

CO: Exercise 3.2

It is possible to start with the flow of value 0 and then proceed, but let's be a bit more clever. Assume we initially start with the flow corresponding to the 3 disjoint paths r, p, b, s ; r, a, c, s and r, q, d, s . The flow on each path is maximized and the flow shown in figure 1 is then obtained. The numbers on arcs are the flow and the capacity, respectively.

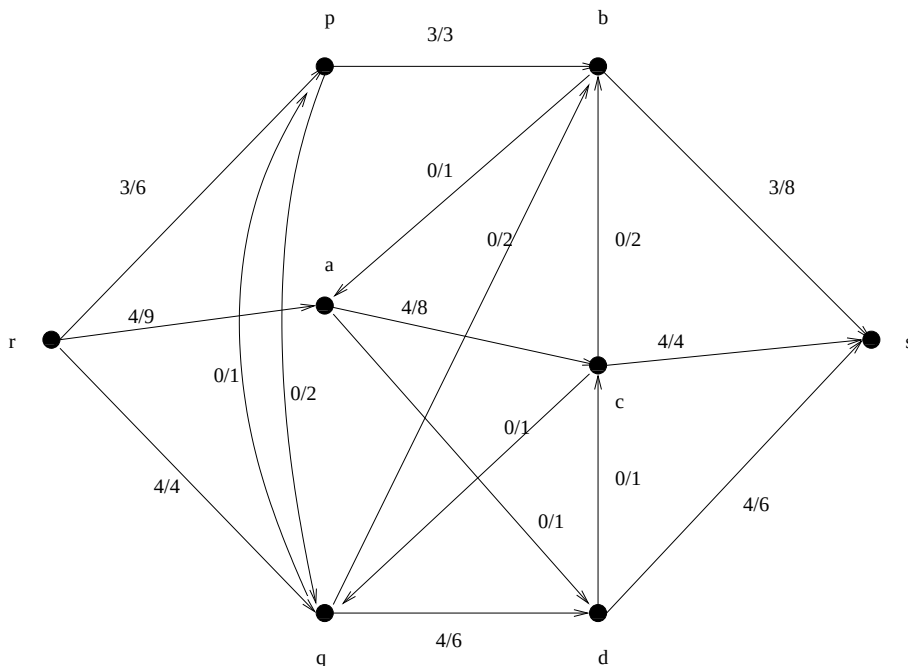


Figure 1: Initial flow used.

Each arc in the original graph gives rise to two arcs (possibly one with value 0, and so this is not shown) in the auxiliary graph. For an arc vw with capacity u_{vw} and flow x_{vw} , the auxiliary graph contains an arc in the same direction with the remaining capacity (i.e. an arc vw with capacity $u_{vw} - x_{vw}$) and an arc in the reverse direction with capacity equal to the flow (i.e. an arc wv with capacity x_{vw}). The auxiliary graph for the flow in figure 1 is shown in figure 2.

Now the task is to find an augmenting path given the flow. This corresponds to finding any path between r and s in the auxiliary graph. If no such path exists, the flow is optimal. In the following, all iterations will not be shown - more augmenting paths will be found and incorporated into the flow at the same time. In figure 2 it can be seen that the path r, a, c, b, s can be augmented by 2 and the path r, p, q, d, s can be augmented by 2. The resulting graph is shown in figure 3 and the corresponding auxiliary graph is shown in figure 4.

In this case two augmenting paths (each with a value of 1) can be found: r, a, c, q, b, s and r, a, d, q, b, s . Augmenting yields the flow shown in figure 5 and the auxiliary graph shown in figure 6.

No augmenting paths can be found in the auxiliary graph in figure 6. Hence the flow in figure 5 with a value of 17 is optimal. A minimum cut consists of all the nodes reachable from r in the

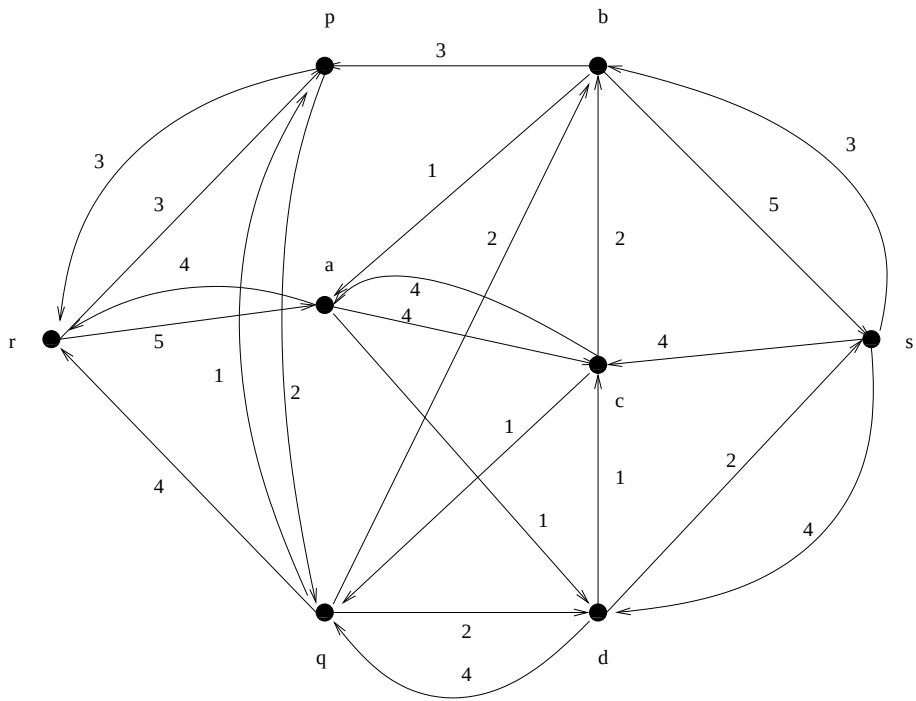


Figure 2: *Auxiliary graph #1.*

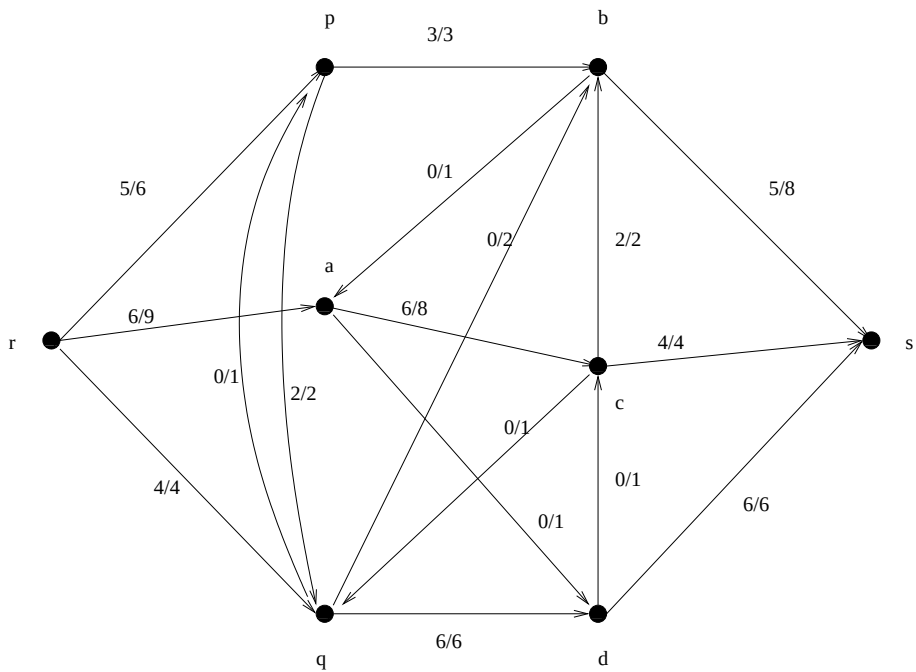


Figure 3: *Flow graph #2.*

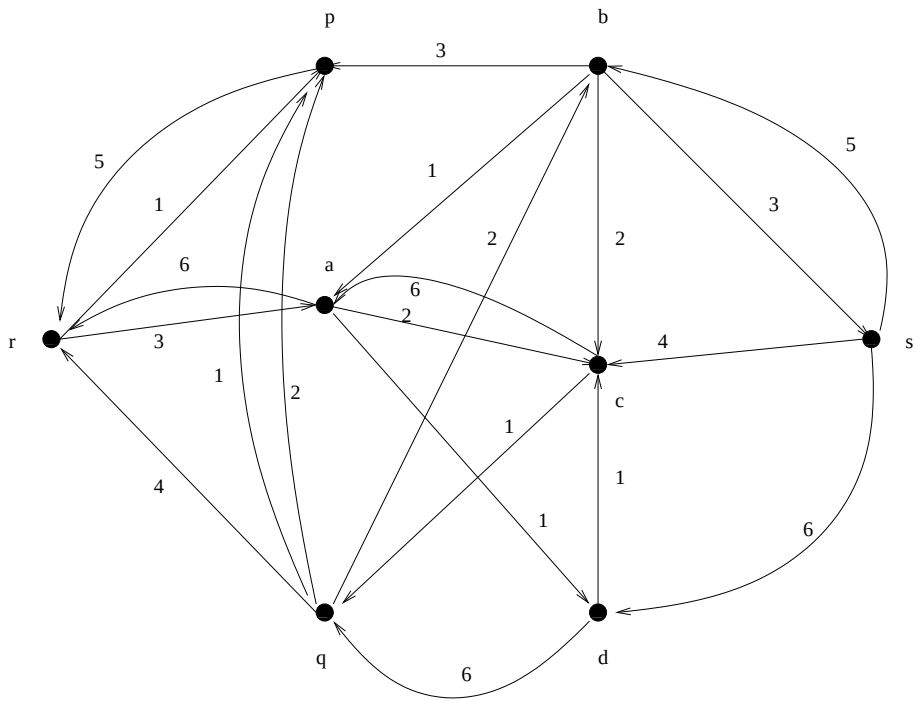


Figure 4: *Auxiliary graph #2.*

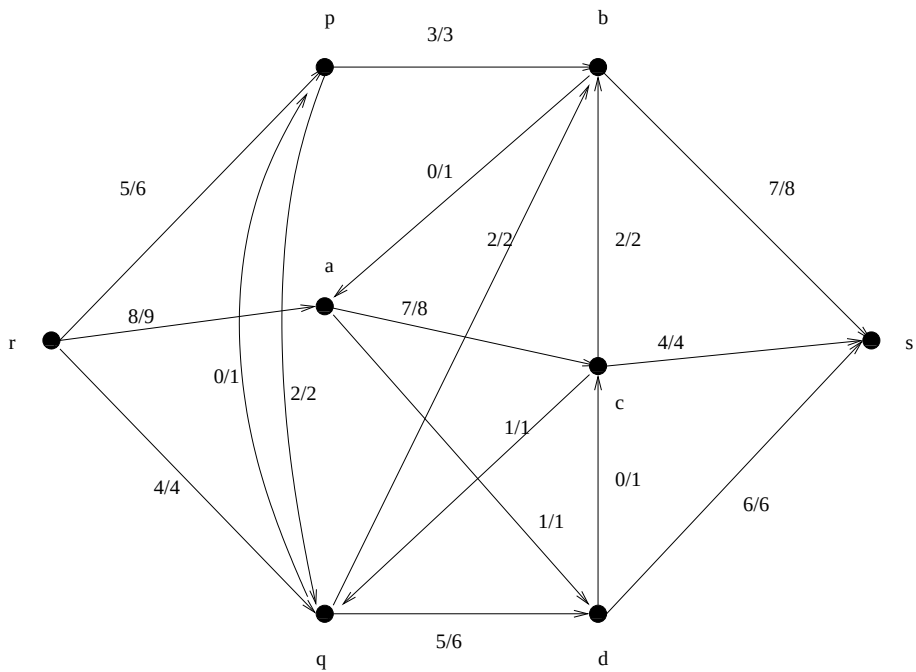


Figure 5: *Flow graph #3.*

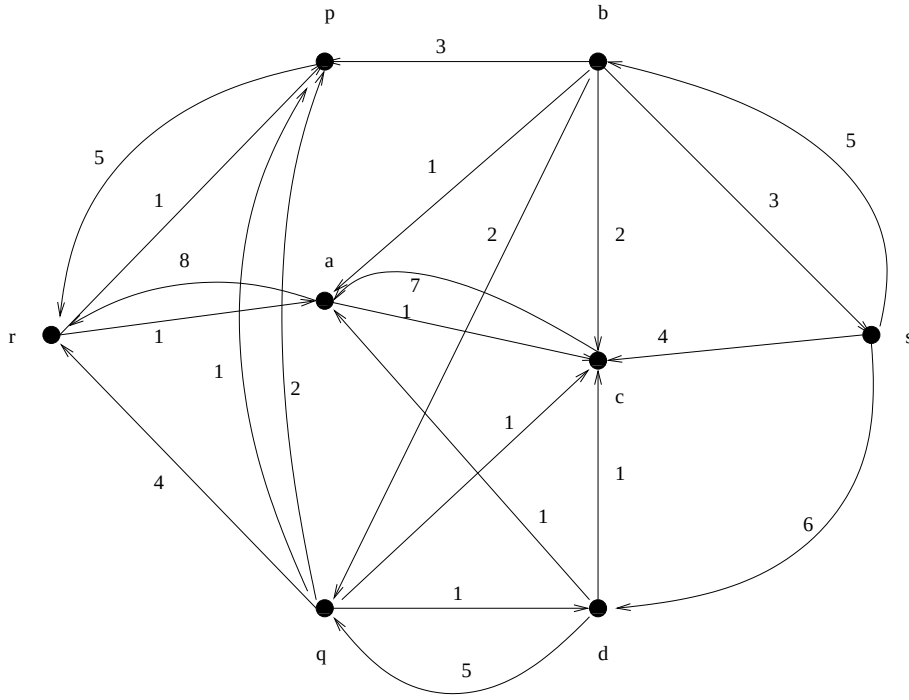


Figure 6: Auxiliary graph #3.

auxiliary graph, i.e. r, p, a, c with a value of 17. (Since finding minimum cut is dual of finding maximum flow. Counting gives 17 as well - try!)

CO: Exercise 3.7

The fact that $\delta(R_i)$ is a *minimum* cut means that for a max-flow x , all arcs in $\delta(R_i)$ are filled to capacity (i.e. $x_{vw} = u_{vw}$) and all arcs in $\delta(\overline{R_i})$ are empty (i.e. $x_{vw} = 0$). Also the reverse holds, i.e. if for a feasible flow all arcs of a cut have these properties, the cut is minimum.

Consider arcs in $\delta(R_1 \cap R_2)$. Three cases exists, either the arc goes to $R_1 \setminus (R_1 \cap R_2)$, $R_2 \setminus (R_1 \cap R_2)$ or $\overline{R_1 \cup R_2}$ (see figure 7). Assuming $\delta(R_1)$ and $\delta(R_2)$ are minimum cuts, then in all cases, the arcs are in either $\delta(R_1)$ or $\delta(R_2)$ (or both) and hence all arcs in $\delta(R_1 \cap R_2)$ are filled to capacity.

Likewise for a max-flow, all arcs in $\delta(\overline{R_1 \cap R_2})$ are empty, since they are all in a minimum cut. Given this (i.e. all out-going arcs are at their limit and in-going arcs are empty with respect to a feasible flow), then $\delta(R_1 \cap R_2)$ is a minimum cut.

The proof is similar for $\delta(R_1 \cup R_2)$.

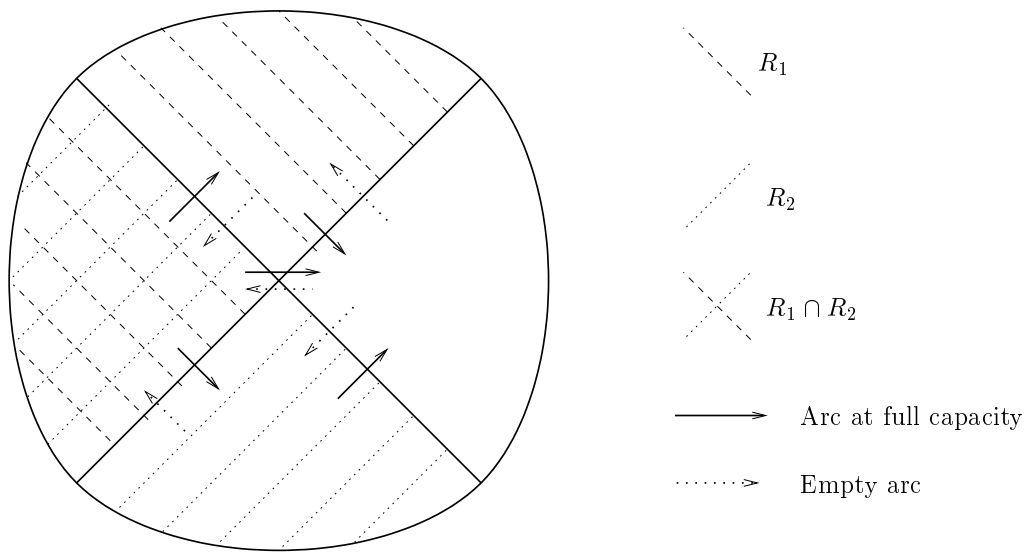


Figure 7: *Minimum cuts in a graph.*

CO: Exercise 3.16

This problem can be solved by modifying the problem and using a max-flow algorithm as described in CO page 47-49.

The following set of edges is a matching of size 8: $\{bH, cA, dC, fE, gF, hJ, iI, jD\}$.

The following set of nodes is an edge cover of size 8: $\{b, f, g, h, A, C, D, I\}$.

Since maximum matching and minimum cover are dual problems and both solutions have value 8, both solutions are optimal.