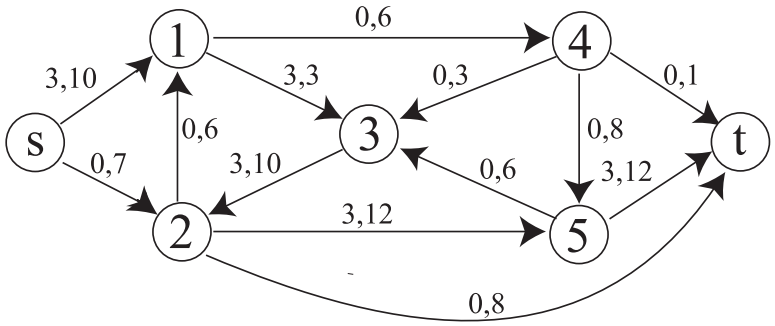
Solved exercises for the course of

Foundations of Operations Research

Roberto Cordone

Maximum flow

Consider the network given in the following picture: the network is partially loaded by a feasible flow; the first number on each arc provides this flow, while the second number provides the capacity of the arc.



Is the given flow maximum or not? Why?

If not, find the maximum flow.

Solution

The flow is not maximum because no $(s-t)$-cut fully consists of saturated forward arcs and empty backward arcs. We will solve the problem with three variants of the augmenting path method. All of them work by finding an augmenting path and routing through this path the maximum feasible amount of additional flow. The three variants differ by the procedure used to find the augmenting path:

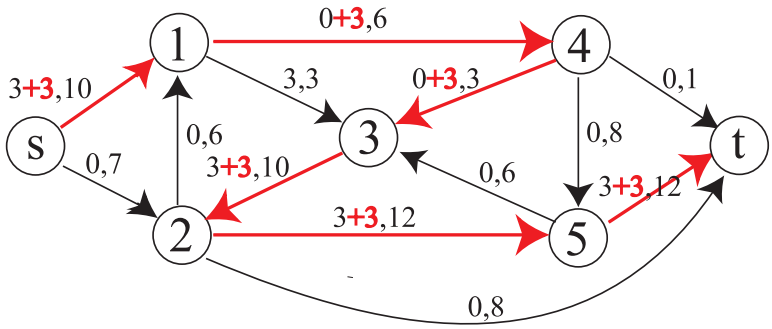
1. the first variant (Ford-Fulkerson's algorithm) finds the path with a depth-first visit of the graph from s;
2. the second variant computes the augmenting path that guarantees the maximum increase of flow;
3. the third variant (Edmonds-Karp's algorithm) finds the augmenting path with the minimum number of arcs, through a breadth-first visit of the graph from s.

Depth-first visit

At each step, the algorithm explores the children of the current node and moves to the first node not yet visited. If all children have already been visited, it backtracks to the father node and explores the following child of the father. By convention, we will explore the children of a node in increasing index order.

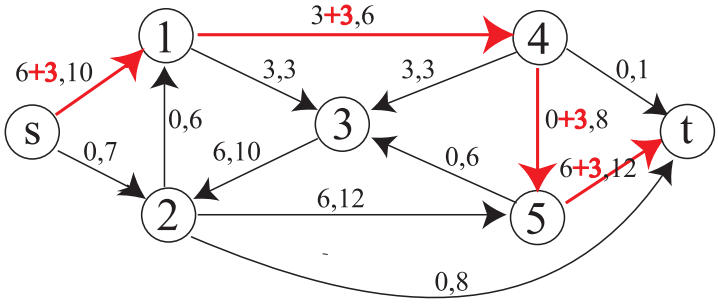
In detail:

- the first node reachable from node s is node 1, and the maximum additional flow that can be transferred is $\delta := \bar{k}_{s1} = k_{s1} - x_{s1} = 10 - 3 = 7$;
- from node 1, we can reach neither node 2 (arc $(1,2)$ does not exist and the backward arc $(2,1)$ is empty) nor node 3 (arc $(1,3)$ is saturated and there is no backward arc), so we consider node 4, to which an additional flow equal to $\delta := \min\left(\delta, \bar{k}_{14}\right) = 6$ can be transferred;
- from node 4 we cannot reach node 1 (the backward arc $(4,1)$ is empty) and node 2; we can reach node 3, and transfer to it a flow equal to $\delta := \min\left(\delta, \bar{k}_{43}\right) = 3$;
- from node 3 we can reach node 2, and transfer to it a flow equal to $\delta := \min\left(\delta, \bar{k}_{32}\right) = 3$
- from node 2 we can reach node 5, and transfer to it a flow equal to $\delta := \min\left(\delta, \bar{k}_{25}\right) = 3$
- from node 5 we can reach node t, and transfer to it a flow equal to $\delta := \min\left(\delta, \bar{k}_{5t}\right) = 3$

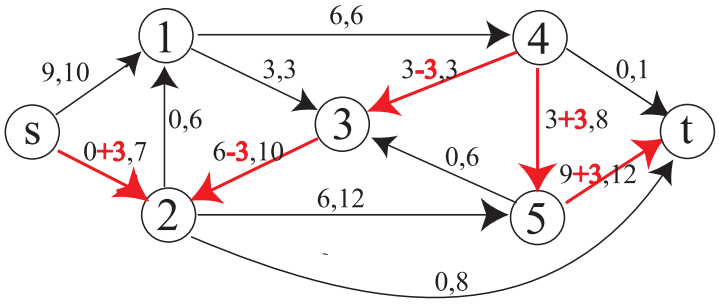


The other augmenting paths are:

1. s-1-4-5-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s1}, \bar{k}_{14}, \bar{k}_{45}, \bar{k}_{5t}\right) = 3$

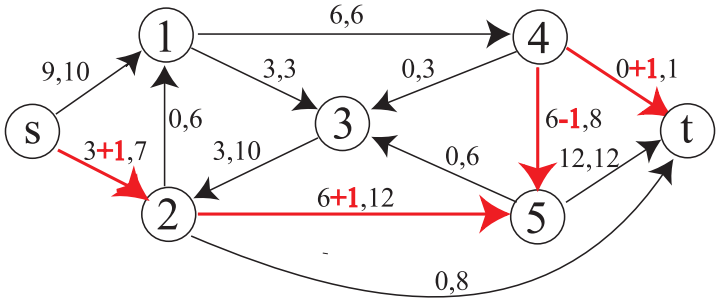


2. Now node 1 can be reached, but we cannot proceed in any direction from it, so that we *back-track* and try with node 2. In the end, we obtain the augmenting path s-2-3-4-5-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s2}, x_{32}, x_{43}, \bar{k}_{45}, \bar{k}_{5t}\right) = 3$.

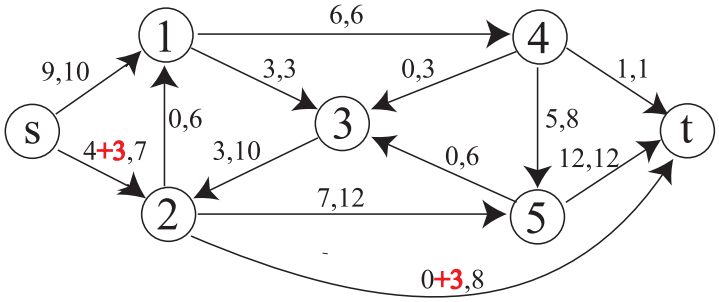


The augmenting path does not seem very smart. Indeed, we have found it with the depth-first search, always starting with the child of minimum index.

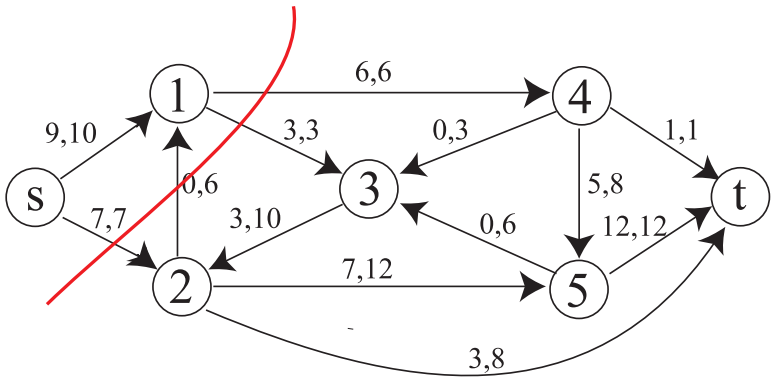
3. The path s-2-3-1 ha no way out. The first augmenting path found is s-2-5-4-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s2}, \bar{k}_{25}, x_{45}, \bar{k}_{4t}\right) = 1$.



4. s-2-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s2}, \bar{k}_{2t}\right) = 3$



The overall flow is equal to $\phi = 16$ and is distributed as in the following picture.



The picture also shows the minimum capacity cut ($k_{\{s,1\}} = k_{14} + k_{13} + k_{s2} = 16$), which proves the optimality of the flow, given that all outgoing arcs are saturated and all ingoing arcs are empty. The cut is induced by the subset of nodes which can be reached from the source s: node 1 can be reached because arc $(s, 1)$ is not yet saturated; the other nodes cannot be reached.

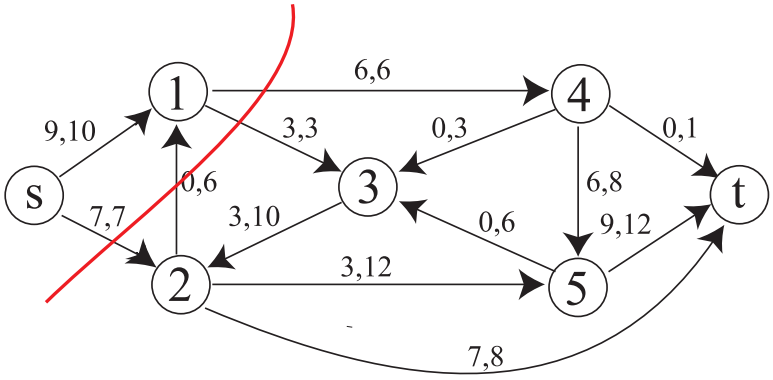
Maximum feasible flow increase

This variant determines at each step the path that allows to transfer the maximum amount of additional flow to the sink. This search can be performed with an algorithm very similar to Dijkstra's algorithm for the shortest path problem: the label of each node represents the maximum amount of flow that can be driven to this node along the already explored paths. This leads to two simple adaptations of the algorithm:

- instead of marking the node with minimum label, we mark the node with the maximum label;
- instead of updating a label with the minimum between the current label and the sum of the label of the last marked node plus the cost of the connecting arc, we update it with the maximum between the current label and the minimum between the label of the last marked node and the residual capacity of the arc (if it is a forward arc) or the flow along the arc (if it is a backward arc).

The algorithm just requires two augmenting paths:

1. s-2-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s2}, \bar{k}_{2t},\right) = 7$
2. s-1-4-5-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s1}, \bar{k}_{14},\right) = 6$



Breadth-first visit

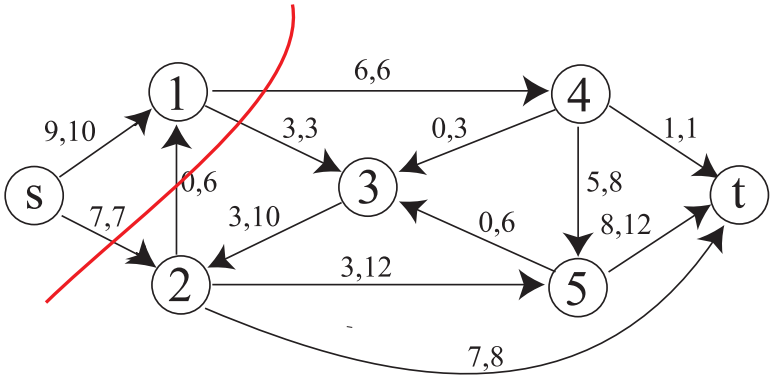
This variant determines at each step the augmenting path with the minimum number of arcs. One can determine such a path with Dijkstra's algorithm assuming that each arc has a unitary cost,but the most efficient way is to solve this special problem is to use a breadth-first visit of the graph, as follows:

1. mark with label 1 all the nodes directly reachable from the source s, through nonsaturated forward arcs and nonempty backward arcs;
2. for each node with label 1, mark with label 2 all the reachable nodes not yet labelled which satisfy the condition stated above;
3. ...

As soon as sink t is labelled, the algorithm terminates.

This algorithm requires three augmenting paths:

1. s-2-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s2}, \bar{k}_{2t},\right) = 7$;
2. s-1-4-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s1}, \bar{k}_{14},\right) = 1$;
3. s-1-4-5-t with a flow increase equal to $\delta = \min\left(\bar{k}_{s1}, \bar{k}_{14},\right) = 5$.



The second and third variant guarantee a polynomial computational time. Actually, the third is the best one in the worst case. In this example, the second variant required less iterations, but this example is not the worst possible case. fs