CIS 575. Introduction to Algorithm Analysis
Material for April 22, 2024

Finding Maximal Flow

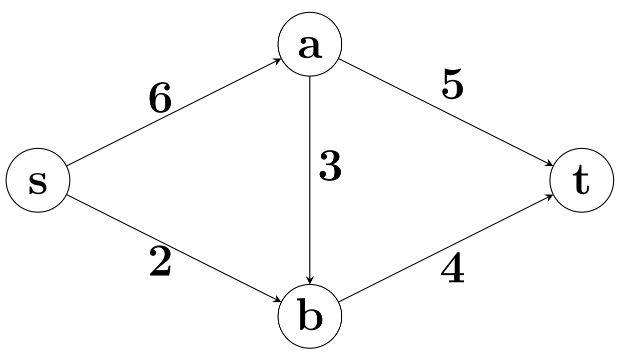
Torben Amtoft

The topic of this note is covered in *Cormen*'s Section 24.2.

1 Initial Approach for Maximum Flow

To find flows in a network, a key notion is the one of an **augmenting path** which is an acyclic path from the source to the sink.

For our previous example



there are three augmenting paths: sat, $sabt$, and sbt.

If a network has no augmenting paths then it obviously cannot have any flow except for the trivial that is zero on all edges. But if an augmenting path exists, we may recursively build a flow:

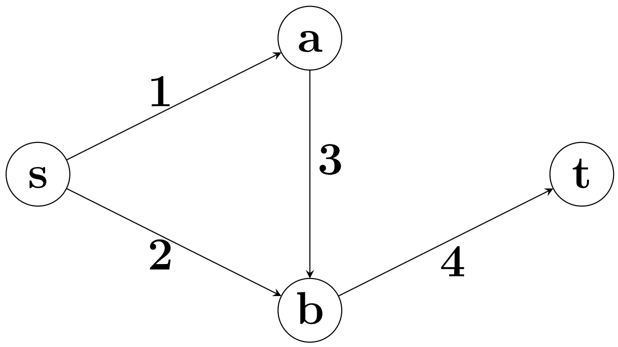
1. let π be an augmenting path, and let m be the minimum capacity on π
2. for each edge in π, subtract m from the capacity, removing edges whose capacity is now zero
3. for the resulting network, recursively build a flow F
4. return the flow F, after adding m to all edges in π.

Since step 2 will remove at least one edge, the recursion will eventually terminate.

Success story Let us try this approach for our example network. We need to choose an augmenting path: assume that we choose *sat* whose minimum capacity is 5; thus we collect a flow F_1 given by

$$F_1(sa) = F_1(at) = 5 \tag{1}$$

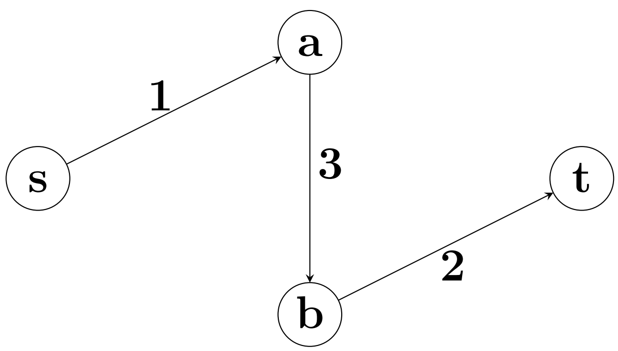
while removing the edge from a to t, and reducing the capacity of the edge from s to a. The resulting network is



where as augmenting path we may choose *sbt* whose minimum capacity is 2; hence we collect a flow F_2 given by

$$F_2(sb) = F_2(bt) = 2 \tag{2}$$

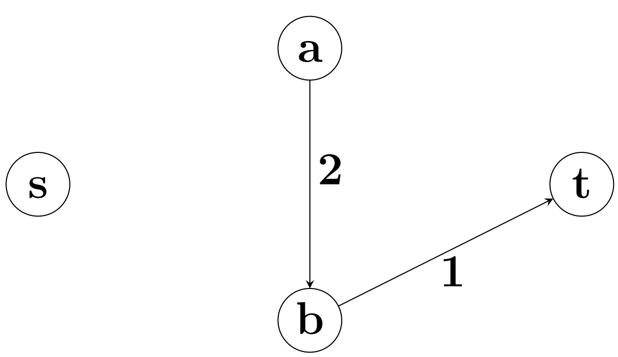
and next consider the network



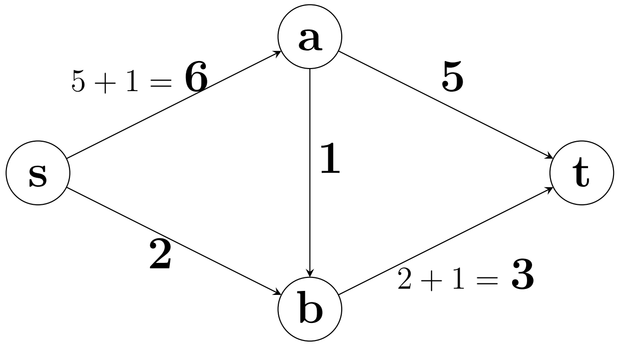
which has only one augmenting path, *sabt* with minimum capacity 1; we thus collect the flow F_3 given by

$$F_3(sa) = F_3(ab) = F_3(bt) = 1 \tag{3}$$

and consider the network

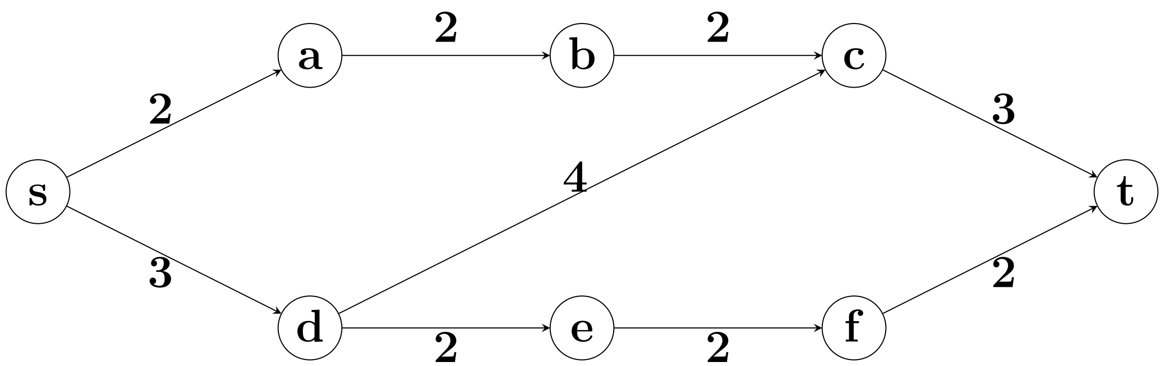


for which no augmenting path exists. Hence we are done, having produced a flow which can be found by adding the flows from (1), (2) and (3); that flow can be depicted as

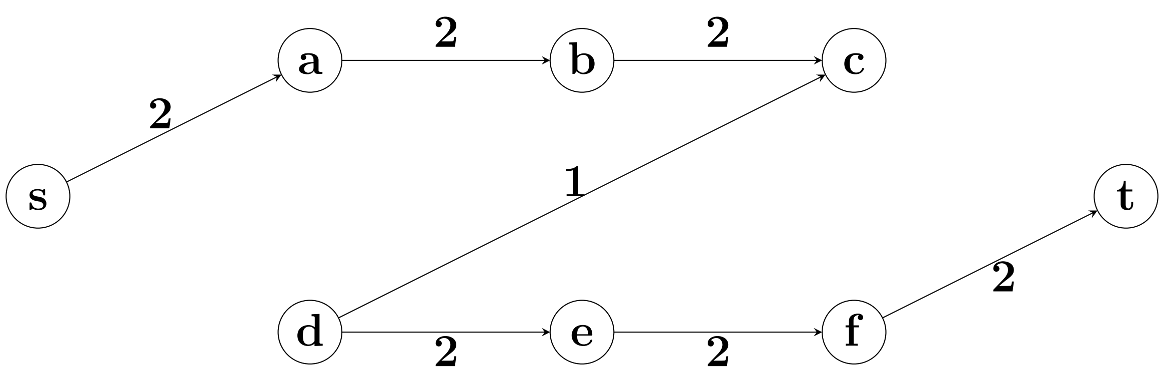


which is indeed, cf. the previous note, a maximum flow for the given network.

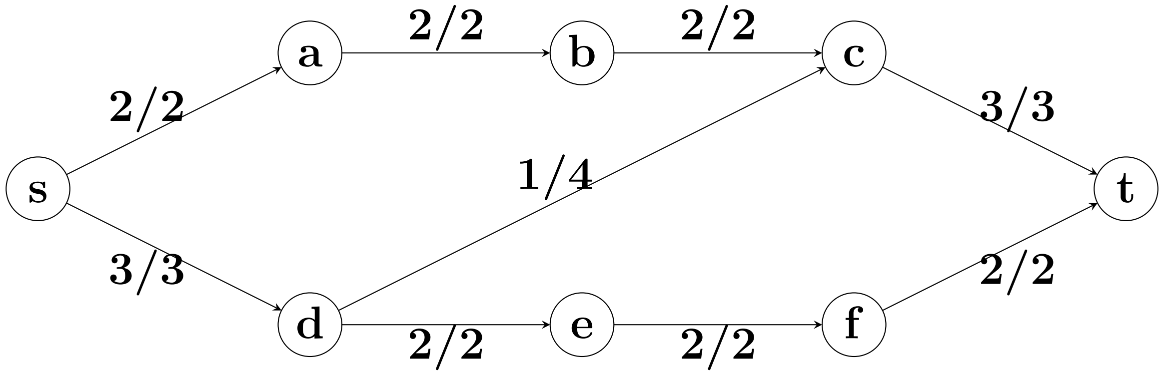
Failure story Unfortunately, in general there is **no guarantee** that the proposed approach will produce a **maximum** flow. To see this, consider the network



If we among the augmenting paths choose the shortest (the one with the fewest edges), which is *sdct* with minimum capacity 3, we must next consider the network



which has *no* augmenting path. Thus the proposed approach will end up producing a flow with value 3, even though the network has a flow with value 5:



We see that we should have given the edge from d to c a flow of only 1, rather than 3.

In the next note, we shall modify our approach so as to be able to (partially) *undo* flow assignments.