.2.1

Functions X'_i and X'_j are well defined.

Proof: In order to prove that, functions X'_i and X'_j are well defined, we need to prove that both x_v and $x_v + k$ cannot belong to S_m, where $m \in \{i, j\}$. Hence, it is sufficient to prove that, for given σ, (where $\sigma + k$ is in permissible range)

$f(\sigma) \neq f(\sigma + k)$.

For given l, by Unique Factorization Theorem, there exists integers p and r such that, $\sigma + k = kp + r$ with $r < k$. This implies that, $\sigma + k = kp + r + k = k(p+1) + r$ is unique factorization corresponding to, $\sigma + k$. Trivially, one of p and $p+\sigma$ will be odd and the other will be even. This proves that, $f(\sigma) \neq f(\sigma + k)$.

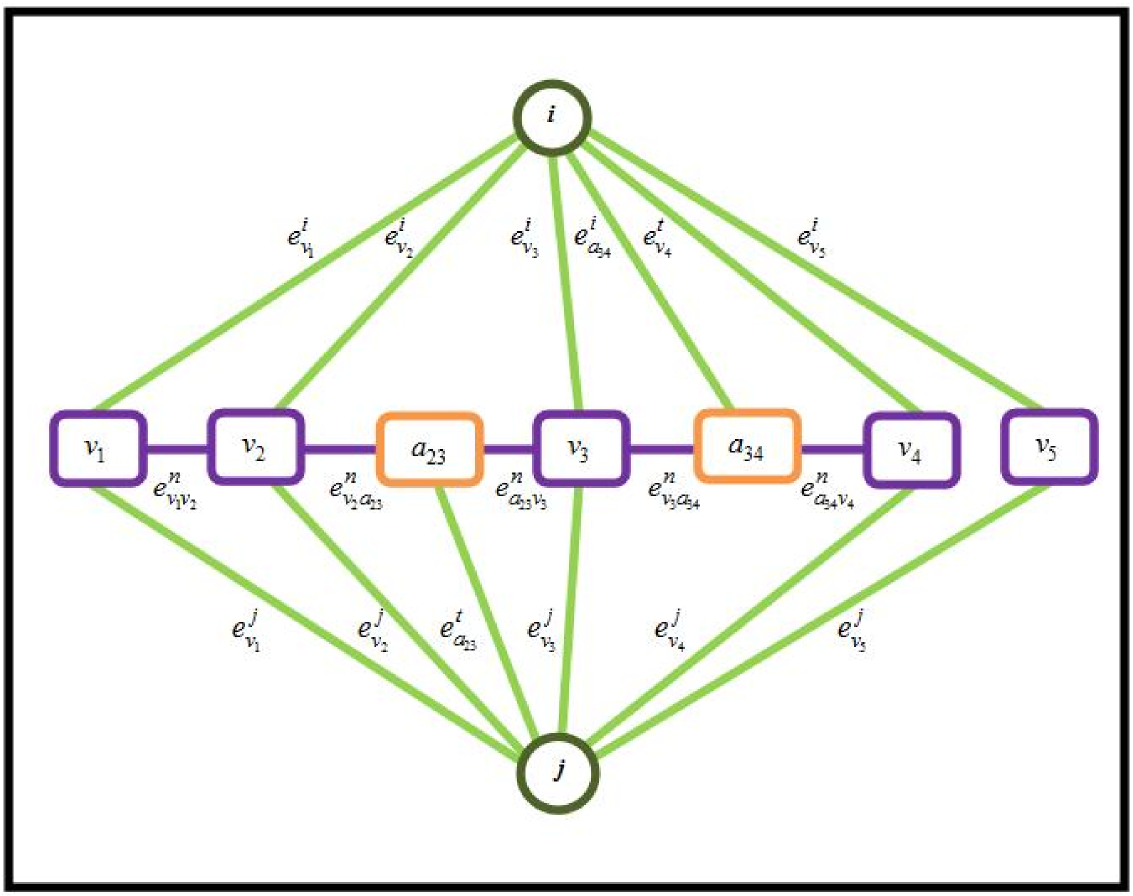

Figure 3.17: Network flow G for graph cut model based on k – shift move

Now, we describe edges of graph $G = (V, E)$ (Refer Figure 3.17). For each pixel *v*, we add two terminal edges e^i_v and e^j_v which connects p with *i* and *j* resp. For every pair of neighboring non-terminal vertices u and v with $X(u)=X(v)$, we add an edge e^n_{uv} joining u and v. For every pair of neighboring vertices u and v satisfying $X(u)\neq X(v)$, but $|X(\mathrm{u})-\mathrm{X(v)}|=|k|$, we have a supplementary vertex a_{uv} and three edges $e^n_{ua_{uv}}, e^n_{a_{uv}v}$ and $e^t_{a_{uv}}$. The edge $e^n_{ua_{uv}}$ connects u with a_{uv}, the edge $e^n_{a_{uv}v}$ connects a_{uv} with v, whereas $e^t_{a_{uv}}$ connects a_{uv} with either i or j as per criteria mentioned in (3.38). For every pair of neighboring vertices u and v with $X(u)\neq X(v)$ and $|X(\mathrm{u})-\mathrm{X(v)}|\neq|k|$, there is no non-terminal edge.

$$
\left.\begin{array}{l}
\text{If } x_u = x_v + k \text{ and } f(x_u)=j,\ e^t_{a_{uv}} \text{ connects } a_{uv} \text{ with } i.\\
\text{If } x_u = x_v + k \text{ and } f(x_u)=i,\ e^t_{a_{uv}} \text{ connects } a_{uv} \text{ with } j.\\
\text{If } x_v = x_u + k \text{ and } f(x_v)=j,\ e^t_{a_{uv}} \text{ connects } a_{uv} \text{ with } i.\\
\text{If } x_v = x_u + k \text{ and } f(x_v)=i,\ e^t_{a_{uv}} \text{ connects } a_{uv} \text{ with } j.
\end{array}\right\} \quad (3.38)
$$

The set $\{u \mid 0\leq x_u + k \leq n-1\}$ gives set of all non-terminal vertices which can be shifted by k units and still the labels are in permissible range. In order to avoid the assignment of dummy label α to any pixel, we define $\varphi_v(\alpha)=\infty$.

Equation (3.39) gives edge weights of all the edges part of the graph.

The weights corresponding to edges are defined as follows:

$$
\left.\begin{array}{l}
|e^j_v| = \varphi_v(x'_j(x_v)),\ \ \forall v\in V\\
|e^i_v| = \varphi_v(x'_i(x_v)),\ \ \forall v\in V\\
|e^n_{ua_{uv}}| = |e^n_{a_{uv}v}| = |e^t_{a_{uv}}| = c_{uv},\ \forall \text{ neighboring pixels}\{u,v\}\text{with}\,|X(u)-X(v)|=|k|\\
|e^n_{uv}| = c_{uv},\ \forall\text{neighboring pixels } \{u,v\}\text{ with } X(u)=X(v)
\end{array}\right\} \quad (3.39)
$$

LEMMA 3.3.2.2

For any non-terminal vertex $v\in V$ and any cut C on G, exactly one terminal edge would be part of C.

Proof: If cut *C* includes both terminal edges, then removal of any of the terminal edges from *C* would be a new cut, which violates the condition of minimality of cut.

If none of the terminal edges is severed by *C*, both the terminal vertices are connected via path formed by these terminal edges. Thus, exactly one terminal edge corresponding to any non-terminal vertex should be contained in the cut.

This gives rise to a natural labeling corresponding to minimum cut C, defined by,

$$x_v^C = \begin{cases} x_i'(x_v), \text{ if } e_v^i \in C \\ x_j'(x_v), \text{ if } e_v^j \in C \end{cases} \quad \forall v \in V \tag{3.40}$$

LEMMA 3.3.2.3

Labeling X^C corresponding to cut C is one k-shift away from the initial labeling.

Proof: From the definition of $x_i'(x_v)$ and $x_j'(x_v)$, it is clear that, possible labels are x_v, $x_v + k$ and *u*. Since weight corresponding to *u* is infinity, a cut C would never contain edge with weight *u*. Possible new labels are only x_v and $x_v + k$, which proves that, the new labeling X^C is one k-shift away from the initial labeling.

This proves the result.

LEMMA 3.3.2.4

Let u and v be neighbors in V with $X(u) = X(v)$. For any cut C and for any non-terminal edge e_{uv}^n,

If cut C contains terminal edges corresponding to i for both u and v (i.e. $e_u^i, e_v^i \in C$) or corresponding to j for both u and v (i.e. $e_u^j, e_v^j \in C$), e_{uv}^n would not be part of C.

Proof: Let C be a cut with $X(u) = X(v)$ and $e_u^i, e_v^i \in C$. To prove that, e_{uv}^n does not belong to C, if possible, assume that, e_{uv}^n is in C. Then, there exists a subset $C_1 = C \setminus \{e_{uv}^n\}$ of C. We will prove that, C_1 is a cut. If possible, assume that, there is a path P connecting both the terminals via e_{uv}^n in G \ C_1. Then, P must have sub-paths P_1 in G \ C_1 joining terminal *j* and non-terminal vertex *u*. Note that, P_1 entirely lies in G \ C. Thus, there exists a path $e_u^i \cup P_1$ in G \ C connecting both the terminals, which is a contradiction with the fact that, C is a cut. Thus, C_1 is a cut. This proves that, e_{uv}^n can not belong to C. (Refer to Figure 3.18(a))

Similarly, it can be proved that, if C is a cut with $X(u) = X(v)$ and $e_u^j, e_v^j \in C$, e_{uv}^n does not belong to C. (Refer to Figure 3.18(b))

This proves the result.

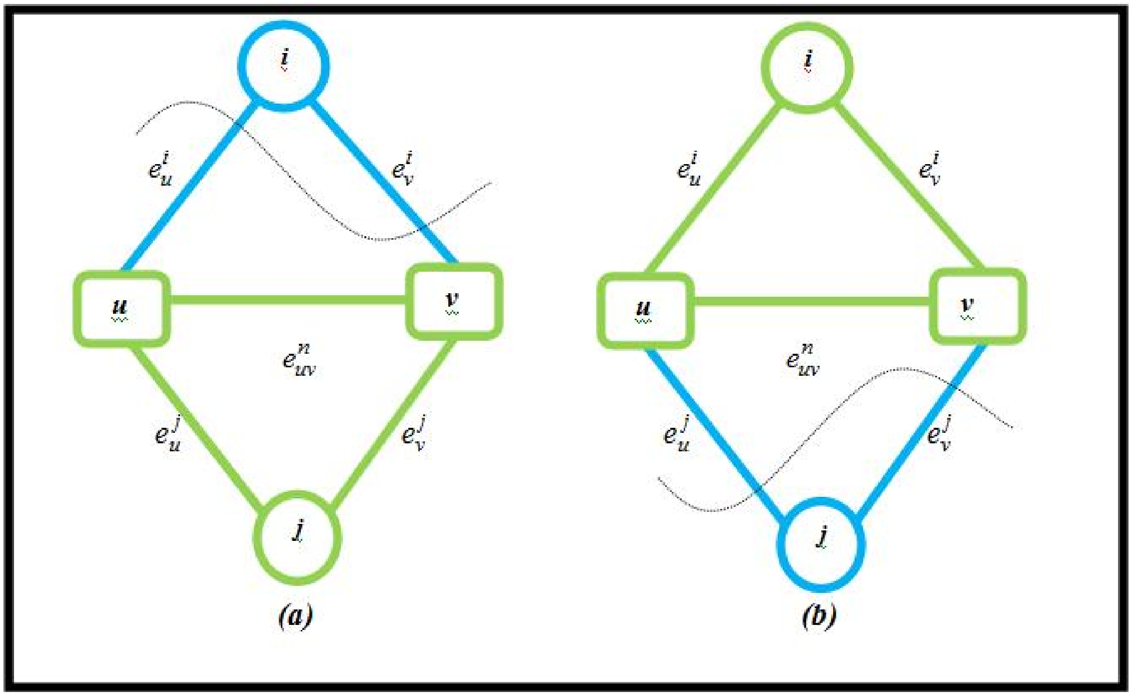

Figure 3.18: For neighboring vertices u and v with $X(u) = X(v)$ *(a) Cut C with* $e_u^i, e_v^i \in C$ *(b) Cut C with* $e_u^j, e_v^j \in C$.

COROLLARY 3.3.2.5

Let u and v be neighbors in V with $X(u) = X(v)$. *For any cut C and for any non-terminal edge* e_{uv}^n,

if the cut C contains t-links corresponding to i for u and corresponding to j for v (i.e. $e_u^i, e_v^j \in C$ *), or corresponding to j for u and corresponding to i for v (i.e.* $e_u^j, e_v^i \in C$*),* e_{uv}^n *must be part of C.*

Proof: Let *C* be a cut with $X(u) = X(v)$ and $e_u^i, e_v^j \in C$. To prove that, e_{uv}^n belongs to *C*, if possible, let's assume that, e_{uv}^n does not belong to *C*. We will show that, *C* is not a cut. Consider a path $i - \left(e_u^i\right) - u - \left(e_{uv}^n\right) - v - \left(e_v^j\right) - j$ connecting *i* and *j*. This path lies in *G* \ *C*. This contradicts with the fact that, *C* is a cut. Thus, e_{uv}^n must belong to *C*.(Refer to Figure 3.19 (b)).

Similarly, we can prove the other case: if *C* is a cut with $X(u) = X(v)$ and $e_u^j, e_v^i \in C$, then e_{uv}^n must belong to *C*. (Refer to Figure 3.19 *(a)*)

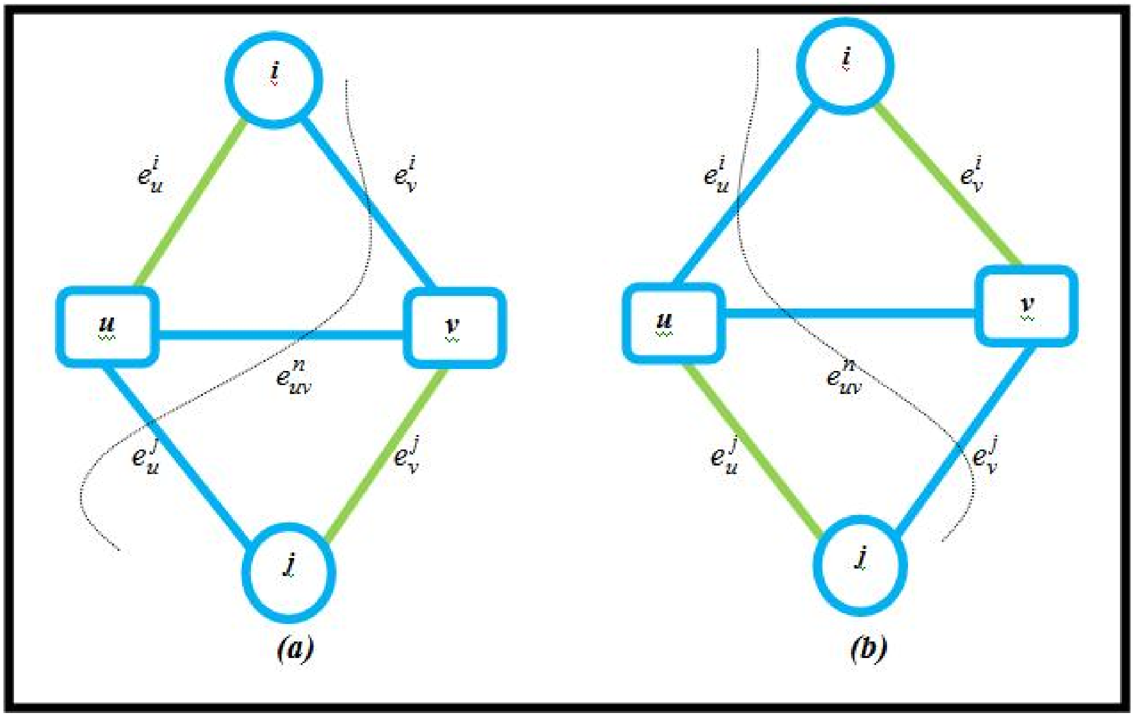

Figure 3.19: For neighboring vertices u and v with $X(u)=X(v)$ (a) Cut C with $e_u^j, e_v^i \in C$ (b) Cut C with $e_u^i, e_v^j \in C$.

THEOREM 3.3.2.6

Let u and v be neighbors in V with $X(u)=X(v)$. The weight of the edges common to both cut C and non-terminal edge e_{uv}^n is $\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))$.

Proof: We want to prove that, the weight of $e_{uv}^n \cap C$ is $\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))$.

There are four cases:

Case *(i)*: $e_u^i, e_v^i \in C$. Then, by Lemma 3.3.2.4, $e_{uv}^n \notin C$, Thus, there is no edge common to e_{uv}^n and C. Thus,

$$\left|e_{uv}^n \cap C\right| = \left|\{\ \}\right| = 0.$$

Also, $\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v)) = \psi_{u,v}(x_i'(x_u), x_i'(x_v))$ ($\because$ Using (3.40))

$$= \psi_{u,v}(x_i'(x_v), x_i'(x_v)) \ (\because X(u)=X(v))$$

$$= c_{uv}\,\mathrm{I}(x_i'(x_v), x_i'(x_v))$$

$$= 0 \ (\because \mathrm{I}(a,a)=0\ \forall a)$$

Thus, $\left|e_{uv}^{n} \cap C\right| = \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v))$.

Case *(ii)*: $e_u^j, e_v^j \in C$ Then, by Lemma 3.2.2.4, $e_{uv}^{n} \notin C$, Thus, there is no edge common to e_{uv}^{n} and C. Thus,

$$\left|e_{uv}^{n} \cap C\right| = \left|\{\ \}\right| = 0.$$

Also, $\psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v)) = \psi_{u,v}(x'_j(x_u), x'_j(x_v))$ ($\because$ Using (3.40))

$$= \psi_{u,v}(x'_j(x_u), x'_j(x_v)) \ (\because X(u) = X(v))$$

$$= c_{uv}\, \mathrm{I}(x'_j(x_u), x'_j(x_v))$$

$$= 0 \ (\because \mathrm{I}(a,a) = 0 \forall a)$$

Thus, $\left|e_{uv}^{n} \cap C\right| = \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v))$.

Case *(iii)*: $e_u^j, e_v^i \in C$. Then, by Corollary 3.2.2.5, $e_{uv}^{n} \in C$. Thus, the edge e_{uv}^{n} is common to both e_{uv}^{n} and C.

$$\left|e_{uv}^{n} \cap C\right| = \left|e_{uv}^{n}\right| = c_{uv}.$$

Also,

$$\psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v)) = \psi_{u,v}(x'_j(x_u), x'_i(x_v)) \left(\because \mathrm{X}_C(u) = x'_j(x_u) \,\mathrm{and}\, \mathrm{X}_C(v) = x'_i(x_v)\right)$$

$$= \psi_{u,v}(x'_j(x_v), x'_i(x_v)) \left(\because X(u) = X(v)\right)$$

$$= c_{uv}\, \mathrm{I}(x'_j(x_u), x'_i(x_v))$$

$$= c_{uv} \ (\because \mathrm{I}(x'_j(x_u), x'_i(x_v)) = 1 \,\mathrm{as}\, x'_j(x_u) \neq x'_i(x_v))$$

Thus, $\left|e_{uv}^{n} \cap C\right| = \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v))$.

Case *(iv)*: $e_u^i, e_v^j \in C$. Then, by Corollary 3.2.2.5, $e_{uv}^{n} \in C$. Thus, the edge e_{uv}^{n} is common to both e_{uv}^{n} and C.

$$\left|e_{uv}^{n} \cap C\right| = \left|e_{uv}^{n}\right| = c_{uv}.$$

Also,

$$\psi_{u,v}(X_C(u), X_C(v)) = \psi_{u,v}(x_i'(x_u), x_j'(x_v))\left(\because X_C(u) = x_i'(x_u) \text{ and } X_C(v) = x_j'(x_v)\right)$$

$$= \psi_{u,v}(x_i'(x_v), x_j'(x_v))\left(\because X(u) = X(v)\right)$$

$$= c_{uv}\, \mathrm{I}(x_i'(x_v), x_j'(x_v))$$

$$= c_{uv}\ (\because \mathrm{I}(x_i'(x_v), x_j'(x_v)) = 1 \text{ as } x_i'(x_v) \neq x_j'(x_v))$$

Thus, $\left|e_{uv}^n \cap C\right| = \psi_{u,v}(X_C(u), X_C(v))$.

This proves the Lemma.

LEMMA 3.3.2.7

Let u and v be neighbors in V with $|X(u) - X(v)| = |k|$.

If $e_{a_{uv}}^t$ *connects supplementary vertex* a_{uv} *with terminal i, there are three edges associated with u, v and* a_{uv}*, i.e.* $e_{ua_{uv}}^n$, $e_{a_{uv}v}^n$ *and* $e_{a_{uv}}^t$. *Then minimum cut C on G satisfies the following:*

(a) If C contains both e_u^i *and* e_v^i*, then, there is only one edge common to both C and* $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\}$*, and it is* $e_{a_{uv}}^t$.

(b) If C contains both e_u^i *and* e_v^j*, then, there is only one edge common to both C and* $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\}$*, and it is* $e_{ua_{uv}}^n$.

(c) If C contains both e_u^j *and* e_v^i*, then, there is only one edge common to both C and* $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\}$*, and it is* $e_{a_{uv}v}^n$.

(d) If C contains both e_u^j *and* e_v^j*, then, there is no edge common to both C and* $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\}$.

Proof: First, let's consider (a). If C is any cut, not necessarily minimum cut and if $e_u^i, e_v^i \in C$, we want to prove that, $e_{a_{uv}}^t \in C$, if it does not contain edges $e_{ua_{uv}}^n$ and $e_{a_{uv}}^t$. (Refer Fig. 3.20 (a)) For that, if possible, assume that, $e_{a_{uv}}^t \notin C$. In this case, C does not contain any edge from $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\}$. It is easy to see that, there exists a path connecting both the terminals via $e_{a_{uv}}^t$, which contradicts with the fact that, C is a cut. Thus, C must contain $e_{a_{uv}}^t$ if it does not contain remaining two edges of $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\}$. It is clear that, there can exist cuts , which contain an edge other than $e_{a_{uv}}^t$ from

$\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$. But in case of such cuts, it must contain at least two edges from $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$ in order to separate both the terminals in the induced graph. It should be noted that, the edge $e^t_{a_{uv}}$ has weight lesser than sum of other two edges of $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$. Hence, if C is a minimum cut, it must contain $e^t_{a_{uv}}$.

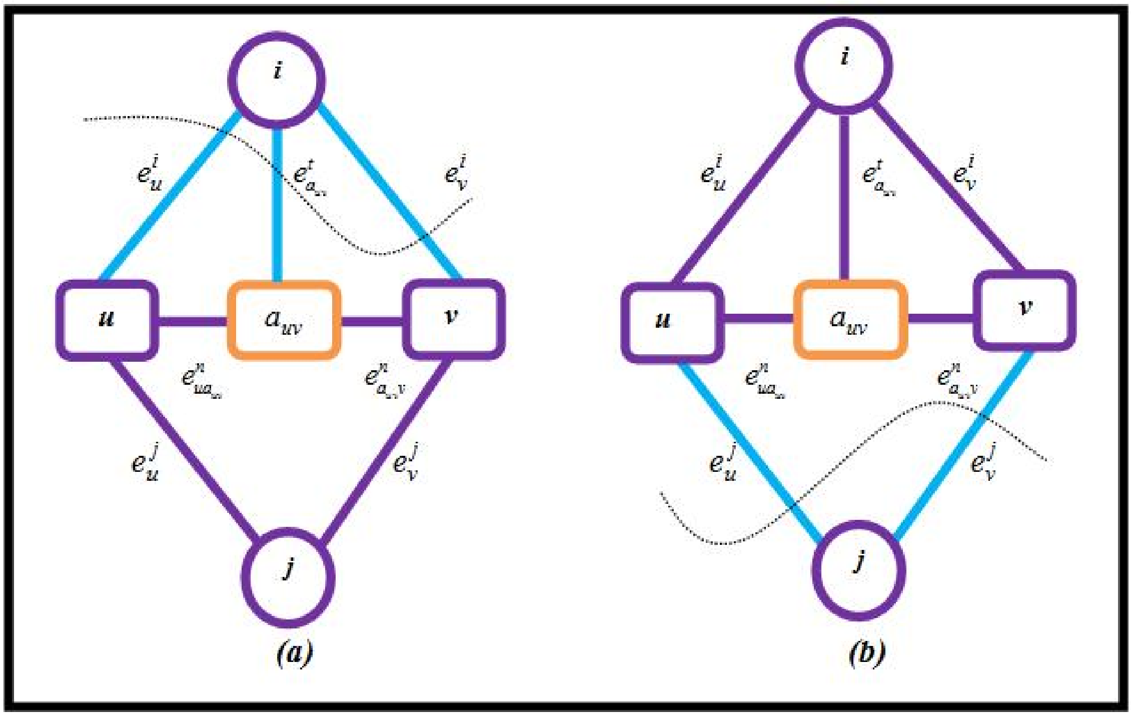

Figure 3.20: For Cut C with with $|X(u)-X(v)|=|k|$ and $e^t_{a_{uv}}$ connects supplementary vertex a_{uv} with terminal i, (a) if e^i_u, $e^i_v \in C$ (b) if $e^j_u, e^j_v \in C$

Let's consider (b). If $e^i_u, e^j_v \in C$, we want to prove that, if C does not contain any other edges from $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$, $e^n_{ua_{uv}} \in C$. (Refer Fig. 3.21 (a)). If possible, assume that, $e^n_{ua_{uv}} \notin C$. Then, there exists a path $i - (e^t_{a_{uv}})$ - a_{uv} - $(e^n_{ua_{uv}}) - u - (e^j_u) - j$ connecting terminals i and j. , which contradicts with the fact that, C is a cut. Thus, C must contain $e^n_{ua_{uv}}$, if it doesn't contain any other edge from $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$. If C contain any edge other than $e^n_{ua_{uv}}$, it must contain two edges from $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$ in order to separate terminals in the induced graph. But cost of sum of such two edges (which is equal to $2.c_{uv}$) is obviously more than that of cutting only $e^n_{ua_{uv}}$ (which is equal to $e^n_{ua_{uv}}$). As C is a minimum cut, it must contain only $e^n_{ua_{uv}}$. This proves (b).

Case (c) is similar to case (b). The proof can be produced with similar argument. (Refer to Figure 3.21(a))

Consider case (d). Given that, e_u^j, $e_v^j \in C$. To prove that, no edge from $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$ belong to C. We need to prove that, C with $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\} \cap C = \phi$ is a cut. For that, observe that, there is no path connecting terminals in G \ C because both the vertices *u* and *v* are disconnected from the terminal *j* (Refer to Figure 3.20 (b)). No other cut will have cost lesser than C. Thus, if C is a minimum cut, it must not contain any edge from the set $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$. This proves the result.

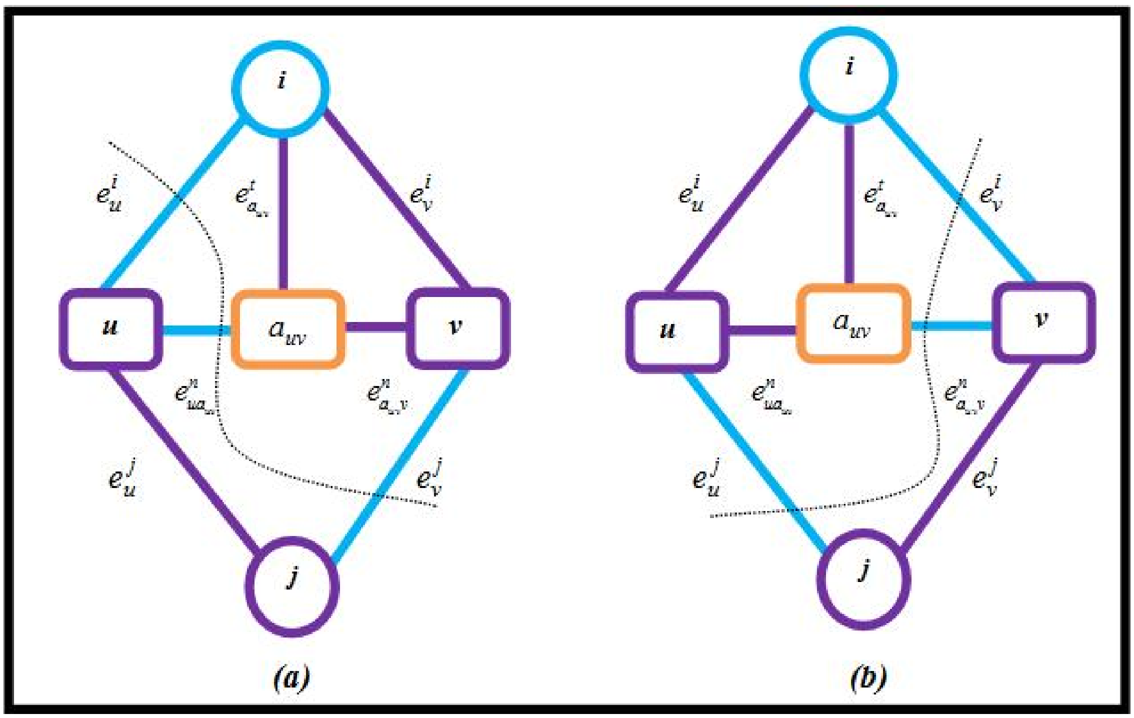

Figure 3.21: For Cut C with with $|X(u)-X(v)|=|k|$ *and* $e^t_{a_{uv}}$ *connects supplementary vertex* a_{uv} *with terminal i, (a) if* $e_u^i, e_v^j \in C$ *(b)if* $e_u^j, e_v^i \in C$

LEMMA 3.3.2.8

Let u and v be neighbors in V with $|X(u)-X(v)|=|k|$.

If $e^t_{a_{uv}}$ *connects supplementary vertex* a_{uv} *with terminal j, there are three edges associated with u, v and* a_{uv}*, i.e.* $e^n_{ua_{uv}}$, $e^n_{a_{uv}v}$ *and* $e^t_{a_{uv}}$. *Then minimum cut C on G satisfies the following:*

(a) If C contains both e_u^i *and* e_v^i*, then, there is no edge common to both C and* $\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\}$.

(b) *If C contains both* e_u^i *and* e_v^j*, then, there is only one edge common to both C and* $\left\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\right\}$*, and it is* $e^n_{a_{uv}v}$.

(c) *If C contains both* e_u^j *and* e_v^i*, then, there is only one edge common to both C and* $\left\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\right\}$*, and it is* $e^n_{ua_{uv}}$.

(d) *If C contains both* e_u^j *and* e_v^j*, then, there is no edge common to both C and* $\left\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\right\}$.

Proof: The result can be proved by arguments similar to that of Lemma 3.3.2.7. (Refer to Figure 3.22 and 3.23)

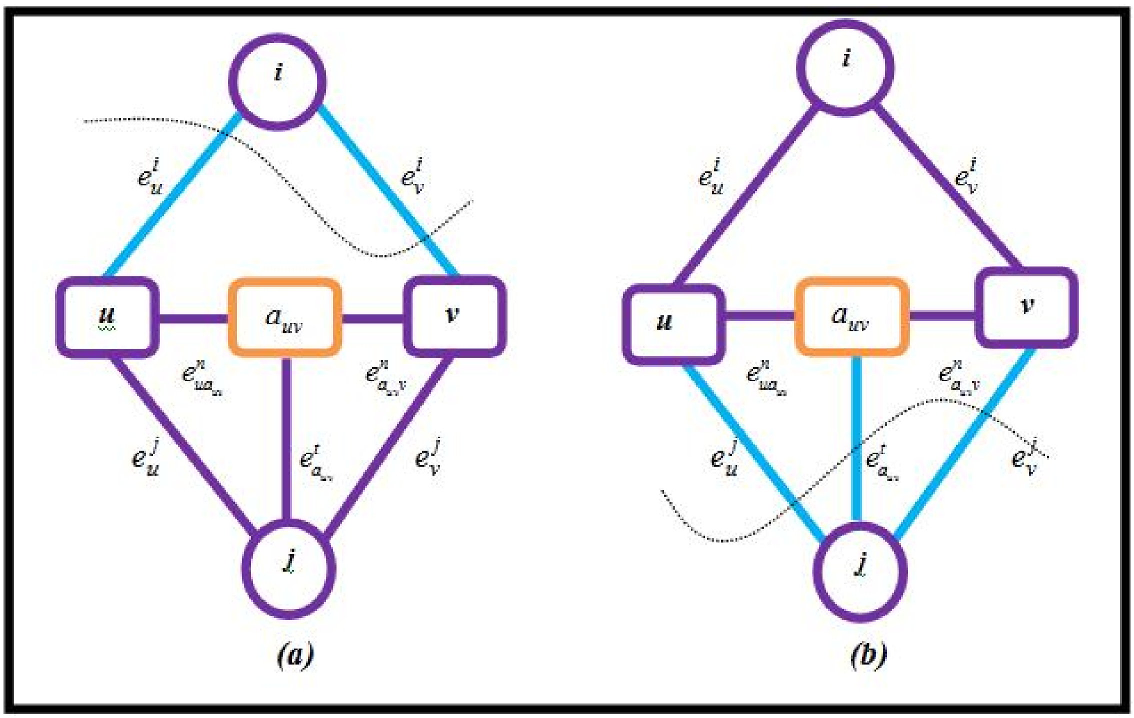

Figure 3.22: For Cut C with with $|X(u) - X(v)| = |k|$ *and* $e^t_{a_{uv}}$ *connects supplementary vertex* a_{uv} *with terminal j, (a) if* $e_u^i,\ e_v^i \in C$ *(b) if* $e_u^j, e_v^j \in C$

Figure 3.22 presents the case when both the terminal edges corresponding to the same terminal vertex are part of the cut.

Figure 3.23 presents the case when terminal edges corresponding to different terminal vertices are part of the cut.

The detailed proof of the result can be easily generated following the proof of Lemma 3.3.2.7 with the help of figures 3.22 and 3.23.

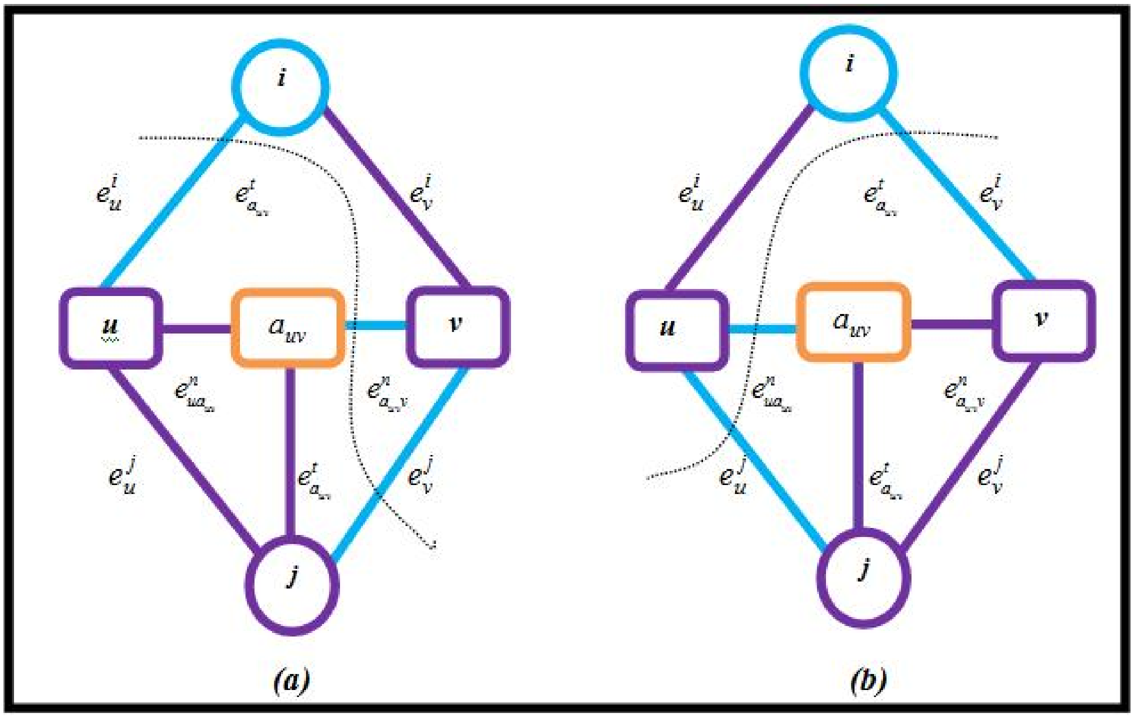

Figure 3.23: For Cut C with with $|X(u)-X(v)|=|k|$ *and* $e_{a_{uv}}^t$ *connects supplementary vertex* a_{uv} *with terminal j, (a) if* e_u^i, $e_v^j \in C$ *(b) if* e_u^j, $e_v^i \in C$

THEOREM 3.3.2.9

Let u and v be neighbors in V with $|X(u)-X(v)|=|k|$. *If C is minimum cut on G, the total weight of edges common to both* $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\}$ *and C will be* $\psi_{u,v}(X_C(u), X_C(v))$.

Proof: We want to prove that, the weight of $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\} \cap C$ is $\psi_{u,v}(X_C(u), X_C(v))$.

Without loss of generality, we assume that, the terminal edge $e_{a_{uv}}^t$ connects the dummy vertex a_{uv} with terminal *i*.

There are four cases:

Case *(i)*: $e_u^i, e_v^i \in C$. Then, by Lemma 3.3.2.7, $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\} \cap C = e_{a_{uv}}^t$. Thus,

$$\left|\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\} \cap C\right| = \left|e_{a_{uv}}^t\right| = c_{uv}.$$

Also, $\psi_{u,v}(X_C(u), X_C(v)) = \psi_{u,v}(x_i'(x_u), x_i'(x_v))$ ($\because$ Using (3.40))

$$= c_{uv} \quad (\because X(u) \neq X(v))$$

Thus, $\left|e_{uv}^{n} \cap C\right| = \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v))$.

Case *(ii)*: $e_u^i, e_v^j \in C$ Then, by Lemma 3.3.2.7, $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\} \cap C = e_{ua_{uv}}^n$. Thus, $\left|\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\} \cap C\right| = \left|e_{ua_{uv}}^n\right| = c_{uv}$.

Also, $\psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v)) = \psi_{u,v}(x_i'(x_u), x_j'(x_v))$ ($\because$ Using (3.40))

$$= c_{uv} \quad (\because X(u) \neq X(v))$$

Thus, $\left|e_{uv}^{n} \cap C\right| = \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v))$.

Case *(iii)*: $e_u^j, e_v^i \in C$. Then, by Lemma 3.3.2.7, $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\} \cap C = e_{a_{uv}v}^n$. Thus, $\left|e_{uv}^n \cap C\right| = \left|e_{uv}^n\right| = c_{uv}$.

Also,

$$\psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v)) = \psi_{u,v}(x_j'(x_u), x_i'(x_v)) \left(\because \mathrm{X}_C(u) = x_j'(x_u) \,\text{and}\, \mathrm{X}_C(v) = x_i'(x_v)\right)$$

$$= c_{uv}\, \mathrm{I}(x_j'(x_u), x_i'(x_v))$$

$$= c_{uv} \ (\because \mathrm{I}(x_j'(x_u), x_i'(x_v)) = 1 \text{ as } x_j'(x_u) \neq x_i'(x_v))$$

Thus, $\left|e_{uv}^{n} \cap C\right| = \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v))$.

Case *(iv)*: $e_u^j, e_v^j \in C$. Then by Lemma 3.3.2.7, $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\} \cap C = \phi$. Thus, there is no edge common to both $\left\{e_{ua_{uv}}^n, e_{a_{uv}v}^n, e_{a_{uv}}^t\right\}$ and C.

$\left|e_{uv}^n \cap C\right| = |\phi| = 0$.

Also,

$$\psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v)) = \psi_{u,v}(x_j'(x_u), x_j'(x_v)) \left(\because \mathrm{X}_C(u) = x_i'(x_u) \,\text{and}\, \mathrm{X}_C(v) = x_j'(x_v)\right)$$

$$= c_{uv}\, \mathrm{I}(x_j'(x_v), x_j'(x_v))$$

$$= 0 \ (\because \mathrm{I}(x_i'(x_v), x_j'(x_v)) = 0 \text{ as } x_i'(x_v) = x_j'(x_v))$$

Thus, $\left|e_{uv}^{n} \cap C\right| = \psi_{u,v}(\mathrm{X}_C(u), \mathrm{X}_C(v))$.

This proves the Theorem.

Note that, minimum cut C on G satisfies both Lemma 3.3.2.7 and Lemma 3.3.2.8. However, there are cuts, which satisfy both the results without being minimum cut.

DEFINITION

A cut C on graph G corresponding to k – shift move is said to be a ***basic cut*** *(or* ***b – cut****)*, if it satisfies Lemma 3.3.2.7 and Lemma 3.3.2.8.

THEOREM 3.3.2.10

The set of all b – cuts on the network flow G (corresponding to k – shift move) and the set of all labeling those are one k – shift move far from initial labeling X are in one to one correspondence.

Proof: From Lemma 3.3.2.3, it follows that, if X_C is a labeling corresponding to b – cut C, then it is one k – shift move away from X.

To prove the other part, let's consider a labeling X', which is one k – shift move far from the initial labeling X. We want to prove that, there is a b –cut C, which corresponds to labeling X_C, which is same as X'. As X' is single k – shift move far from the initial labeling X, it will be of the form,

$X'(v) = X(v)$ or $X'(v) = X(v) + k$.

Define C as follows:

$e_v^i \in C$ if $X'(v) = X(v) + k$ and $X(v) + k \in S_i$

$e_v^j \in C$ if $X'(v) = X(v) + k$ and $X(v) + k \in S_j$

$e_v^i \in C$ if $X'(v) = X(v)$ and $X(v) \in S_i$

$e_v^j \in C$ if $X'(v) = X(v)$ and $X(v) \in S_j$

$e_{uv}^n \in C$, if $X(u) = X(v)$ and either $e_u^i, e_v^j \in C$ or $e_u^j, e_v^i \in C$

If $|X(u) - X(v)| = |k|$ and $e_{a_{uv}}^t$ connects a_{uv} with the terminal i, then

$e_{a_{uv}}^t \in C$, if e_u^i, $e_v^i \in C$

$e_{ua_{uv}}^n \in C$, if $e_u^i, e_v^j \in C$

$e_{a_{uv}v}^n \in C$, if $e_u^j, e_v^i \in C$

If $|X(u)-X(v)|=|k|$ and $e^t_{a_{uv}}$ connects a_{uv} with the terminal j, then

$e^t_{a_{uv}} \in C$, if e^j_u, $e^j_v \in C$

$e^n_{ua_{uv}} \in C$, if $e^j_u, e^i_v \in C$

$e^n_{a_{uv}v} \in C$, if $e^i_u, e^j_v \in C$

If is clear from the construction of C, that it is a b – cut and X_C is same as X'.

This proves the theorem.

THEOREM 3.3.2.11

If C is a b – cut on graph G corresponding to k – shift move, the difference between cost of C and $O(X_C)$ is constant.

Proof: Basic cut contains three types of links namely (i) terminal edges (i.e e^i_v and e^j_v), (ii) non-terminal edges for neighboring pixels u and v with $X(u)=X(v)$ (i.e. e^n_{uv}) (iii) edges from $\left\{e^n_{ua_{uv}}, e^n_{a_{uv}v}, e^t_{a_{uv}}\right\}$.

Hence, $$|C|=\sum_{v\in V}\left|C\cap\{e^i_v,e^j_v\}\right|+\sum_{\substack{\{u,v\}\in N\\ X(u)=X(v)}}\left|C\cap e^n_{uv}\right|+\sum_{\substack{\{u,v\}\in N\\ |X(u)-X(v)|=|k|\\ e\in\left\{e^n_{ua_{uv}},e^n_{a_{uv}v},e^t_{a_{uv}}\right\}}}\left|C\cap e\right|. \quad (3.41)$$

For $v\in V$, the first term $\left|C\cap\{e^i_v,e^j_v\}\right|$ takes care of weights of terminal edges. There are two cases:

If C contains e^i_v, the first term would be $\left|e^i_v\right|$, which would be $\varphi_v(x'_i(x_v))=\varphi_v(X_C(v))$.

If C contains e^j_v, the first term would be $\left|e^j_v\right|$, which would be $\varphi_v(x'_j(x_v))=\varphi_v(X_C(v))$

Hence, in both the cases,

$$\left|C\cap\{e^i_v,e^j_v\}\right|=\varphi_v(X_C)$$

Hence, $$\sum_{v\in V}\left|C\cap\{e^i_v,e^j_v\}\right|=\sum_{v\in V}\varphi_v(X_C) \quad (3.42)$$

The second term of (3.41) involves weights of all non-terminal edges for neighboring pixels u and v with $X(u)=X(v)$ contained in the cut. From Theorem 3.3.2.6,

$$\left|e_{uv}^{n}\cap C\right|=\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))$$

Hence, $\sum_{\{u,v\}\in N}\left|e_{uv}^{n}\cap C\right|=\sum_{\substack{\{u,v\}\in N\\ X(u)=X(v)}}\left|e_{uv}^{n}\cap C\right|$

$$=\sum_{\substack{\{u,v\}\in N\\ X(u)=X(v)}}\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v)) \tag{3.43}$$

The third term of (3.41) involves weights of all edges e of $\left\{e_{ua_{uv}}^{n},e_{a_{uv}v}^{n},e_{a_{uv}}^{t}\right\}$ related to neighbor u and v with $\left|X(u)-X(v)\right|=\left|k\right|$. But, from lemma 4.6,

$$\left|C\cap e\right|=\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))$$

Thus, $\sum_{e\in\left\{e_{ua_{uv}}^{n},e_{a_{uv}v}^{n},e_{a_{uv}}^{t}\right\}}\left|C\cap e\right|=\sum_{\substack{\{u,v\}\in N\\ |X(u)-X(v)|=|k|}}\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))$ (3.44)

Using (3.42) to (3.44) in (3.41), the total cost of a cut C will be,

$$\left|C\right|=\sum_{v\in V}\varphi_v(X_C)+\sum_{\substack{\{u,v\}\in N\\ X(u)=X(v)}}\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))+\sum_{\substack{\{u,v\}\in N\\ |X(u)-X(v)|=|k|}}\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))$$

$$=\left(\sum_{v\in V}\varphi_v(X_C)+\sum_{\{u,v\}\in N}\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))\right)-\left(\sum_{\substack{\{u,v\}\in N\\ |X(u)-X(v)|\neq|k|}}\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))\right)$$

$$=\left(O\left(X_C\right)\right)+\left(-\sum_{\substack{\{u,v\}\in N\\ |X(u)-X(v)|\neq|k|}}\psi_{u,v}(\mathrm{X}_C(u),\mathrm{X}_C(v))\right)$$

$$=\left(O\left(X_C\right)\right)+\left(-\sum_{\substack{\{u,v\}\in N\\ |X(u)-X(v)|\neq|k|}}c_{uv}\right)(\because\psi_{u,v}(\alpha,\beta)=c_{uv}\text{ if }\alpha\neq\beta\ )$$

$$=\left(O\left(X_C\right)\right)+K,$$

where, $K = \left(- \sum_{\substack{\{u,v\} \in N \\ |X(u)-X(v)| \neq |k|}} c_{uv} \right)$ is a constant.

This means, $|C| - (O(X_C)) = K$, which proves the result.

COROLLARY 3.3.2.12

Let C be a minimum cut on graph G corresponding to k – shift move. Then, X_C *minimizes the objective function (3.27) over the space of all labeling those are single k – shift move away from the initial labeling.*

Proof: By Theorem 3.3.2.11, $|C| - (O(X_C)) = K$.

i.e. $(O(X_C)) = |C| + K$. (3.44)

By Theorem 3.3.2.10, the set of all b – cuts on G are in one to one correspondence with set of all labeling those are single k – shift move far from the initial labeling. Hence, it is enough to prove that, X_C minimizes the objective functions over all possible labeling corresponding to b – cut on G.

As $|C|$ is minimum for minimum cut C over all cuts C on G, the result follows from (3.44).