

CO: Exercise 3.46

A feasible preflow and a valid labelling is obtained, shown in figure 1. The numbers on arcs are the capacity and the flow, respectively.

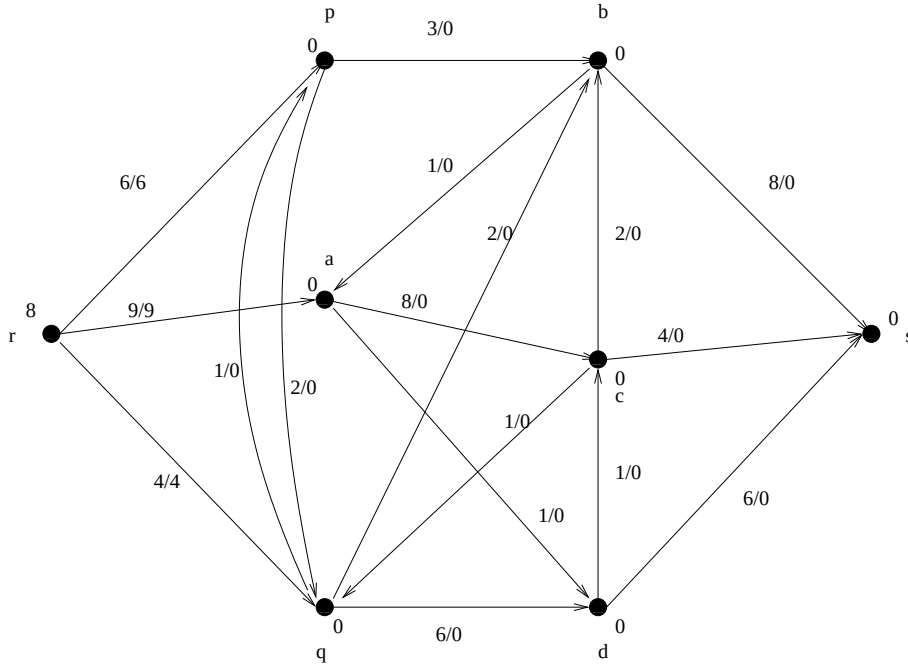


Figure 1: Feasible preflow and valid labelling used initially.

Nodes p , a and q are active since they have a positive net flow. Also they have the same label - 0, so anyone of them can be picked. Pick p arbitrarily and “process” it. An arc vw is admissible if there exists an arc in the *auxiliary graph* from v to w and $d(v) = d(w) + 1$. No such arc exists, hence we relabel p to $d(p) = \min(d(w) + 1 : w \in \{b, q, r\}) = \min(0 + 1, 0 + 1, 8 + 1) = 1$. In principle we need to draw the auxiliary graph in order to figure out which arcs leaves p , but in practice we can simply figure out “where flow can be pushed from p given current flow”. In this case it can be pushed to b and q since these arcs are empty and flow can also be pushed back to r since the flow on arc rp is larger than 0.

Given the new label of p arc pb and pq are admissible so we push flow along both of the arcs - 3 and 2 respectively. The graph is shown in figure 2.

p is still active since the net flow is positive ($=1$), but no admissible arc exists, so we relabel p . Only one arc leaves p in the auxiliary graph, namely arc pr . Hence the new label is $\min(8 + 1) = 9$. Process p again since it is the active node with highest label. The only admissible arc is pr , hence the remaining flow is pushed back to r , i.e. the flow on arc rp is reduced to 5. The graph is shown in figure 3.

Now a , b and q are active and again one can be selected arbitrarily and processed. This is continued until the graph in figure 4 is obtained.

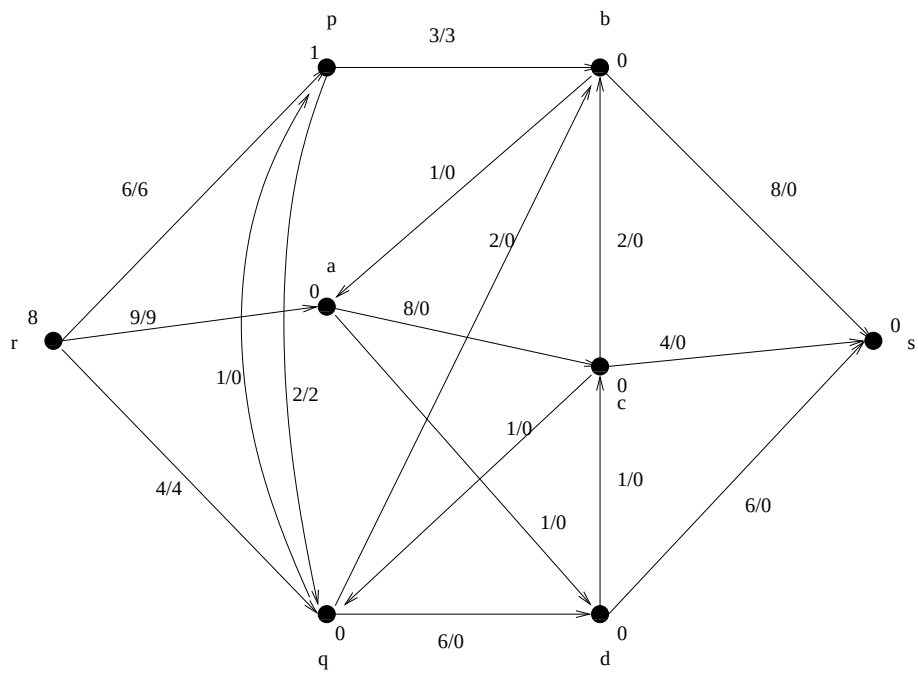


Figure 2: Graph # 2.

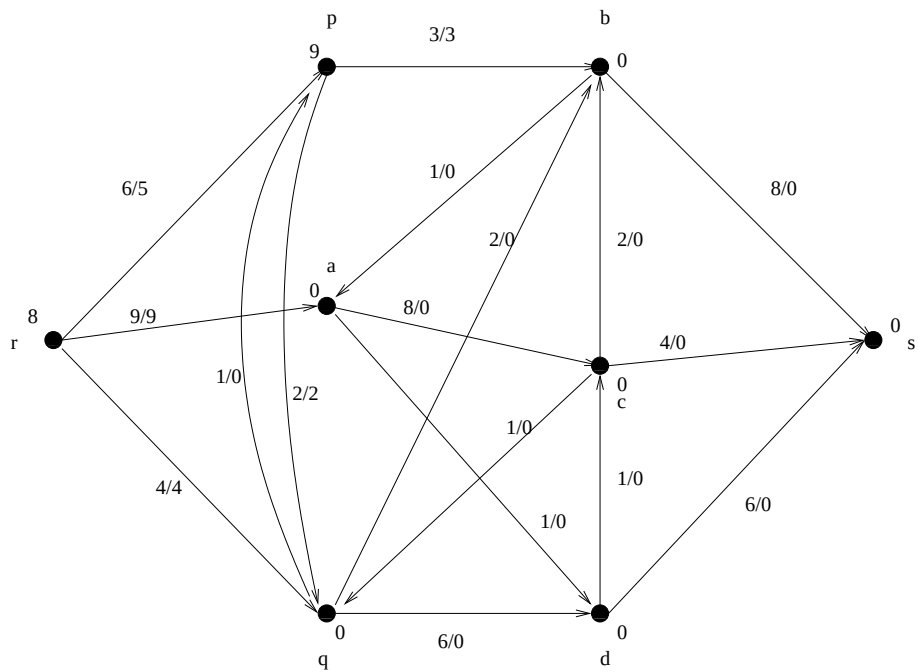


Figure 3: Graph # 3.

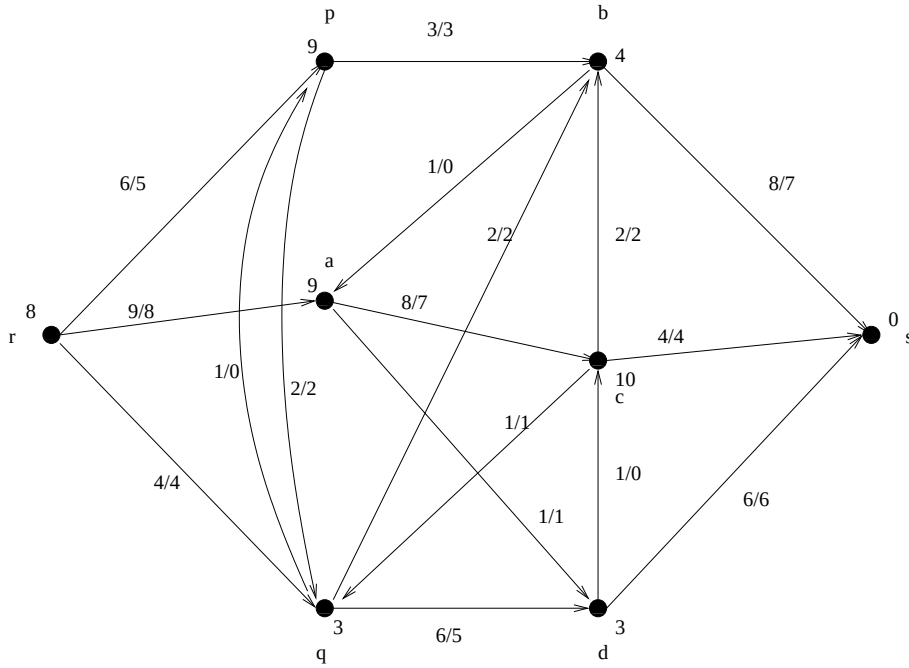


Figure 4: Last graph.

CO: Exercise 3.48

In the following we refer to the *auxiliary graph* and remember that $d_x()$ is defined for the auxiliary graph. Notice that if s can be reached from v then $d_x(v, s) < n$ (since there are n nodes), so in that case $d(v) = d_x(v, s)$. Otherwise $d(v) = n + d_x(v, r)$.

Equation 3.14 page 63 gives the definition of a valid labelling, let's check if this holds. $d_x(s, s) = d_x(r, r) = 0$ so, as required, $d(s) = 0$ and $d(r) = n$ since s can never be reached from r . This follows from the fact (proposition 3.22 page 64) that given a feasible preflow and the existence of a valid labelling for x , a (r, s) cut exists such that all forward edges are filled and all backward edges are empty. This means that in the auxiliary graph, no edges cross the cut and s cannot be reached from r .

Additionally, in order to be a valid labelling, all arcs vw have to fulfill $d(v) \leq d(w) + 1$. To check that this is in fact the case, 4 possibilities exist:

v and w can both reach s. In that case $d(v)$ and $d(w)$ are shortest paths to s . $d(v)$ is then at most as high as $d(w) + 1$, since a shortest path can use vw and the shortest path from w to s . I.e. $d(v) \leq d(w) + 1$.

Neither v nor w can reach s. In that case $d(v) - n$ and $d(w) - n$ are shortest paths to r . I.e. $d(v) - n \leq d(w) - n + 1 \Leftrightarrow d(v) \leq d(w) + 1$.

v can reach s, w cannot. In that case $d(v) < n$ and $d(w) \geq n$. Hence $d(v) < n \leq d(w) < d(w) + 1$.

w can reach s, v cannot. This cannot be the case, since there is an arc from v to w .

Hence the described labelling is valid.

$d(v)$ is in general a lower bound on $d_x(v, s)$ and $d(v) - n$ is a lower bound on $d_x(v, r)$ (see top of page 65 including Lemma 3.24). The described $d(v)$ is the best lower bound.

Calculating $d(v)$ requires knowing shortest-paths (each arc has length 1) from any node v to r and any node v to s , i.e. $O(2n)$ shortest path calculations in total. Instead all arcs can be “turned the other way” and the shortest-path from r to any node v and the shortest-path from s to any node v can be found. This can be done by only 2 shortest-path runs. Dijkstra’s algorithm requires $O(m \log n)$ and 2 runs of this have the same complexity - $O(m \log n)$.