

## LESSON: MF5: Max-flow Generalization

\*\*\*

(Slide 1) Max-flow with Demands

### Demands

#### Max-flow with demands:

input: flow network: directed  $G=(V,E)$  with  $s,t \in V$   
capacities  $c(e) > 0$  for  $e \in E$

& demands  $d(e) \geq 0$  for  $e \in E$

Feasible flow is a flow  $f$  where:

for  $e \in E$ ,  $d(e) \leq f(e) \leq c(e)$

Is there a feasible flow?

Let's look now at a generalization of Max-flow problem: We're going to add in demand constraints along every edge. We're going to see how to reduce this generalization of the Max-flow problem to the standard Max-flow problem. Let's formalize this generalization of the Max-flow problem.

The input starts off the same as before for the Max-flow problem. So we have a flow network, which consists of a directed graph,  $G$ , with a specified start and end vertex,  $s$  and  $t$ , along with the capacities for every edge.

Now the new aspect is that we're also given the demands for every edge. So, for edge  $e$ , we're giving the non-negative number,  $d(e)$ , which specifies the demand for this edge. Now our goal is to find a feasible flow.

What exactly do we mean by feasible flow? A feasible flow is a valid flow as before. So, along every edge, the flow is at most its capacity, but, we have the additional constraint that the flow must be at least the demand along that edge. And of course, the other constraint still holds as before: that, for every internal vertex - for every vertex except for  $s$  and  $t$  - we want the flow in to equal the flow out.

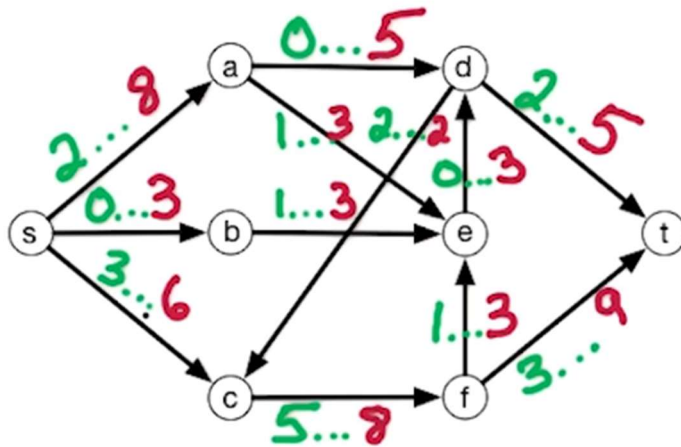
So we're looking for a valid flow as before with the additional constraint that the flow along every edge is at least the demand along that edge. Now if we can find a feasible flow, of course we want a feasible flow of maximum size. But the first problem is whether we can even find a feasible flow.

Is there a way to satisfy all of the demands? That's the first question we're going to address. Can we figure out whether there is a feasible flow for this input? Once we find a feasible flow, then will be straightforward to augment it into a maximum size feasible flow.

\*\*\*

(1<sup>st</sup> Slide 2) Feasible Flow Example

## Feasible flow example

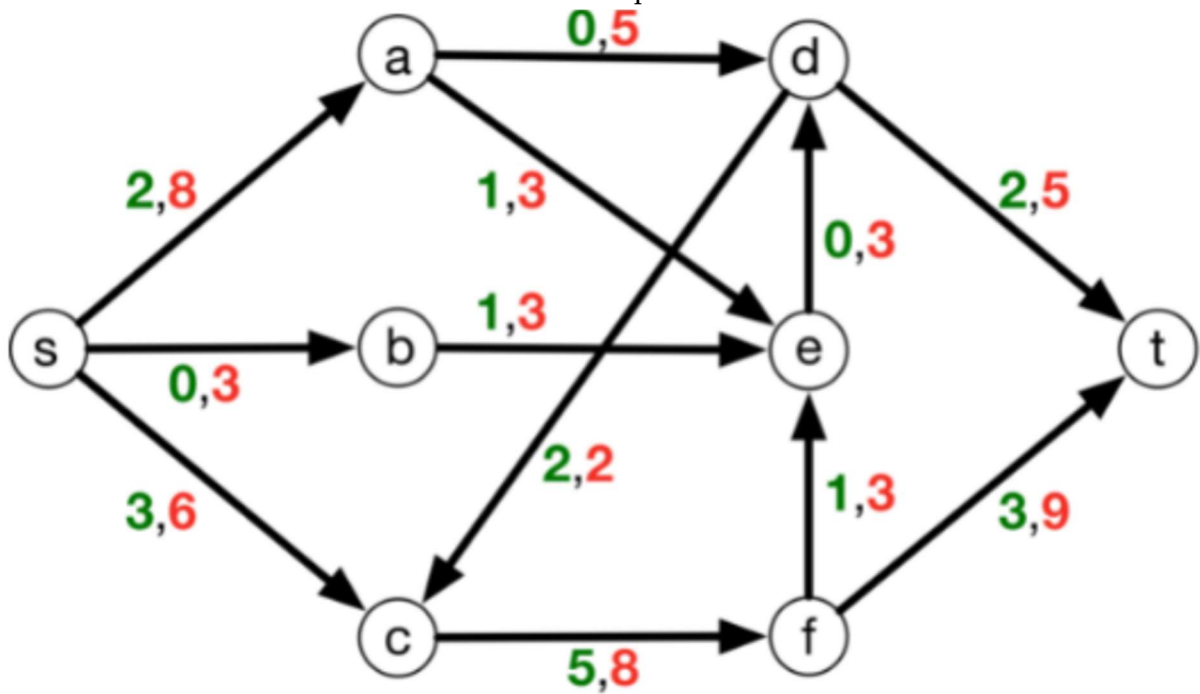


Let us take a look at one of our flow network examples from earlier. And let us look at the feasible flow problem in this example. So here is a directed graph with the specified start and end vertex. Let's mark the demand and the capacities along every edge: the green numbers specify the demand along the edge and the red numbers specify the capacity along the edge.

So along this edge (s,a) , we want the flow to be at least two units and at most eight units. Along this edge (s,b) , there is no demand constraint. It has got to be at least 0 and at most 3. Here, along (s,b), it has got to be at least 3 and at most 6.

Is there a feasible flow for this example?

MF5: Max-flow Generalization: Feasible Flow Example



Try to enter a valid flow that meets the demand and capacity constraints for the graph.

Question 1: provide a valid flow for sa

Question 2: sb

Question 3: sc

Question 4: ad

Question 5: ae

Question 6: be

Question 7: cf

Question 8: dc

Question 9: dt

Question 10: ed

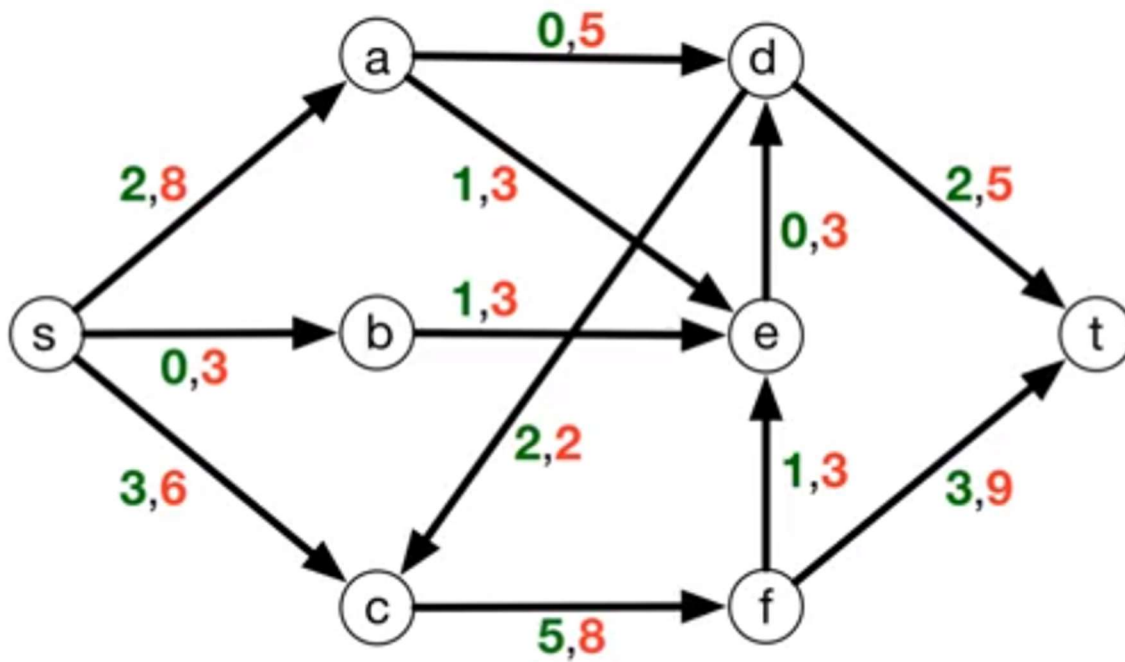
Question 11: fe

Question 12: ft

\*\*\*

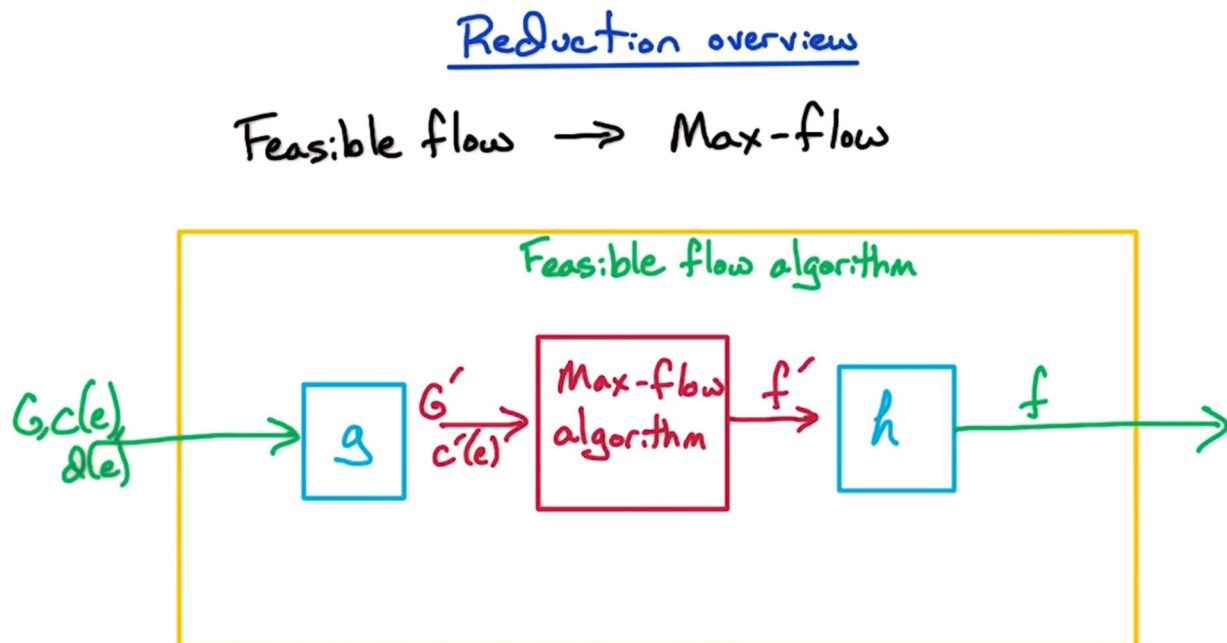
(2<sup>nd</sup> Slide 2) Feasible Flow Example (Answer)

Try to enter a valid flow that meets the demand and capacity constraints for the graph.



\*\*\*

(Slide 3) Reduction Overview



Now, we're going to show a reduction from the feasible flow problem to the Max-flow problem.

Recalling the feasible flow problem, we're trying to find a feasible flow, not necessarily of maximum size. In the Max-flow problem we're trying to find a valid flow of maximum size.

Formally, what do we mean that we're reducing this problem to this problem? What it means is that we're going to use our algorithms - our polynomial time algorithms - for the Max-flow problem as a black box. So, it's as if we have this subroutine that we downloaded from the library and we don't want to touch the insides of this subroutine. We can simply give it an input flow network and we'll get out a solution - a flow,  $f'$  of maximum size for this input flow network.

Now, we want to use this black box algorithm for Max-flow to build an algorithm for the feasible flow problem. What we're going to do is:

- we're going to take an input to the feasible flow problem.

What exactly is input to a feasible flow problem? - well, a directed graph  $G$ , capacities along every edge, specified by  $c(e)$ , and there's also demands along every edge specified by  $d(e)$ .

- Then, we define a function  $g$  which transforms this input to the feasible flow problem, into a valid input to the Max-flow problem.
  - Once again what does the input to the Max-flow problem look like? Well, it consists of a directed graph - not necessarily the same directed graph that we have for the feasible flow problem - we're going to make a new graph.

So, we're going to define a new directed graph  $G'$ , based on the directed graph  $G$ .

And, we also need to specify the capacities along every edge. So, we'll define  $c'(e)$  to be the capacity along edges in this graph  $G'$ .

But, there are no demands in this graph. The input to the standard Max-flow problem does not have any demand constraints.

So, we have to somehow encode these demands in these capacities and in this graph  $G'$ .

- Then, we're going to run our black box algorithm for the Max-flow problem on this input. We're going to get some output  $f'$ ,
- and then we have to transform this output into a solution to the original feasible flow problem.

So, we have to define a function, which we denote as  $h$ , which transforms this output, this blackbox solution to the Max-flow problem on this input graph,  $G'$  with capacity  $c'(e)$ .

And  $h$  has to transform  $f'$  into a flow  $f$ , where  $f$  is a feasible flow for this input,  $G$ , capacity  $c(e)$  and demands  $d(e)$ .

So, to reduce feasible flow to Max-flow, what we need to do is:

- we need to define this transformation function  $g$  which transforms an input from feasible flow into an input for Max-flow.
- and we need to define this function  $h$ , which transforms the solution  $f'$  for the Max-flow problem on this input  $G'$ , and produces a solution  $f$ , which is a feasible flow for this

input network,  $g$  with capacity  $C$  and demands  $d$ .

So, in order to reduce feasible flow to Max-flow, we need to specify this function transformation function  $g$  and this transformation function  $h$ .  $g$  transforms the input and  $h$  transforms the solutions.

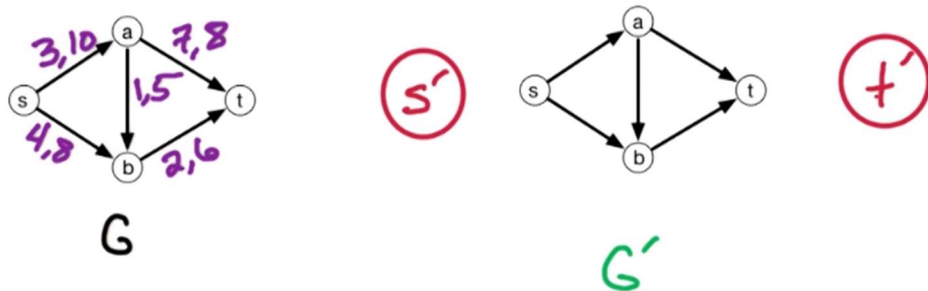


\*\*\*

(Slide 4) Reduction Vertices

### Reduction: vertices

Given input  $G, c(e), d(e)$  for feasible flow  
-construct input  $G', c'(e)$  for max-flow



So let's dive into the reduction.

Let's take a particular input for the feasible flow problem. This is specified by a directed graph  $G$ , capacities along the edges, and demands along the edges.

Now we need this to construct an input for the max flow problem. This is specified by a directed graph which we'll call  $G'$  and capacities along the edges which was specified by  $c'$ . These capacities don't have to be the same as the original capacities. In fact, we need these capacities to encode not just the capacities in the original graph, but also the demands in the original graph.

Let's take a look at a particular example ... very simple example: Here's a small example of four vertices:

- this is our directed graph  $G$ , which is our input to the feasible flow problem, and we have some example capacities and demands - We've written a pair of numbers for each edge.

The first number, for example  $(s, a)$  (3, 10) - 3 specifies the demand, and 10 - the second number - specifies the capacity.

- Now let's try to create a graph  $G'$ . We want to keep all this information, so we're going to keep the same graph  $G$  and we're going to add additional vertices to it.
  - So this is going to be our directed graph  $G'$  which is going to be the input to the max flow problem. So we're going to start with the same graph  $G$  that was the input to the feasible flow problem and we're going to add a new start vertex which will denote as  $s'$  a new end vertex which will denote as  $t'$ .

So the original  $s$  and  $t$  will now serve as internal vertices. These are the vertices of this new graph  $G'$ .

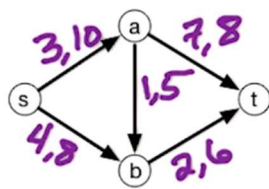
Let's now look at the edges that are in this graph and also the capacities along the edges in this new graph.

\*\*\*

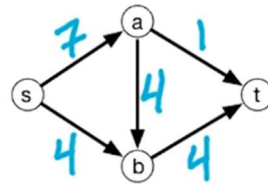
(Slide 5) Original Edges

## Original edges

Given input  $G, c(e), d(e)$  for feasible flow  
 -construct input  $G', c'(e)$  for max-flow  
 $c'(e) = c(e) - d(e)$



$s'$



$t'$

Intuition:  $f(e) \geq d(e) \iff f'(e) \geq 0$   
 $c(e) \geq f(e) \iff c'(e) \geq f'(e)$

Now the edges in the original flow network,  $G$ , remain in this new flow network  $G'$ . What will change is the capacities along these edges.

Now we want the capacities along these edges to capture the capacities along the original network AND the demands in the original network. What we want to capture is that there is non-negative flow along an edge if and only if we can construct a flow in the original network where the flow is at least the demand. And, we also want that the flow in this new graph,  $G'$ , satisfies this new capacity constraint if and only if this flow  $f$ , that we're going to construct, satisfies the capacity constraint in the original network.

Now, the idea is that we want to flow 0 in this new network to correspond to flow  $d$  in this original network. So this flow  $f'$  is going to be a shift of this flow  $f$  by  $d$  units.

We're going to shift the capacity  $c$  as well by  $d$  units. So, the capacities along these edges, which were edges in the original network, will be the original capacities minus the demand.

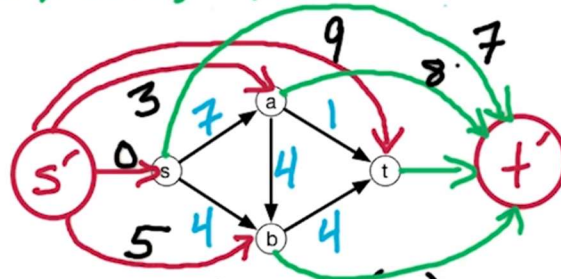
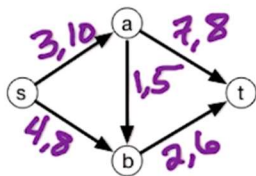
So, whereas this edge  $(s,a)$  originally had capacity 10, now it'll have capacity 7. This edge  $(a,t)$  will have a capacity of 1,  $(a,b)$  4,  $(b,t)$  4, and also  $(s,b)$  5.

\*\*\*

(Slide 6) New Edges

## New edges

Given input  $G, c(e), d(e)$  for feasible flow  
 -construct input  $G', c'(e)$  for max-flow



for  $v \in V$ , add  $s' \rightarrow v$  with  $c'(\overrightarrow{s'v}) = d^{in}(v)$   
 add  $v \rightarrow t'$  with  $c'(\overrightarrow{vt'}) = d^{out}(v)$

So what we've done so far in our reduction is for each edge in the original network  $G$ , we've added that edge to this new network  $G'$ . And, the capacity of that edge in the new network is the original capacity shifted by the demand. Now we're going to have to add additional edges to this graph. Let's get some intuition for what these additional edges have to accomplish.

Now we want a valid flow here in this graph  $G'$  to correspond to a feasible flow in  $G$ . So, suppose we have a zero flow here in  $G'$ . So we have flow zero from  $s$  to  $a$ , zero from  $a$  to  $b$ , and zero from  $a$  to  $t$ .

Now what is that going to correspond to over here in  $G$ ? Well, zero flow along this edge  $(s,a)$  in  $G'$  is going to correspond to flow three along this edge  $(s,a)$  in  $G$ . Zero here (along  $(a,b)$  in  $G'$ ) is going to correspond to flow one here (along  $(a,b)$  in  $G$ ), and zero here (along  $(a,t)$  in  $G'$ ) is going to correspond to flow seven here (along  $(a,t)$  in  $G$ ), because the minimum flow here (in  $G'$ ) is going to correspond to the minimum flow here (in  $G$ ).

Now certainly zero flow in  $G'$  is valid. The flow into  $a$  is zero and the flow out of  $a$  is zero.

But in  $G$ , for vertex  $a$ , we have flow in of 3 units and we have flow out of 8 units. So it's not a valid flow in  $G$ . The flow into  $a$  does not equal the flow out of  $a$ . So, we need to offset that

somehow in this graph so that a valid flow in  $G'$  will correspond to at least a valid flow in  $G$ .

We're going to accomplish that by adding an edge from  $s'$  to each of these vertices of the original network, and from each of these vertices of the original network to  $t'$  will have an edge.

Let's look first at the edges from  $s'$  to each of the vertices of the original network. So for each vertex in the original network, we add this edge from  $s'$  to  $v$ . Now, what is the capacity of this edge from  $s'$  to  $v$  in this new network  $G'$ ?

- $(s',a)$ : well, we shifted the capacity along this edge into  $a$  by 3 units corresponding to the demand into  $a$ . We want to offset that by this edge from  $s'$  to  $a$ . So we're going to define the capacity of this edge to be the demand into the vertex  $v$ . So the total demand into the vertex  $a$  in the original network is 3 units. So the capacity of this edge from  $s'$  to  $a$  in  $G'$  will be three units.
- $(s',t)$ : Now,  $t$  has total demand in of  $7 + 2$ , so 9 units. So this edge  $(s',t)$  will have capacity 9,
- $(s',b)$ :  $b$  has demand in five units, so this edge  $(s',b)$  has capacity 5
- $(s',s)$ :  $s$ , of course, has zero demand in. So this edge has zero capacity.

Similarly, we are going to add edges from each of these vertices of the original network to  $t'$ : Now we'll put these in a different color just to differentiate them from the edges from  $s'$ .

So we have  $(s,t')$ ,  $(a,t')$ ,  $(t,t')$  and  $(b,t')$ . What is the capacity of these edges to the vertex  $t'$ ? well, from  $s'$ , that captured the demand into the vertex  $v$  and these edges to  $t'$  are going to capture the demand out of the vertex  $v$ .

- $(s,t')$ : So vertex  $s$  has demand out of seven, so this edge has capacity seven.
- $(a,t')$ : vertex  $a$  has total demand out of eight, so this edge has capacity eight.
- $(t,t')$ :  $t$ , of course, has zero demand out,
- $(b,t')$ : and  $b$ , in this example, has demand out of two.

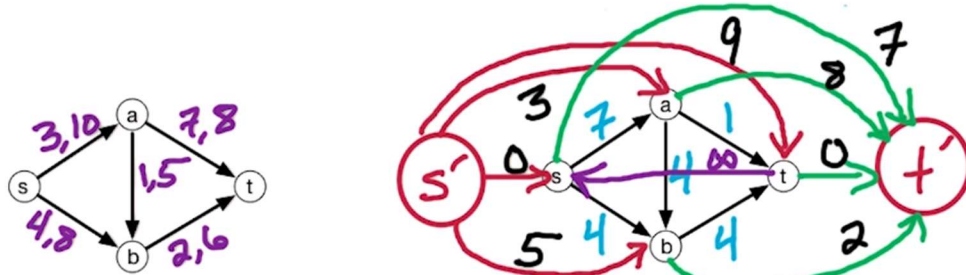
So, that fully specifies our graph  $G'$  which is our input to the Max-flow problem.

Now we have to discuss how a solution to this Max-flow instance corresponds to a solution to the feasible flow problem.

\*\*\*

(Slide 7) One More Edge

## One more edge



for  $v \in V$ , add  $s' \rightarrow v$  with  $c'(\overrightarrow{s'v}) = Q^{in}(v)$   
 add  $v \rightarrow t'$  with  $c'(\overrightarrow{vt'}) = Q^{out}(v)$   
 add  $t \rightarrow s$  with  $c'(\overrightarrow{ts}) = \infty$

There's one additional edge that I almost forgot to mention in this graph  $G'$  that we're constructing.

Notice, in the original graph,  $s$  is the start vertex and  $t$  is the end vertex. In this new graph,  $s$  and  $t$  are internal vertices. So, the flow in has to equal the flow-out for both these vertices. Whereas in the original graph, we don't have that constraint. Notice there's no flow in to  $s$  - there's no capacity into this vertex  $s$  - and  $t$  has no capacity. So, in the current set up we don't have any flow into  $s$  and any flow out of  $t$ .

For the new graph, how do we get around that, so that we can have the flow into  $s$  equal to the flow out and similarly for  $t$ ? - well, we know the flow out of  $s$  equals the flow into  $t$ , so we simply have to offset the flow out of  $t$  and the flow in to  $s$ .

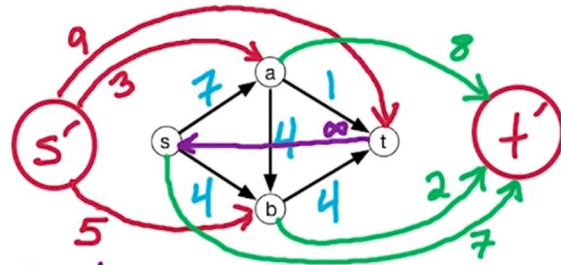
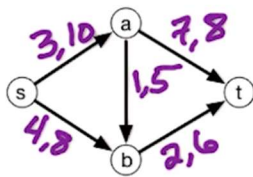
So, we compare these up with each other. So, we're going to add this edge from  $t$  to  $s$ . And we don't have any constraints on the flow along this edge - our flow goes into  $t$ . We want to send that back to  $t$ . So, we give this edge capacity infinity. So, to summarize, we're adding this edge from  $t$  to  $s$  and the capacity is infinite. Now, we really have completed our construction of this graph  $G'$ .

\*\*\*

(Slide 8) Saturating Flows

## Saturating flows

Lemma:  $G$  has a feasible flow  $\iff G'$  has a saturating flow



for flow  $f'$  in  $G'$ ,  $\text{size}(f') \leq D$

Say  $f'$  is saturating if  $\text{size}(f') = D$

Now, let's take a look at this graph we constructed. What is the maximum flow size that can occur in this graph? Certainly, it's upperbounded by the capacity - the total capacity out of  $s'$  or the total capacity into  $t'$ . Let's take a look at those quantities.

Let capital  $D$  denote the total demand in this original network  $G$ . So, we're going to sum over edges in this original network of the demand along that edge. Notice, this is the same as summing over the vertices of the total demand into the vertex, because every edge has one tail. And similarly, we can count from the head of the edge. So, we can sum over vertices of the total demand out of each vertex.

Now, in this graph, what is the total capacity out of  $s'$ ? Well, the capacity along this edge from  $s'$  to  $a$  is defined as demand into  $a$ . So this total capacity out of vertex  $s'$  is a sum over vertices in the original graph of the demand into that vertex - that's equal to the quantity  $D$ . Similarly, if we look at that the total capacity into the vertex  $t'$ . That's equal to the total demand out of the vertices, which is also equal to capital  $D$ .

So, if we look at the value of flow  $f'$  in this new graph  $G'$  that we constructed, what do we know about the size of this flow  $f'$  in this graph  $G'$ ? - We know it's upperbounded by the total capacity out of  $s'$  or the total capacity into  $t'$ . Both of those quantities are  $D$ . So, the size of this

flow  $f'$  is in most capital  $D$ . Therefore, we say this flow  $f'$  is saturating if the size of this flow  $f'$  is capital  $D$ . This means that this flow  $f'$  is of maximum size. So, all of these edges out of  $s'$  are fully capacitated, they're fully saturated. And all these edges into  $t'$  are fully saturated. So, in a saturating flow  $f'$ , we fully capacitated the edges out of  $s'$ , and fully capacitated the edges into  $t'$ .

What we're going to prove is that  $G$ , our original network, has a feasible flow if and only if, this new network  $G'$  (that we constructed) has a saturating flow. Once we prove this lemma, this gives us a way to find a feasible flow.

What we do is we take our input to the feasible flow problem, we construct  $G'$ , we run Max-flow on this graph,  $G'$ , and we check if the size of the Max-flow is equal to capital  $D$ , and therefore, if it is saturating or not. If it is a saturating flow, then we know that there is a feasible flow and actually our proof of this lemma will give us a construction.

So, we can take this saturating flow  $f'$  and construct a feasible flow  $f$  for this original network  $G$ . So, what we've done so far is transform the input to the feasible flow problem into an input to the Max-flow problem.

And now, by proving this lemma, we're going to show how to transform the solution to the saturating flow problem or to the Max-flow problem on this graph  $G'$ , into a solution to the feasible flow problem.



\*\*\*

(Slide 9) Saturating Feasible

Saturating  $\Rightarrow$  Feasible

Take saturating flow  $f'$  in  $G'$

- construct feasible flow  $f$  in  $G$ .

$$\text{Let } f(e) = f'(e) + d(e)$$

Check:

a)  $f$  is valid

b)  $f$  is feasible ✓

$$f'(e) \geq 0 \Rightarrow f(e) \geq d(e)$$

$$c'(e) \geq f'(e) \Rightarrow f(e) \leq c'(e) + d(e) = c(e)$$

( $\Rightarrow$ ) Let's first prove that the existence of a saturating flow  $G'$  implies the existence of a feasible flow in  $G$ .

So what we're going to do is we're going to take a saturating flow  $f'$  in this graph  $G'$  that we constructed, and we're going to use that to construct the feasible flow  $F$  in the original flow network  $G$ .

Now the construction is quite straightforward. Recall that the capacities in  $G'$  are shifted by the demand  $d$ . So, we're going to let the flow in  $G$  be equal to the flow in  $G'$ , plus the demand. If we have a zero flow in  $G'$ , then we'll have the minimum flow possible in  $G$ , which is the demand  $d(e)$ . And if we send, say 2 units of flow in  $G'$  along this edge  $e$ , then will send  $2 + d(e)$  in  $G$ .

Now there's two things we need to check.

(1) We need to check that  $f$  is a valid flow. So for all the internal vertices in this graph  $G$ , the flow in has to equal the flow out, and to be valid, we need that the flow along an edge is in most its capacity, and additionally,

(2) we need to check that  $f$  is a feasible flow. That means it satisfies all the demand constraints.

Let's start with this second condition. Let's check that  $f$  is a feasible flow. Now  $f$  is a flow. What does that mean for the flow along a particular edge  $e$ ? It's got to be non-negative. So, what does that imply about this flow  $f$  that we just constructed? Well then the flow  $f$  along the edge  $e$  is at least  $d(e)$  - zero flow corresponds to  $d(e)$  flow - and since the flow is non-negative, that means these flows  $f$  satisfy the demand constraints, so  $f$  is feasible.

While we're at it, let's look at the capacity constraints - the upper bound on these flows. We know this flow is at most  $c'(e)$  along the edge  $E$ . That implies that  $f(e) \leq c'(e) + d(e)$ . Recall that  $c'(e)$  was defined as  $c(e) - d(e)$ . Therefore, this quantity  $c'(e) + d(e) = c(e)$  in the original network  $G$ .

So, this flow  $f$  (that we constructed) satisfies demand constraints and satisfies the capacity constraints. So all that remains is to check that the flow in to every vertex equals the flow out.

\*\*\*

(Slide 10)  $f$  is Valid

Need to show:  $f$  is valid  
 $f^{\text{in}}(v) = f^{\text{out}}(v)$  for  $v \in V - s - t$

we know:  $f'^{\text{in}}(v) = f'^{\text{out}}(v)$

$$\begin{aligned} f'^{\text{in}}(v) &= f'(s'v) + \sum_{u \in V} f'(uv) \\ &= \cancel{Q^{\text{in}}(v)} + \sum_{u \in V} (f(uv) - \cancel{Q(uv)}) \\ &= f^{\text{in}}(v) \end{aligned}$$

Now in order to show that this flow  $F$  is valid, we need to show that for every vertex, the flow in equals the flow out of the vertex. Now, this is for every internal vertex. So this is for every vertex in the original graph except for  $s$  and  $t$ .

Now this is for the flow  $f$  that we constructed. What do we know? - we know that this flow  $f'$  was an output from the Max-flow problem. So,  $f'$  is a valid flow for the graph  $G'$ . So we know in this graph  $G'$ , the flow into the vertex  $v$  equals the flow out from the vertex  $v$ . So we want to use this fact  $f'^{\text{in}}(v) = f'^{\text{out}}(v)$  to imply this fact  $f^{\text{in}}(v) = f^{\text{out}}(v)$ .

So let's take a look at this first quantity -  $f'^{\text{in}}(v)$  - the flow into the vertex  $v$  in this graph  $G'$ . Now, which vertices have edges to the vertex  $v$ ? - this is in the graph  $G'$  that we constructed. Well, this new vertex  $s'$  has an edge to every vertex. So there's an edge from  $s'$  to  $v$ , and for every vertex in the original graph, if  $u$  was connected to  $v$  in the original graph, then it's still connected in this new graph  $G'$ .

Now what do we know about this first term ( $f'(s'v)$ )? - we assume that  $f'$  was a saturating flow. - that means that all the edges out of  $s'$  are fully capacitated. So this flow  $f'$  along this edge is up to its full capacity.

(describing the algebra on the equations) What's the capacity of this edge  $s'v$ ? - it was defined to

be the demand into the vertex  $v$ :  $f'(s'v) = d^{\text{in}}(v)$ . Now what about this latter quantity ( $\sum f'(uv)$ ,  $u \in V$ )? Well let's rewrite this flow  $f'$  along this edge in terms of  $f$ . The flow  $f'(uv)$ , by our definition, equals the flow of  $f$  along this edge  $uv$  plus the demand. Therefore,  $f'$  equals  $f$  along this edge minus the demand along this edge. Now, we can factor out this sum over  $u$  in each of these terms and then this latter quantity is minus the sum over  $u$  of the demand into  $v$  - so that sum is equal to the demand into  $v$ . So this term and this term cancel each other out and what are we left with? We're left with the flow into  $v$  in this flow  $F$ . So that means the flow into  $v$  under this flow  $f'$  is equal to the flow into  $v$  in this flow  $F$ .

Now that's not too surprising because we set up this flow, this capacity along this edge from  $s'$  to offset this difference. And, this was by design. But, we have this nice fact - the flow **into**  $v$  is the same in  $f'$  and in  $f$ ; and, similarly you can show that the flow **out of**  $v$  under  $f'$  and  $f$  are the same by similar algebra.

So therefore, if the flow into  $v$  is the same as the flow out of  $v$  under  $f'$ , and  $f'$  and  $f$  have the same flow into  $v$  and the same flow out of  $v$ , then we also have this statement that we want to prove. So, we've shown that  $f$  is a valid flow.

So we've shown that we can take a saturating flow  $f'$  and we can construct a feasible flow  $f$  by simply setting  $f$  equal to the flow  $f'$  plus the demand along the edge.

Now we have to prove the reverse direction.

\*\*\*

(Slide 11) Feasible Saturating

Feasible  $\Rightarrow$  Saturating

Take a feasible flow  $f$   
- construct saturating  $f'$  in  $G'$   
for  $e \in E$ , let  $f'(e) = f(e) - d(e)$   
for  $v \in V$ , let  $f'(\overrightarrow{s'v}) = Q^{in}(v)$   
 $f'(\overrightarrow{vt'}) = Q^{out}(v)$   
 $f'(\overrightarrow{ts}) = \text{size}(f)$

To prove the reverse direction, what we want to do now is to take a feasible flow  $f$  in the graph  $G$ , and we want to construct a saturating flow  $f'$  in this new graph  $G'$ .

For each edge of the original network  $G$ , we're going to define the new flow along that edge to be the old flow shifted down by the demand along this edge. That defines a flow for all the edges in the original network.

What about the new edges that we added to this graph  $G'$  when we did the construction? - well, for every vertex in the original network, we have an edge from  $s'$  and edge to  $t'$ . So, we have this edge from  $s'$  to  $v$  and  $v$  to  $t'$ . Now we're trying to construct a saturating flow. So each of these edges is supposed to be fully capacitated. So we might as well set this flow to be equal to its capacity - which is the total demand into  $v$  for this edge and the total demand out of  $v$  for the edge to  $t'$ .

That defines a flow  $f'$  for all the edges except for one. We have this edge from  $t$  to  $s$  which had infinite capacity. We want to set the flow along this edge to be equal to the size of this flow  $f$ .

That defines the flow  $f'$ . Now we have to check that  $f'$  is valid and it's a saturating flow.

\*\*\*

(Slide 12)  $f'$  is Valid

$f'$  is valid

$f$  is feasible

$$f(e) \geq d(e) \Rightarrow f'(e) \geq 0$$

$$f(e) \leq c(e) \Rightarrow f'(e) \leq c(e) - d(e) = c'(e)$$

Now we know that  $f$  is feasible. What does that mean?

- That means that the flow  $f$  along this edge  $e$  is at least its demand.

Now  $f'$  along this edge  $e$  is defined to be  $f$  along this edge minus the demand. So since this is true, we have the flow  $f'$  along this edge is at least zero. So  $f'$  is non-negative.

- Similarly, we have that  $f$  is at most its capacity.

Therefore,  $f'$  is at most the capacity minus the demand which is the definition of the capacity in this network  $G'$ .

So we know that this new flow  $f'$  that we constructed is non-negative and satisfies the capacity constraints.

Now we simply have to check that  $f'$  satisfies the flow-in equals the flow-out constraint.

\*\*\*

(Slide 13)  $f'$  Constraints

### $f'$ constraints

Need to check:  $f'^{\text{in}}(v) = f'^{\text{out}}(v)$

Know:  $f^{\text{in}}(v) = f^{\text{out}}(v)$

$$f'^{\text{in}}(v) = \cancel{d^{\text{in}}(v)} + \sum_{u \in V} (f(\vec{uv}) - \cancel{d(\vec{uv})}) = f^{\text{in}}(v)$$
$$f'^{\text{out}}(v) = f^{\text{out}}(v)$$

So we need to check that the flow into every vertex, except for  $s'$  and  $t'$ , under  $f'$ , equals the flow out.

Now we started from  $f$ , and we know  $f^{\text{in}}(v) = f^{\text{out}}(v)$ . So now we're going to do a little bit of algebra that's similar to before. So we to go from this fact to show this fact  $f'^{\text{in}}(v) = f'^{\text{out}}(v)$ .

(describing the algebra on the equations) Now, what is  $f'^{\text{in}}(v)$  - the flow into the vertex  $v$  under  $f'$ ? -

- where we have the flow from  $s'$ , we set that equal to the demand into the vertex  $v$ :

$$f'^{\text{in}}(v) = d^{\text{in}}(v) - \text{So this is the flow from } s'.$$

For every other edge, the flow from that vertex  $u$  to  $v$  in  $f'$ , it is defined to be the flow in  $f$  minus the demand along this edge.

$$f'^{\text{in}}(v) = d^{\text{in}}(v) + \sum (f(uv) - d(uv)), u \in V$$

As before,  $\sum d(uv)$ , the sum over  $u$  of this demand into  $v$ , is the same as  $d^{\text{in}}(v)$ . They're both equal to the total demand into  $v$ . So they offset each other, and we're left with the total flow into  $v$  under  $f$ . So as before, the flow into  $v$  under  $f$ , is the same as the flow into  $v$  under  $f'$ . And similarly, the flow out of  $v$  under  $f'$ , equals the flow out of  $v$  under  $f$ , by similar calculation. Therefore, since the flow into  $v$  equals the flow out of  $v$  under  $f$ ,

and  $f^{\text{in}}(v) = f'^{\text{in}}(v)$  and  $f^{\text{out}}(v) = f'^{\text{out}}(v)$ , we also have  $f'^{\text{in}}(v) = f'^{\text{out}}_v$ . And therefore,  $f'$  is a valid flow.



\*\*\*

(Slide 14) Max Feasible Flow

### Max feasible flow

$G$  has a feasible flow  $\iff G'$  has a saturating flow

Find a feasible flow  $f$  for  $G$ , if one exists

Start at  $f$ :

Augment in residual graph  $G^f$ .

$$c_f(\vec{vw}) = \begin{cases} c(\vec{vw}) - f(\vec{vw}) & \text{if } \vec{vw} \in E \\ f(\vec{wv}) - d(\vec{wv}) & \text{if } \vec{wv} \in E \\ 0 & \text{otherwise} \end{cases}$$

So we've proved this lemma:  $G$ , our original input, has a feasible flow if and only if our new construction  $G'$  has a saturating flow.

So, to solve the feasible flow problem to find a feasible flow in this input graph, we simply

- construct this graph  $G'$ ,
- we run the Max-flow problem,
- we check whether the size of this Max-flow equals  $D$ , and therefore whether or not it's a saturating flow. If it is a saturating flow, then we've shown how to transform that saturating flow  $f'$  into a feasible flow  $f$  for this graph  $G$ . So, we know how to find a feasible flow  $f$  for this graph  $G$  if one exists.

Once we have such a feasible flow  $f$ , can we transform it? Can we augment it to find a flow, a feasible flow of maximum size? - so, can we find a maximum size feasible flow?

Now recall how our Max-flow algorithms worked, either Ford-Fulkerson or Edmonds-Karp.

- Both of them start with a zero flow. So every edge has flow zero initially
- Then we build the residual graph and,
- we find an augmenting path from  $s$  to  $t$  if one exists.

- If there exists one, we augment along that path then we build a new residual graph and we repeat.

We keep augmenting along a path from  $s$  to  $t$  in the residual graph until there's no path from  $s$  to  $t$  in the residual graph, and therefore, the flow that we're at is a Max-flow.

Now we want to do a similar procedure for the feasible flow.

- So here, we're going to start at this feasible flow  $f$  that we found.  
Instead of starting at the zero flow as we did for the Max-flow algorithms, here we're going to start with a feasible flow,
- and then we build the residual graph and we try to augment this flow.
- So we'd look for a path from  $s$  to  $t$  in this residual graph and
- augment along it if it exists and if none exists, we'll know that we're at a maximum size feasible flow.

Now the only difference from before is that this residual graph is slightly different - the capacity of this edge from  $v$  to  $w$  in the residual graph for this current flow  $f$ :

- for forward edges, it's defined to be the leftover capacity - So it's the capacity minus the current flow.
- Now for reverse edges - so, if there's an edge from  $w$  to  $v$  in the original graph, then we add the edge from  $v$  to  $w$  if there is flow from  $w$  to  $v$ . So normally in the residual graph - the capacity of this edge from  $v$  to  $w$  - this reverse edge - will be the flow along this edge.

But now, we have the constraint that the flow can never go below the demand. So it can't decrease the flow below the demand. So the amount we can decrease the flow along this edge from  $w$  to  $v$  is by how much the flow exceeds the demand. So we set the capacity of this reverse edge to be the flow along this edge minus the demand, and finally its zero otherwise.

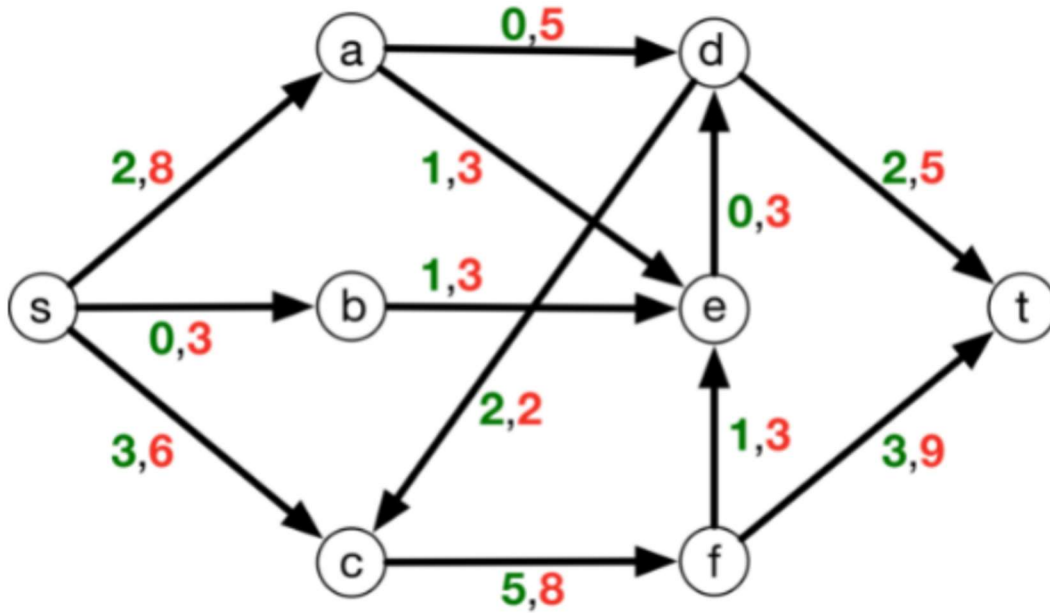
So, to find a max feasible flow, we run our Max-flow algorithm with the following changes:

- Instead of starting at zero flow, we start at a feasible flow,
- and when we're building the residual graph, the reverse edges, their capacities are not defined as the flow along the edge, but as how much the flow exceeds the demand.

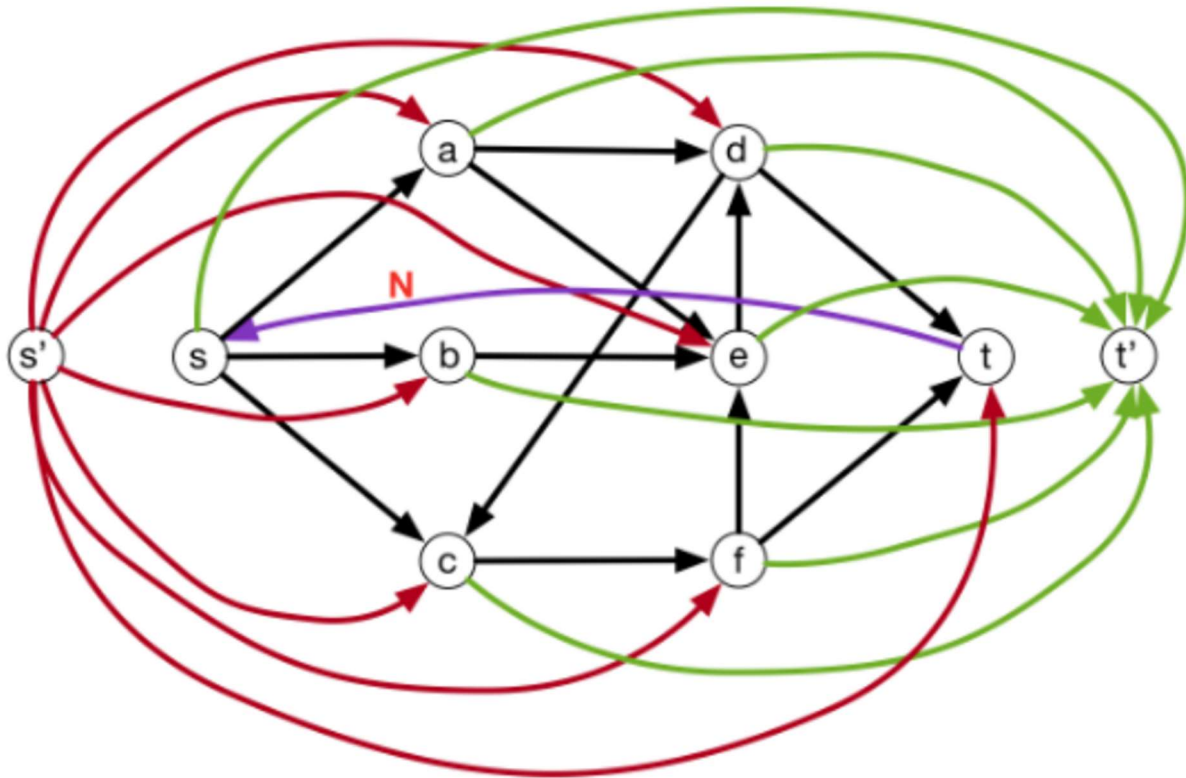
\*\*\*

(1<sup>st</sup> Slide 15) Reduction Example

MF5: Max-flow Generalization: Reduction Example



Above is an example input graph  $G$  to the Feasible Flow problem. Below is the graph  $G'$  as defined in the reduction to the Max-flow problem. The edge  $t$  to  $s$  (purple) has infinite capacity which is indicated in the figure by  $N$ .



Specify the capacities in the above graph  $G'$

Question 1:  $(s', a)$

Question 2:  $(s', b)$

Question 3:  $(s', c)$

Question 4:  $(s', d)$

Question 5:  $(s', e)$

Question 6:  $(s', f)$

Question 7:  $(s', t)$

Question 8:  $(a, t')$

Question 9:  $(b, t')$

Question 10:  $(c, t')$

Question 11:  $(d, t')$

Question 12:  $(e, t')$

Question 13:  $(f, t')$

Question 14:  $(f, t')$

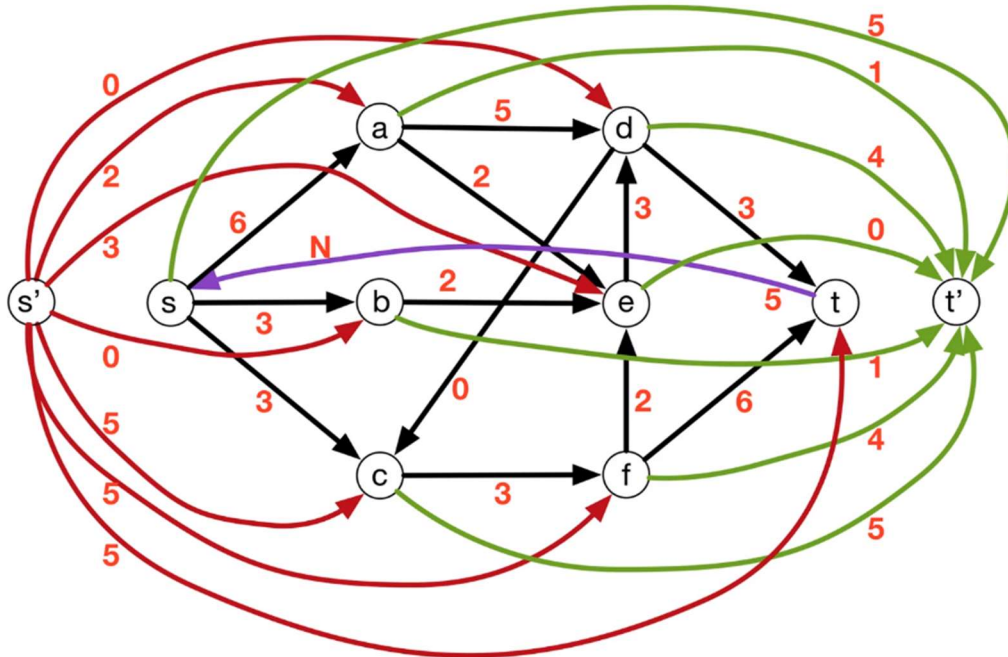
Question 15: Is there a saturating flow in this graph?

Question 16: Is there a feasible flow in the original graph  $G$ ?

\*\*\*

(2<sup>nd</sup> Slide 15) Reduction Example (Answer)

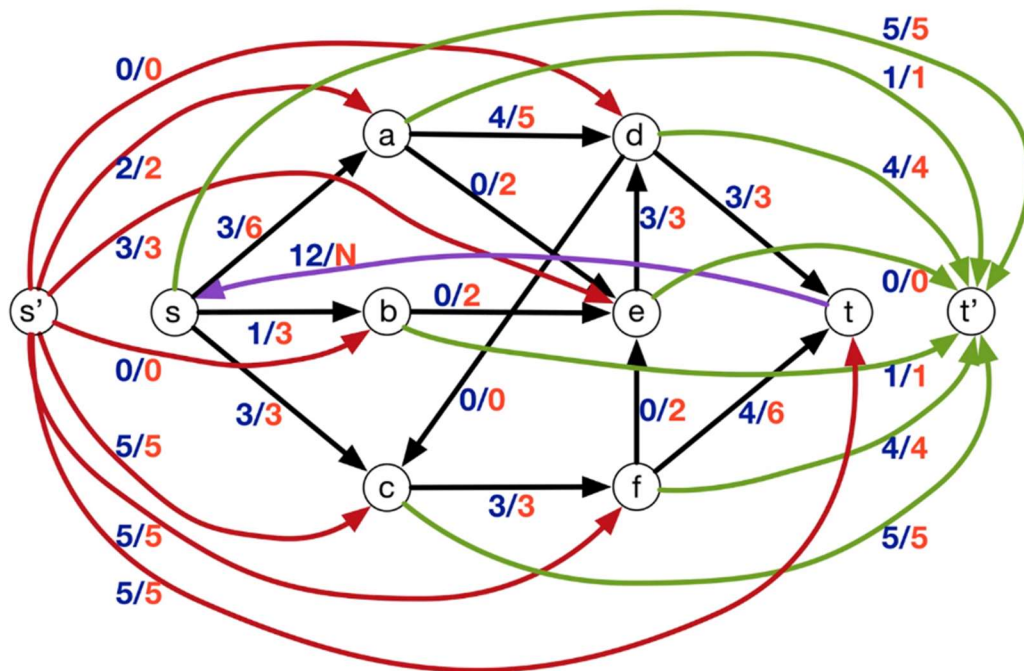
Solution



The max-flow graph  $G'$  with capacities shown.

Solution

The below figure shows a saturating flow in the graph  $G'$ .



### Solution

The below figure shows a feasible flow in the original graph G.

