Day 1

Discrete Optimization*

1.1 What is Discrete Optimization

Discrete optimization is a way of finding the best solution out of finite number of possibilities in a computationally efficient way.

1.1.1 Pseudo Boolean function

Pseudo Boolean function is function whose domain is Boolean and then range is Real.
$f : \{0, 1\} \to R$
$x_1, x_2 \in \{0, 1\}$
$f(x) = 3x_1 + 4\bar{x}_1 + 5x_2 + 3\bar{x}_2 + 6\bar{x}_1, x_2$
when we randomly put values of $x_1 = 1, x_2 = 1$
then $f = 3 \times 1 + 4 \times 0 + 5 \times 1 + 6 \times 0 + 10 \times 0 = 8$
so values of f(x)=8 is might not be the minimum value of function. Lets see how to find minimum value of Pseudo Boolean function $f(x)$.
Example 2. $f(x) = 3x_1 + 4\bar{x}_1 + 5x_2 + 3\bar{x}_2 + 6\bar{x}_1, x_2$
Now task is to minimize $f(x)$
Solution:
$f : \{0, 1\} \to R$
$x_1, x_2 \in \{0, 1\}$
Goal min $f(x)$

1.1.2 Minimization of Pseudo Boolean function

So lets see minimization of these type of function graph cut.
for this we first convert pseudo-Boolean function into equivalent flow network, then find min cut of the graph which give optimal assignment for variable x_1, x_2.

*Lecturer: Dr.Anand Mishra. Scribe: Neelu Verma.

Method 1: Unconstrained Optimization

So this is unconstrained Optimization used for assignment of each of these x_1, x_2.
we can put one by one all possible combination of x_1 and x_2 and then find $f(x)$, the min value of $f(x)$ is the optimum value
1. $x_1 = 0, x_2 = 0$ $f(x) = 3 \times 0 + 4 \times 1 + 5 \times 0 + 3 \times 1 + 6 \times 1 \times 0$
$f(x) = 7$

2. $x_1 = 0, x_2 = 1$
$f(x) = 3 \times 0 + 4 \times 1 + 5 \times 1 + 3 \times 0 + 6 \times 1 \times 1$
$f(x) = 15$

3. $x_1 = 1, x_2 = 0$
$f(x) = 3 \times 1 + 4 \times 0 + 5 \times 0 + 3 \times 1 + 6 \times 0 \times 0$
$f(x) = 6$

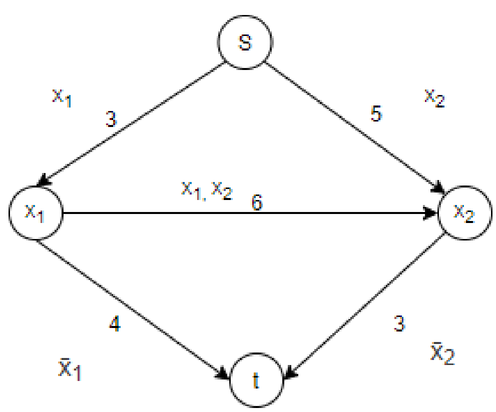
4. $x_1 = 1, x_2 = 1$
$f(x) = 3 \times 1 + 4 \times 0 + 5 \times 1 + 3 \times 0 + 6 \times 0 \times 1$
$f(x) = 8$
So the optimum or minimum value of $f(x)$ is 6 and optimal assignment of x_1 and x_2 is 1, 0.

Method 2: Graph cut

step 1. graph construction
Graph construction as follows:

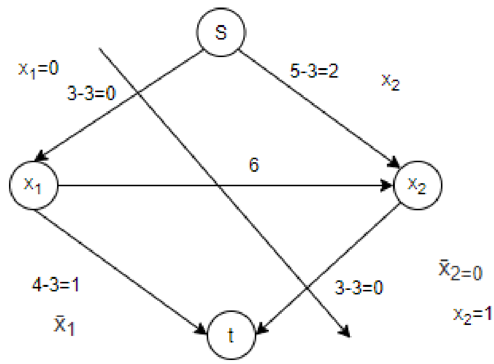
1. Every cut of that graph corresponds to some assignment to variables
2. Min cut = minimum cost assignment

for every variable there will be a node and take 2 additional node called Source and Sink and then make connections



(Note) $\rightarrow$ No edges will go to sink to source side.

Step 2. Finding Min cut: - once you have graph then our task is to find the assignment of x_1 and x_2 by min cut.
there is some ways of choose path for example if I choose $S - x_2 - t$ path then flow will be 3 because 3 is the minimum and x_2 to t path will be exhausted and S to x_2 become 5-3 =2 in some way in path $S - x_2 - t$ flow will be 3 because S to x, flow is 3 then S to x_1 path exhausted and x_1 to t path become 1.



So $f(x) =$ min cut =6 (3+3) $x_1 = 1, x_2 = 0$

1.2 Flow

Let $G = (V, E)$ be a flow network with a capacity function c, let s be the source of the network and t be the sink.
Then a flow in G is defined as a real valued function
$f : V \times V \rightarrow R$ that satisfies following three properties:

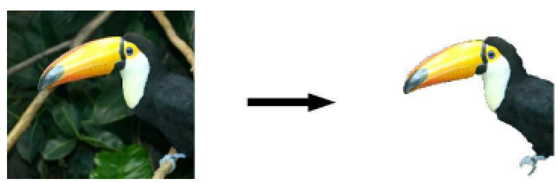
1. Capacity constraint: $f(u, v) \leq c(u, v)$; $\forall u, v \in V$
2. Skew Symmetry: $f(u, v) = -f(v, u)$; $\forall u, v \in V$
3. Flow Conservation: $\sum_{u \in V} f(u, v) = 0$; $\forall u \in V - \{s, t\}$
   or equivalently
   $\sum_{v \in V} f(v, u) = 0$; $\forall u \in V - \{s, t\}$

Day 2

Applications of graph cut[*]

2.1 Image segmentation problem as Binary label cases

Example: The segmentation could be anything. Here I am taking example of bird segmentation from the image Image of bird



The energy we need to minimize is of given:
$E(X) = \sum_i C_i X_i + \sum_{i,j} C_{i,j}\ X_i(1 - X_j)$
Global minima gives correct value of which pixel to take or which pixel not to take
we wish to find out the global minimal of that function
$x* = argmin_x E(x)$

250	240	255
5	230	9
6	235	10

$\longrightarrow$

1	1	1
0	1	0
0	1	0

lets have 3 * 3 matrix and some pixel value associated with each of the pixel.
The goal is to segment out

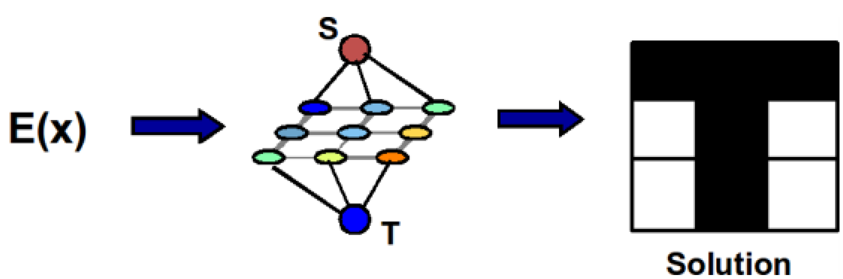
[*]Lecturer: Dr.Anand Mishra. Scribe: Neelu Verma.

Image is scale of [0,255], variable are x_1 to x_9 each have some cost.
so something closer to 0 give 0
and something closer to 255 give 1
cost of assigning x_1 to 1
$(255 - p(x_1))x_1$
$(255 - p(x_i))x_1 + p(x_1).\bar{x}_1$
so cost is for single pixel or single assignment If we want consistency in assignment
so lets x_i and x_j are neighbour and if they take any label then.
if some pixel are object then its neighbour is also a object.
for Example we have mouse in image
if $x_i \rightarrow 0$
$x_j \rightarrow 1$
then they will never be cost.
$\sum_{i,j \in N} C_{i,j}\ x_i x_j + C_{j,i}, x_j, \bar{x}_i$
once have this formulation then we have assignment.So that foreground will be segmented.

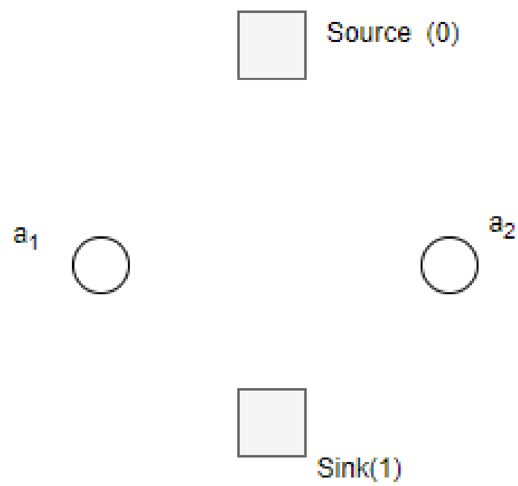
Graph Construction

Construct a graph such that:

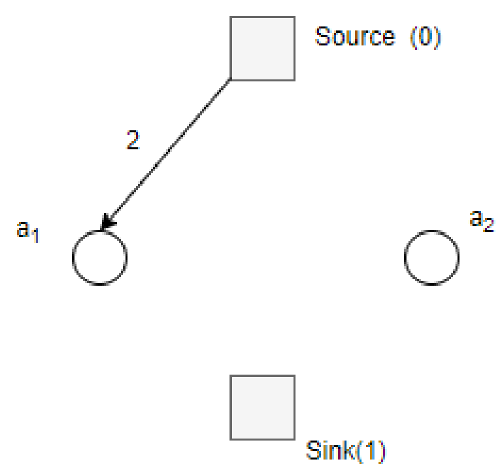
1. Any cut corresponds to an assignment of x
2. The cost of the cut is equal to energy of x : $E(x)$



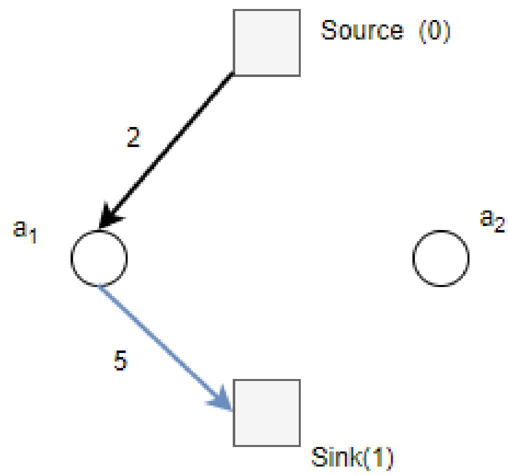
Let us understand how a graph is constructed for a given energy: (for simplicity let us assume there are two pixels a_1 and a_2 $E(a_1, a_2) =$



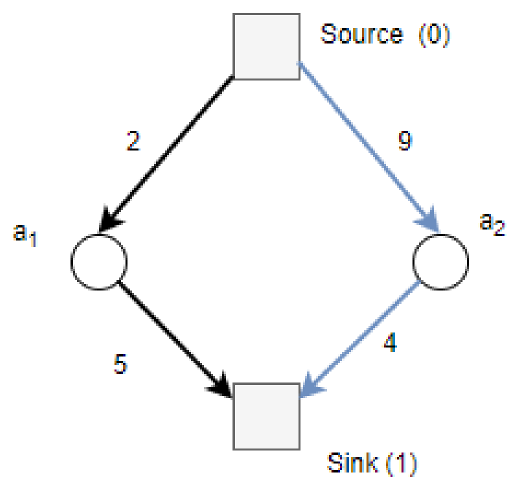
$E(a_1, a_2) = 2a_1$



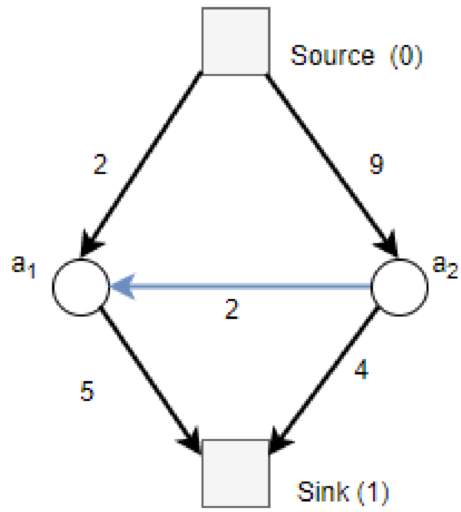
$E(a_1, a_2) = 2a_1 + 5\bar{a}_1$



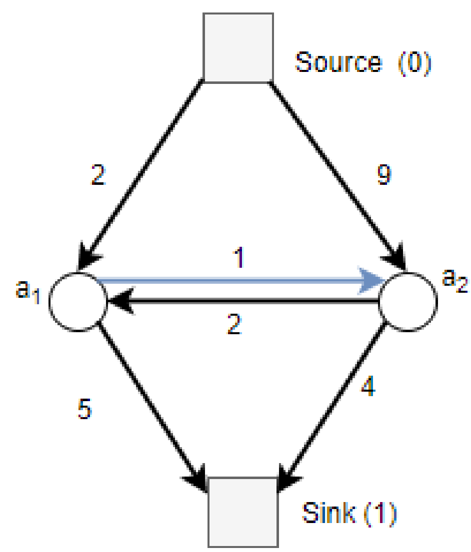
$E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2$



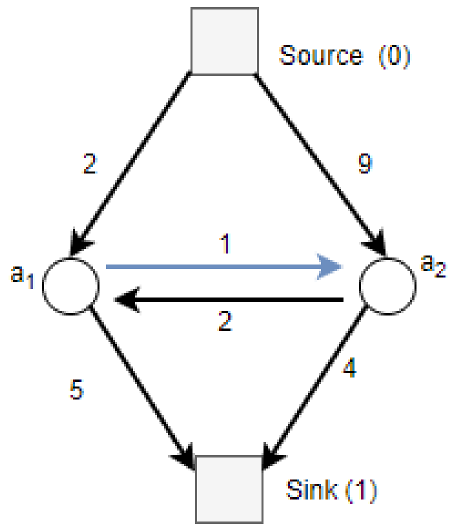
$E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2$



$E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$



$E(a_1, a_2) = 2a_1 + 5\bar{a}_1 + 9a_2 + 4\bar{a}_2 + 2a_1\bar{a}_2 + \bar{a}_1 a_2$



What is a sub-modular function

Let f be a function defined over set of boolean variables $x = \{x_1, x_2, ..., x_n\}$, then

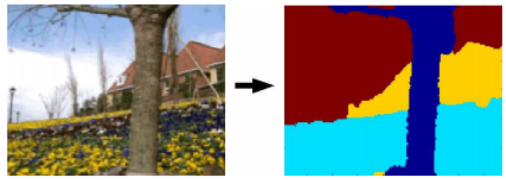
1. 1 All functions of one boolean variables are sub modular.
2. A function f of two boolean variable is sub-modular if
   $f(0,0) + f(1,1) \leq f(0,1) + f(1,0)$.
3. In general a function is sub-modular if all its projection to two variables are sub-modular.

2.2 Application of Graph cut as multi label cases

You might be wondering I am till now only talking about bi-label problems where a pixel can take either source (foreground) or sink(background).
But in practice there are many vision problems which can be posed in a multi labelling framework.

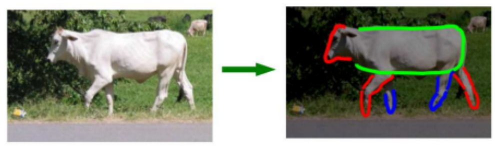
Example 1:

Find the labels for tree, sky, house and ground.



Example 2:

I am interested in finding parts of an object.

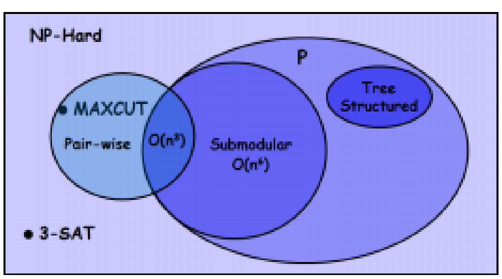


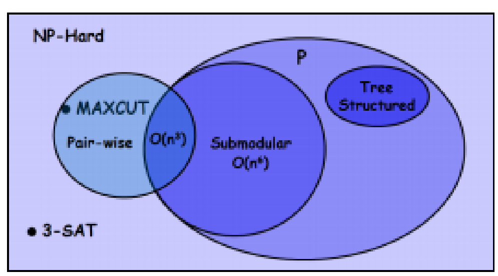
More examples:

1. Stereo correspondence
2. Image de-noising
3. Image In-painting

What energy functions can be minimized?

General energy functions: NP hard to minimize, only approximate solutions available
Easy energy functions (sub-modular functions) are graph re-presentable and solvable in polynomial time.





Same as bi-label cases, the goal is to find a labelling that assigns each pixel $p \in P$ a label $f_p \in L$ where f is both consistent with observed data and piecewise smooth. i.e. we wish to find the global minima of energy function of following form:

$$E(f) = E_{data}(f) + E_{smooth}(f)$$

The form of $E_{data}(f)$ is typically, $E_{data}(f) = \sum_{p \in P} D_p(f_p)$ Where D_p measures the agreement

of inferred label from the observed data.
In image restoration, for example, $D_p(f_p) is (f_p - i_p)^2$ where i_p is the observed intensity at pixel p.

$E(f) = E_{data}(f) + E_{smooth}(f)$
$E_{smooth}(f)$ hassymbol following form typically, $E_{smooth}(f) = \sum_{(p,q)\in N} V_{pq}(f_p, f_q)$
The smoothness term E_{smooth} is used to impose spatial smoothness. It should have discontinuity preserving property.

2.3 Question:

Can we minimize this form of energy always using graph cut?
We have seen for $|L| = 2$, this energy is exactly minimization.
We can also proof that we can find global minima of such energy in case when $|L|$ =finite but $V_{pq}(f_p, f_q) = |f_p - f_q|$.
Unfortunately, such V_{pq} is not discontinuity preserving and thus can not be applied for many vision application. In general, minimizing such energy function is an NP hard problem.
Although we can get approximate solution with known factor of global minima in case when V_{pq} is either a metric or semi-metric.

2.4 Semi-Metric and Metric on space of labels

A function $V(.,.)$ is called a semi-metric on the space of labels
$\alpha, \beta \in L$ if it satisfies following two properties:

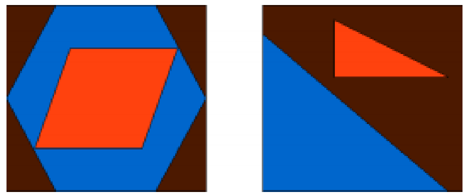
1. $V(\alpha, \beta) = V(\beta, \alpha)$
2. $V(\alpha, \beta) = 0 \Leftrightarrow \alpha = \beta$

If $V(,)$ also satisfies triangle inequality i.e.
$V(\alpha, \beta) = V(\alpha, \gamma) + V(\gamma, \beta)$
for $\alpha, \beta, \gamma \in L$ then it is a metric.
Example: function $V_{pq} = min(K, |f_P - f_q|^2)$ is a semi-metric whereas Potts model $V = \delta(f_p - f_q)$ is a metric.

2.5 Move algorithm for approximate solution

Now we will describe two move algorithms which find out the approximate solution for the energy function.
First move algorithm is known as $\alpha - \beta$ swap which works when V is a semi-metric.
Second move algorithm we call as α expansion, it works only when V is metric.

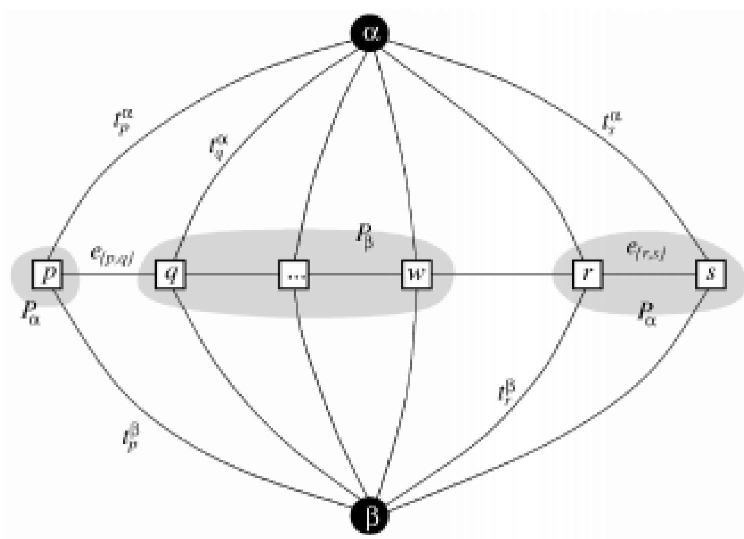
Before we formally define these moves let us first understand how labelling is associated with partitioning of pixels. Any labelling f can be uniquely represented by a partition of image pixels $P = P_l : l \in L$ where P_l is subset of pixels assigned label l.



Here two different labelling (or equivalently partitioning) is shown.

2.6 $\alpha - \beta$ swap

Given a pair of labels α, β a move from partition P to partition P' is called an $\alpha - \beta$ swap if $P_l = P'_l$ for any label $l \neq \alpha, \beta$



2.7 α expansion

Given a label α a move from partition P to partition P' is called an α expansion if $P_\alpha \subset P'_\alpha$ and $P'_l \subset P_l$ for any label $l \neq \alpha$.

2.8 $\alpha - \beta$ swap: basic algorithm

1 Start with an arbitrary labelling f
2 Set success $\rightarrow 0$
3 for each pair of label $\{\alpha, \beta\} \in L$
1 Find $\hat{f} = argmin_{f'} E(f')$ within one $\alpha - \beta$ swap of f
2 If $E(\hat{f}) < E(f)$ then
Set $f \leftarrow \hat{f}$
Set success $\leftarrow 1$
4 if success == 1 then go to step 2 else return f.

2.9 α expansion: basic algorithm

1 Start with an arbitrary labelling f
2 Set success $\leftarrow 0$
3 for every label $\alpha \in L$

1. Find $\hat{f} = argmin_{f'} E(f')$ within one α expansion of f
2. If $E(\hat{f}) < E(f)$ then
   Set $f \leftarrow \hat{f}$
   Set success $\leftarrow 1$

4 if success == 1 then go to step 2 else return f.

Important properties of swap and expansion algorithm

1. A cycle in a swap move algorithm takes $O(|L|2)$ iterations, and a cycle in expansion move takes $O(|L|)$ iterations.
2. The algorithms are guaranteed to terminate in finite number of cycles.
3. Expansion move produces a labeling f such that $E(f^*) \leq E(f) \leq 2k.E(f)$ where f is the global minimum and constant
   $k = \frac{max\{V(\alpha,\beta):\alpha \neq \beta\}}{min\{V(\alpha,\beta):\alpha \neq \beta\}}$

2.10 $\alpha - \beta$ swap: graph cuts

Goal: Given an input labelling f(partition P) and a pair of labels α, β, we wish to find a labelling $\hat{f}$ that minimizes E over all labelling within one $\alpha - \beta$ swap of f. This swap move can be performed using graph cut. Let us understand it by an example. Let us suppose there are two partitions of pixels one with labels $\alpha(P\alpha)$ and the other one with labels $\beta(P\beta)$. Suppose $P_\alpha = \{p, q, r\}$

$P_\beta = \{s, t, ..., w\}$

The graph is constructed to find out the $\alpha - \beta$ swap. The source and target of the graph are α and β respectively. The cut of the graph determines which pixels will change their labels and which others will retain it.

2.11 α Expansion: graph cuts

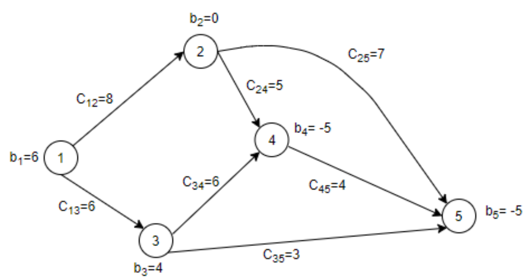
In the similar way α expansion is performed using graph cut. For this a graph is constructed with source and target as α and non-α respectively, and graph cut is performed for each label in L.

Day 3

Minimum cost flow network*

So problem are as follows:-

There are some supply nodes say (1, 3) and (4, 5) are demand nodes and 2 is an intermediate node which are neither doing supply nor doing demand.
There are some unit of thing these nodes are producing,consuming or neither producing nor consuming.



c is the cost of i to j node.

Problem:

Find out minimum cost flow for transporting all the material from supply nodes to demand nodes.
Solution: Optimization formulation
min $\sum_i \sum_j c_{ij}x_{ij}$ suggested to

$$x_{12} + x_{13} = b_1$$
$$-x_{12} + x_{24} + x_{25} = b_2$$
$$-x_{13} + x_{34} + x_{35} = b_3$$

*Lecturer: Dr.Anand Mishra. Scribe: Neelu Verma.

$$-x_{34} - x_{24} + x_{45} = b_4$$
$$-x_{25} - x_{45} - x_{35} = b_5$$

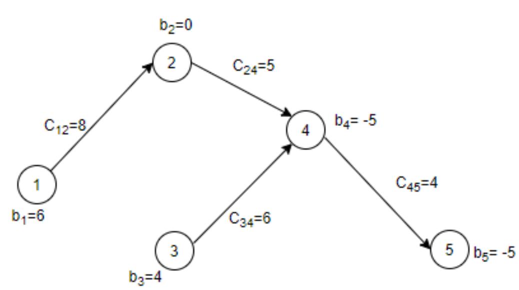
primal dual formulation.
Optimization formulation-

$$\max \sum_j w_j b_j$$

suggeated to
$w_i - w_j \leq c_{ij}$
use some auxiliary variable for each of these constraints and sum it up.
Maximization in primal = Maximization problem in in dual.
multiplying by w_1, w_2, w_3, w_4, w_5 corresponding and sum of all 5 equation.

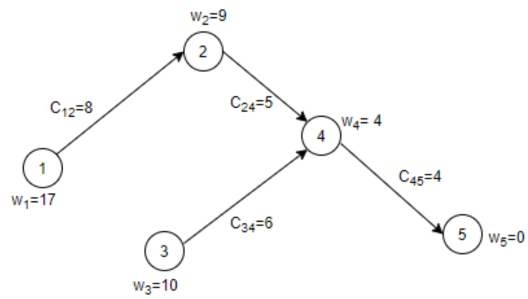
$$x_{12} + x_{13} = b_1 \times w_1$$
$$-x_{12} + x_{24} + x_{25} = b_2 \times w_2$$
$$-x_{13} + x_{34} + x_{35} = b_3 \times w_3$$
$$-x_{34} - x_{24} + x_{45} = b_4 \times w_4$$
$$-x_{25} - x_{45} - x_{35} = b_5 \times w_5$$
$$(w_1 - w_2)x_{12} + (w_1 - w_3)x_{13} + (w_2 - w_4)x_{24}..... = \sum_j w_j b_j$$

$\sum(w_i - w_j)x_{ij} = \sum_j w_j b_j$
Now i put a constraints that is $w_i - w_j \leq c_{ij}$
that is equivalently write $\sum\sum(w_i - w_j) \leq \sum\sum c_{ij}x_{ij}$
Now we want to minimize it $\sum\sum c_{ij}x_{ij}$
which is equivalent to maximize (Dual problem) $\max \sum_j w_j b_j$
suggested to $w_i - w_j \leq c_{ij}$ so cost can be calculated as:
$6 \times 8 + 6 \times 5 + 4 \times 6 + 5 \times 4 = 122$

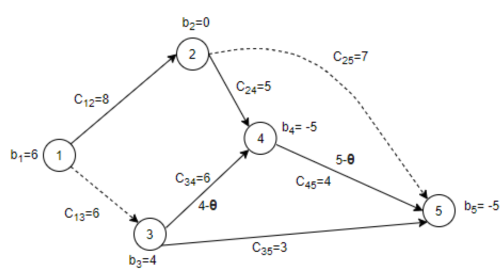


Method: Network simplex method

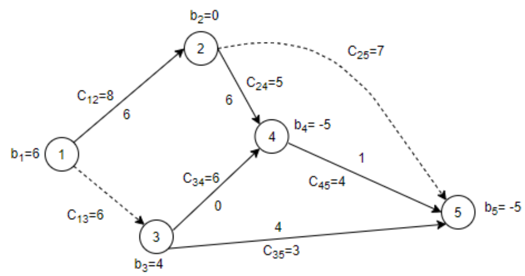
Network simplex method for n node we get $n - 1$ edges and find feasible solution



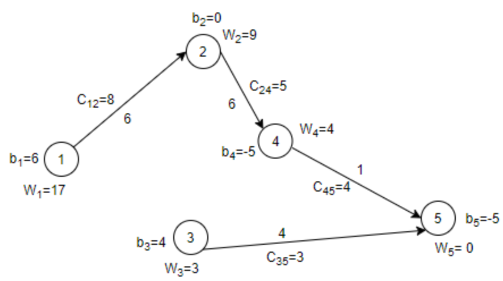
Now invite $w_5 = 0$ and in reverse order compute $w_1, w_2, ...$
using $w_i, w_j = C_{ij}$ or $w_i = c_{ij} + w_j$
$w_2 = c_{24} + w_4 = 5 + 4 = 9$
$w_1 = c_{12} + w_2 = 8 + 9 = 17$
$w_3 = c_{34} + w_4 = 6 + 4 = 10$
$w_4 = 4 + 0 = 4,$
These are basic solution and given edges are basic edges
for non - basic edges verify if
$w_i - w_j - c_{ij}$ for equation (1, 3) $w_1 - w_3 - c_{13}$
17 - 10 - 6 = 1
for equation (2, 5) $w_2 - w_5 - c_{25}$
9 - 0 - 7 = 2
for equation (3, 5) $w_3 - w_5 - c_{35}$
10 - 0 - 3 = 7
—(include because major mistake)——
is negative or not because constraints satisfy we get -ve edge.
So we can say our solution is too bad because it is not satisfies any constraints
so alternative path should be subtracted by θ



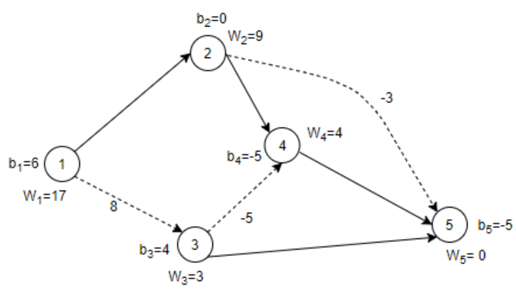
so we are interested in calculating maximum amount of θ
if $\theta = 4$ (because cannot send negative value, so θ not greater than 4)



now new cost is
$Cost = 6 \times 8 + 6 \times 5 + 4 \times 3 + 1 \times 4 = 94$
so feasible solution become



Repeat again
lets $w_5 = 0$ again calculating for non - basic edges



How to edges satisfying constraint-
edge (1, 3) not giving negative value so we will include that value in optimal solution.
So by keep repeating these steps will all non - basic edges satisfy.
$w_i - w_j - c_{ij}$ is negative .
you finally get optimal Cost = 81.

Day 4

Graph Coloring*

4.1 Motivation of graph coloring:

The motivation is comes from the example of Time - Table scheduling.
Lets say, I have five courses.These courses are Physics, Calculus, Electronic, Mathematics-I, Operating System.

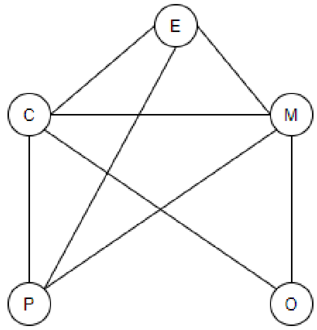
	Physics	Calculus	Electrical	Mathematics 1	Operating system
Physics		X	X	X	
Calculus	X			X	X
Electrical	X	X			
Mathematics 1	X	X	X		X
Operating system	X	X		X	

Suppose physics and calculus have common students represent by X, in this way other classes also have common students.So this problem solve by coloring .
Where common student of two classes represent by edge and classes represented.
So represent this table in (SOMETHING MISSING) by node

Nodes and edges

So graph from corresponding table is

*Lecturer: Dr.Anand Mishra. Scribe: Neelu Verma.



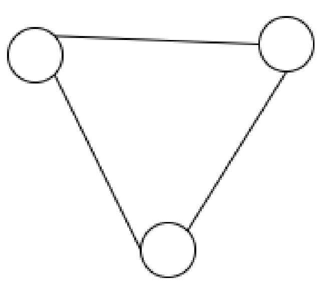
slots are now color

8 - 9 AM	G → green
9 - 10 AM	R → red
10 - 11 AM	Y → yellow
11 - 12 AM	C → cyan
12 - 1 PM	B → blue

So the slots can be though of as some color. So how many slots done we need or how many color requires .
So no class will clash.
If C is red so no other connectly node will be red.

4.2 Definition:

A K-coloring of graph G is a labeling $f : V(G) \rightarrow S$
where S is a set of K colors. A k-coloring is proper if adjacent vertices have different label.



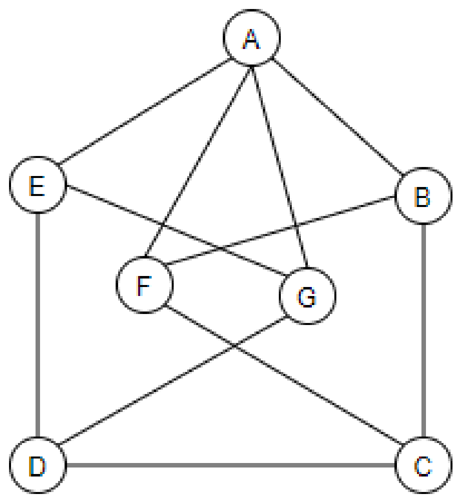
$S = \{Red, Green, Blue\}$
$K = 3$ then you get proper coloring.

Chromatic number X=(G)=the least K such that G is K-colorable.
For example Chromatic number for this graph is 3.

4.2.1 Compute the clique number:

Problem 1:

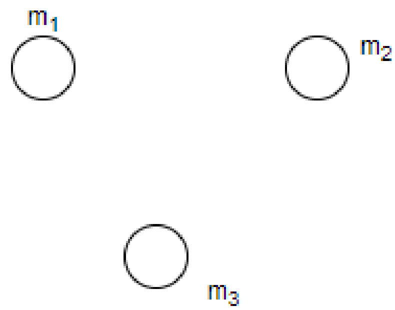
Compute the clique number, the independence number and the chromatic number of the graph below.



Clique number = 3, Independence number = 2, Chromatic number = 4.
ID set = $\{A, D\}$

Problem 2:

Prove that the chromatic number of a graph equals the maximum of the chromatic numbers.
Solution: Lets say there are 3 number of component and one of them is maximum.
Let m1 is max:



Suppose there are K components in the graph and component C_k requires m_k colors chromatic number of graph is the max chromatic number of its component
$C_m \rightarrow m_{max}$
Suppose m_1 require 10 color, m_2 requires and m_3 require 4.
So chromatic number of graph = 10.
Clique number ($w(G)$)
Chromatic number ($X(G)$)
Independence number ($\alpha(G)$)

$$X(G) \geq \frac{n(G)}{\alpha(G)}$$

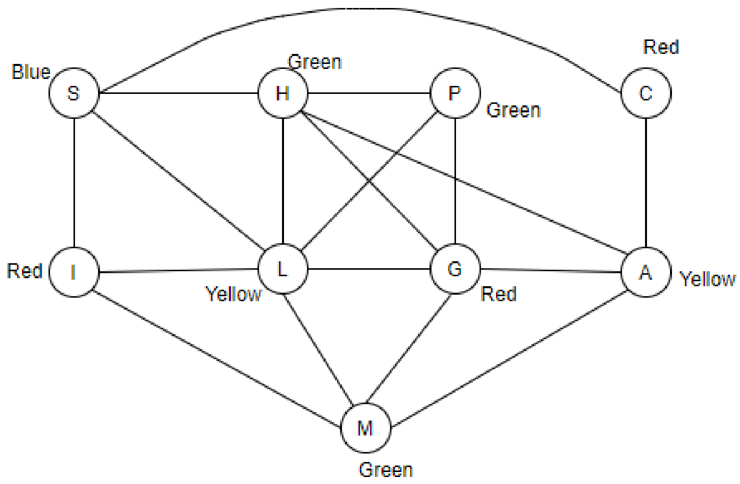
if $\alpha(G)$ is 1 (in case of fully connected graph) than equal. Otherwise chromatic number is greater than clique.

4.3 Greedy Algorithm for graph Coloring

Algorithm steps:

Let us have set and color set
$V(G) = V_1, V_2, ..., V_m$
$S = C_1, C_2, C_3, ..., C_k$
Step 1. Randomly choose any vertex V_k and assign first unused color.
Step 2. Again randomly choose vertex and if they have used color from the K^{th} color, then use first available color.
Step 3. Repeat step 2.

Example 1:



step 1- Choose vertices in following random order.

G, L, H, P, M, A, I, S, C

step 2- Choose first vertex and assign first color
step 3- Choose next vertex from the list and if it connected with previous assigned vertex then use unused color.
step 4- follow these steps until vertex assigned some color

References:

- 1. Introduction to Graph Theory - Douglas B West - 2nd Edition
- 2. Graph Theory, A NPTEL Course, by S.A. Choudum,IIT Madras
- 3. Fast Approximate Energy Minimization via Graph Cuts