Week 10

Theorems - Graph Theory*

10.1 Pseudo Boolean Function

A function with the domain is 0,1 and range is a real number, and the function is like:

$$f(x) = 3x_1 + 4\bar{x}_1 + 5x_2 + 3\bar{x}_2 + 6\bar{x}_1\, x_2$$

This is a type of pseudo Boolean function which has 2 types of terms, terms with single variable and terms with 2 variables. This x_1 and x_2 are binary variables where $x_1, x_2 \in \{0, 1\}$

The task is to minimise this function Min f(x) This is a constraint of optimisation and a minimiser of x so a brute force method, will be like you will have to assignment of each of there. The brute force method will be like X1, x2 and whatever the value of x will make a truth table like below:

x_1	x_2	f(x)
0	0	-
0	1	-
1	0	-
1	1	-

There will be some values of f(x)

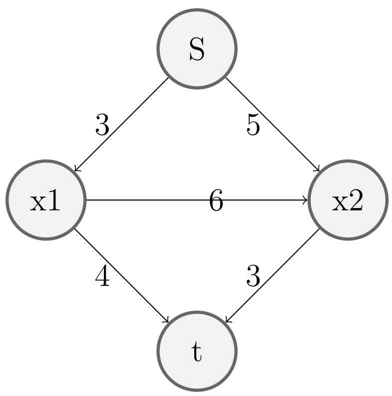
This is a brute force method and this may not be practical in case where you your this x is are very high. Even if xis are thousands nodes then the calculation we will have to do is 2 to the power of 1000. It will be a very high number for thousand variables Therefore the brute force method is not something which is of use we need something better and a graph cut provides that.

In graph cut what you do is to minimise this, so the graph cut, we construct a graph ,

*Lecturer: Dr. Anand Mishra. Scribe: Abhilash Sharma.

so unbar one or lets look at number one, lets construct the graph construction and this is a program as follows.
1. Every cut of that graph corresponds to some assignment of assignment of variables in this case xis.
2. We need such graph and we also need the mean cut is equal to minimum cost assignment.

If we could come up with this graph, then we are done. Mean cut of a graph in a polynomial time and we may be able to use that also and find the mean cut. To construct the graph is easy.



We have to minimize the below function

$$f(x) = 3x_1 + 4\bar{x}_1 + 5x_2 + 3\bar{x}_2 + 6\bar{x}_1\ x_2$$

We have to minimise this. We will have a variable for every node. There are 2 variables so 2 nodes. Two additional nodes will be there source and sink (special nodes) The connections making is required. If we are coming out of some node then we put a bar, a incoming edge comes from source to that node otherwise.
$6\bar{x}_1\ x_2$ will be from x_1 to x_2 as there is a bar on x_1 so, the flow will be out of x_1 and going to x_2

NOT ALL FUNCTIONS CAN BE OPTIMIZED. There are lot of applications with this sort of graph
It can be generalised for n variables. Now we have to find the mean cut
Let us select this path S $\rightarrow$ x2, so the value will be 3 and the node will be exhausted. X2 $\rightarrow$ t will be exhausted. S $\rightarrow$ x2 will be 5-3 now The other one is s $\rightarrow$ x1 and x1 $\rightarrow$ t will be 4-3 and s $\rightarrow$ X1 will be exhausted. The 2 edges are exhausted and we cannot do anything more.
So 3+3 = 6 is the mean cut. And this is the **answer**.

And the mean cut will be a line cutting x1 $\rightarrow$ x2. So we will be assigning x2 = 0 and x1 = 1.

f(x) = 3+0 + 0 + 3$\bar{x}_2$ that is 3 so it will be 6.
Mean cut is 6 and x2 is in the Source side and x1 is in the target side, so whatever is in the source side we assign zero, and for target side we assign 1.

Application of this theory:
One of the application is in computer vision where it is used for image segmentation. The task is as follows, there is an image of board, and the code is to segment the image out. This can be used using Graph cut with appropriate formulation of optimisation function right so that's the only thing.
An example of image segmentation in energy minimisation, when we say energy minimisation it is same as objective function minimisation. Read energy as objective function.

FUNCTION COMES HERE FROM IMG1:
The cost comes with some kind of probabilistic estimate. We model each of these var as pixels. Its a matrix of pixels. xi can be background or foreground. For exam of these pixels we have certain cost, that we learn from user distribution or some input,. We have to find the global minima of the function.

Create a 3x3 image and then there are some pixel values with some pixel values. Goal is to segment it out as follows which will be 1 and zeros in matrix. Image has scale of 0-255 Define the cost function as: Variables as x1, x2, x3…x9, each having some cost of assignment as some value close to 255 is 1 and close to zero is zero.

p(x1) for first row and first column = 250 (255-p(x1)) x1 + p(x1).$\bar{x}1$ We can generalise the cost as a formula $\Sigma(255 - p(xi))xi + p(xi).\bar{x}1$
If xi and xj are neighbor lets say, then if they take different label then it will take some cost. For any bigger image then usually your neighbors are same if the pixel is object its neighbour is also more likely to be object m, its not like random, so image is not random, and images have some structure.
If an image is mouse, then the neighbor pixel is also likely to be mouse.

We apply graph cut, and find the cut where xi is assigned is one and where xi is zero we assign it as zero. We assign it to 3x3 image, and the foreground will be segmented.

Revision: Create a graph such that any cut corresponds to an assignment of two of x and the cost of the cut is energy of x or f(x) minimum of cut is equal to that particular assignment.
A similar example: 2a1 we will have edges for 2 , 5$\bar{a}1$ we are coming from a1 therefore 5$\bar{a}1$ similarly 9a2, 4$\bar{a}2$ means we are coming out of a2 and then 2a1$\bar{a}2$ will be coming out of a2 and going to a1 similarly a1$\bar{a}2$ means we are going out of a1 and going to a2.

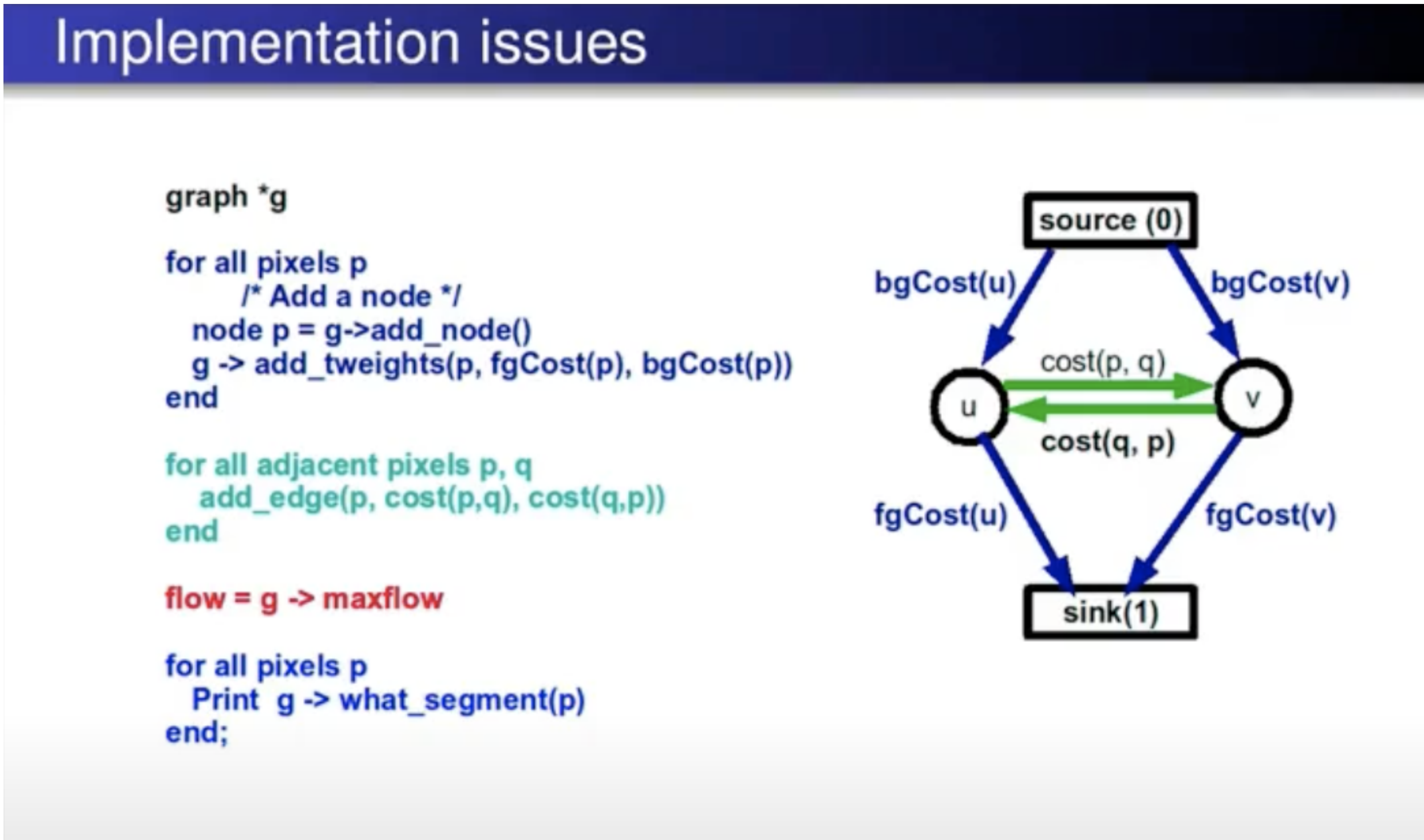


Image ref: Lecture slides

There are few implementation issues. Algorithm of which is mentioned above

What type of function can Graph cut can support:
In general if any function is pseudo Boolean function this is np hard to minimise, only approx solution are available. Easy energy function (sub modular functions) are graph representations and solvable in polynomial time.

Sub modular function has some property:
1. Let f be a Boolean function or a be a function defined over a set of bool var x1 to xn, then all Boolean variable are sub modular.
2. A function f is of 2 bool variable is sub-modular if it is a one bool variable like we don't have the terms with 2 variable, then its by default a sub modular
$f(0,0) + f(1,1) \leq f(0,1) + f(1,0)$
3. In general a function is sub-modular if all its projection to 2 variables are sub modular.

10.2 Max flow, Min Cut theorem

It is due to Ford and Fulkerson. Max flow is another algorithmic hammer that's being used to solve a wide variety of problems.

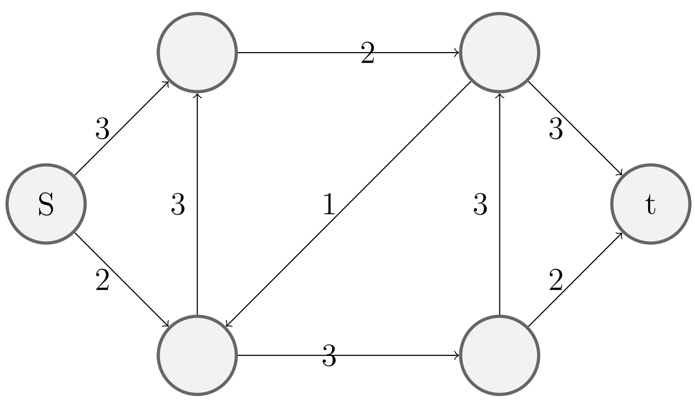
What is a flow network: Its a graph, having vertices and edges. We will look at directed graphs. IN flow network we have source (S) and a Sink (T).

There is some flow coming out of S and is gone to have some constraints capacity of edges going towards T. We cannot allow accumulation in intermediate nodes. It can be a commodity of water flowing. Everything coming to vertex has to leave vertex.

Lets look a flow network: we will follow the same example through the lectures. We will have directed edges (U, V) U to V. (U,V) $\in$ E, and each edge is going to have a non-negative capacity, C(u,v) , if there is no edge between two vertices then it is assume to be zero. IF (U,V) $\notin$ E, C(U,V)=0

This means that there is no road so that you can drive your car from U to V.

Let us see a graph:



This graph can have cycles in it. The numbers in the graph are the capacities C(U,V).

We have 2 numbers associated with each edge, one is capacity and other is the flow that goes through the edge. We gave this constraint that the flow can never exceed the capacity, and that a constraint associated with each edge.

A real simple example of flow in this network.

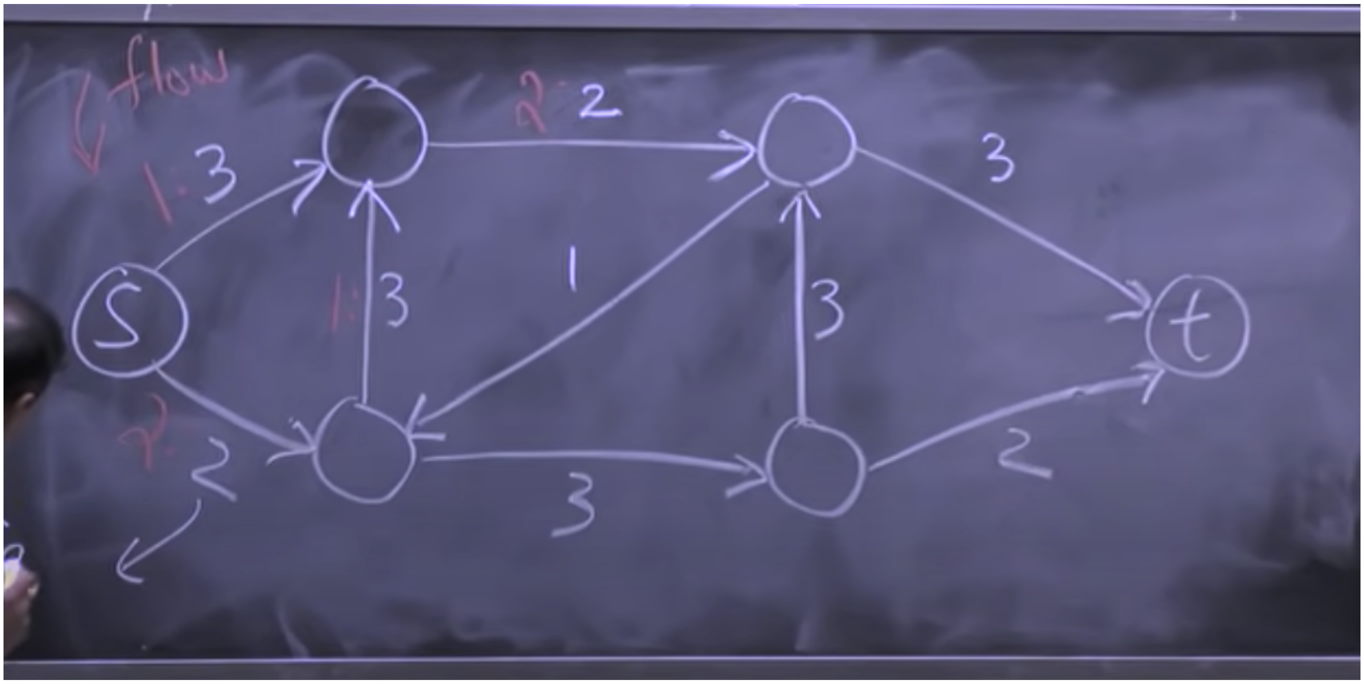


Image ref: MIT course content

Law of conservation states that any value that goes into a node which is not S or T has to leave the node.
We can have flows that are cyclic. In one of the cycle, we subtract One from the flows value. So we have 2 coming into the node, 1 going out and 1 going out, and nothing coming back to the node, so that is good to go. This is a conservation law.
We have to maximise the flow from source and push it into the sink. The value coming to T is 2+1.

Can we have a way to increase the flow, we can anyways decrease, by making all the values zero?
We can decrease flow in couple of edges and increase the overall flow. After alterating the values of flow we get a maximum of 4 at T.

We keep on improving the current flow gradually. But we never exceed the capacity.

Definition: Given a flow network G, find a flow with maximum value on G.

In the above example, we can think to have a value of 5 at T, because we can get it from the previous nodes, but if we consider the theory of conservation, we cannot obey those laws and get 5 units from Source S and get it on sink T.
We will prove it using Ford Fulkerson algorithm.

Assumptions:

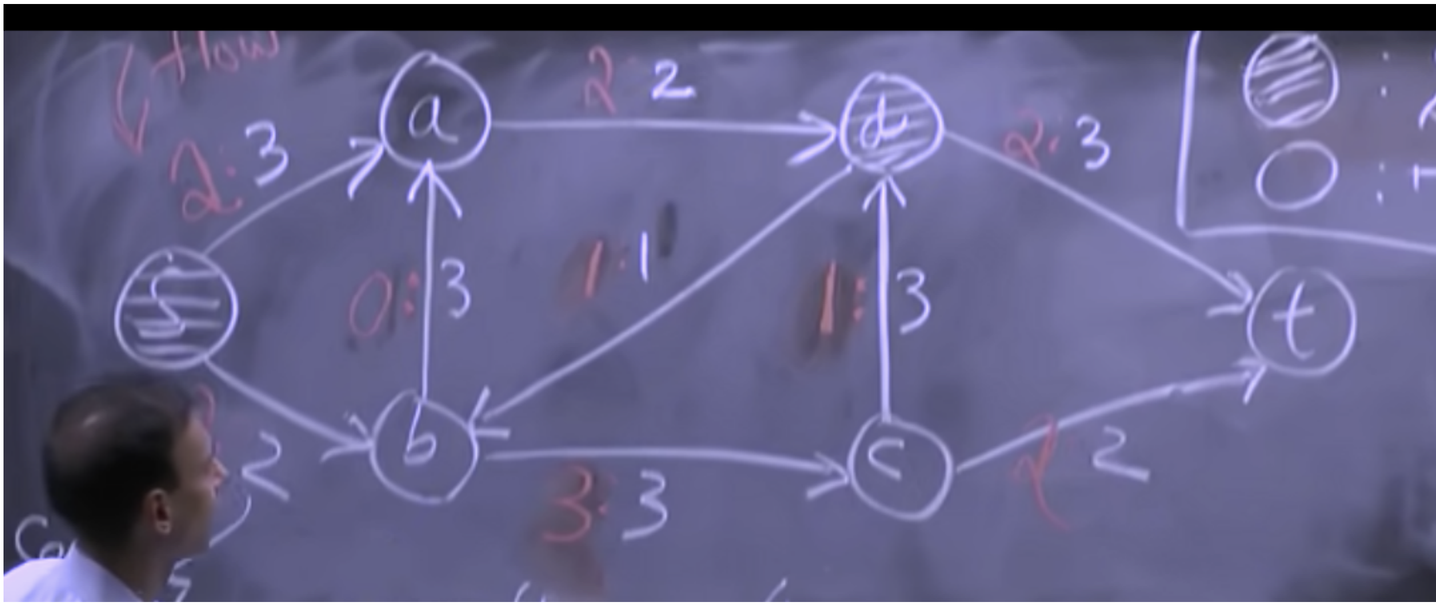


Image ref: MIT course content

If we have cycles and assume that 1 unit is coming to S1 node and 1 is going out of it., then there is zero flow network. We cannot have 2 going out to S1 as there has to be something more coming in towards Node S1.

We are going to disallow cycles of two kinds:
a. No self loo edges allowed. B. And subgraphs like S1U and US1 and they have non zero capacities. Transform U into something which has equal flow coming in and going out.

Flow: A flow on G is function f: VxV $\rightarrow$ R satisfying the capacity constraint For all u,v E V, f(u,v) $\leq$ C(u,v) Flow conservation: $\forall u \in V$ - {s, t}

f $\Sigma f(u, v) = 0$

Skew symmetry : $\forall u, v \in V$, f(u,v) = - f(v,u)

If there is a flow from u to v then the value of that flow is the negation of flow from v to u.

The value of flow F is denoted and $\|f\|$ i.e. cardinality of f

$$\|f\| = \Sigma f(s, v) = f(s, V)$$

Given a flow network one particular quantity has to be maximised, it is based on how much we can push out from s. And all added up together.

Properties of flow:
1. f(x,x) = 0 this means it is only a single member, and we do not allow loops.
2. f(alb) + f(b,a) =0 because of skew symmetry.
3. f(x, y) = - f(y, x)
4. f(xUy, z) = f(x, z)+ f(y, z) if x intersection y is not null

Theorem:
$\|f\| = f(V, t)$

F is the value which gets pushed from Source to Sink.

Proof: $\|f\| = f(s, V)$, definition of cardinality of f or value of f
$\Rightarrow f(V, V) - f(V - s, V)$

Can we say something about these quantities ¿ f(V, V) = 0
$\|f\| = f(V, V - s)$
= f(V, t) + f(V, V-s-t)
=f(V, t) - f(V-s-t, V)

Because we have property that corresponds to an intermediate vertex, the flow goes out to all vertices, the conservation says it has to be zero. So f(V-s-t, V) = 0
Therefore $|f| = f(V, t)$ **Proved**

Concept of cut: Cut is basically a partition one nodes, we cannot have node on both side, it is a disjoint sets.
Source is on one side of the cut and T is another side of the cut.
Cut is (S,T) of a flow network G G=(V,E)
Is a partition of V such that s E S and t E T

If f is a flow on G, then the flow across the cut f(S,T)

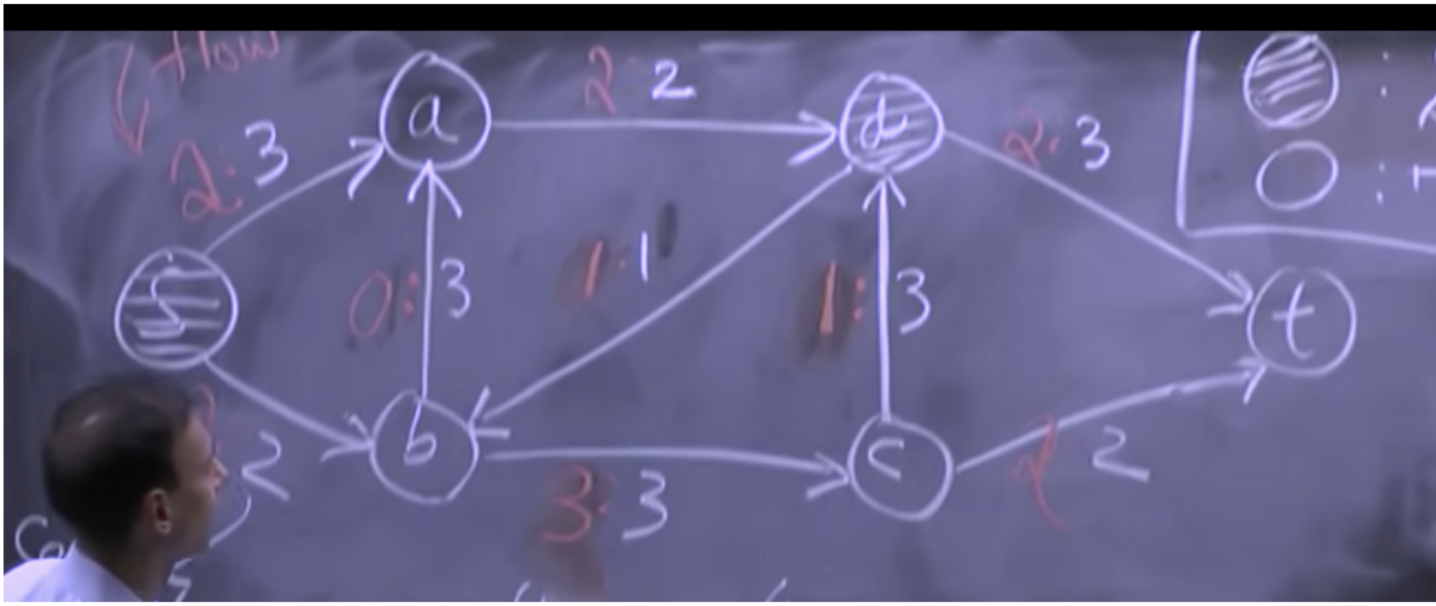


Image ref: MIT course content

f(S, T) = (2 + 2) + (-2 + 1 - 1 + 2)
where
sa = 2
sb = 2
da = -2
db = 1
dc = -1
dt = 2

Look at every pair of vertices and using Skew symmetry see what edges are coming out or coming in
Capacity of a cut c(S, T) = (3 + 2) + (1 + 3)
where
sa = 3 sb = 2
db = 1
dt = 3

Therefore Capacity = 9

Theorem:
The value of any flow is bounded by the capacity of any cut

Lemma:
For any flow f and any cut (S, T) We have $|f| = f(S,T)$

Proof:
For any given cut, the flow of that cut is bounded by the capacity of that cut.
f(S,T) = f(S, V) - f(S,S)
Because f(S,S) = 0
Therefore f(S,T) = f(S,V) =f(s, V) + f(S-s, V)
f(S-s, V) does not contain t, therefore we can use flow conservation, and its value will be Zero
Hence f(s, V) = $|f|$