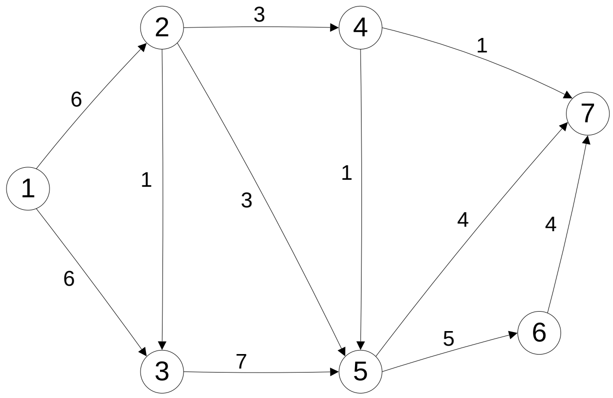
2.11 Maximum flow and minimum cut

Given the following network with capacities on the arcs



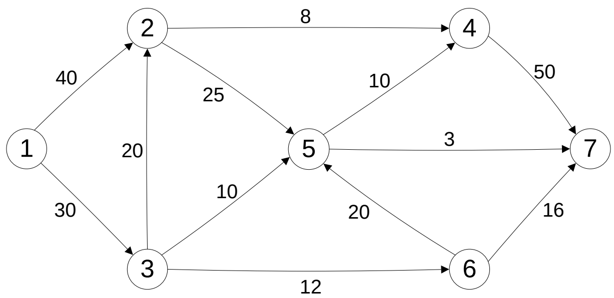
find a maximum flow from node 1 to 7 and the corresponding minimum cut.

2.12 Node capacities

Propose a way to deal with maximum flow problems with capacities on both nodes and arcs. Find a maximum flow for the network of the previous exercise, with a node capacity of 2 on node 6.

2.13 Maximum flow with a strictly positive initial flow

Given the following network with capacities on the arcs



find a maximum flow from node 1 to 7, starting from a feasible flow of value 10, obtained by sending 10 units on the path 1-3-6-5-4-7. Show the corresponding minimum cut.

2.14 Software house

A software house has to handle 3 projects, P_1, P_2, P_3, over the next 4 months. P_1 can only begin after month 1, and must be completed within month 3. P_2 and P_3 can begin at month 1, and must be completed, respectively, within month 4 and 2. The projects require, respectively, 8, 10, and 12 man-months. For each month, 8 engineers are available. Due to the internal structure of the company, at most 6 engineers can be work, at the same time, on the same project.

Determine whether it is possible to complete the projects within the time constraints. Describe how to reduce this problem to the problem of finding a maximum flow on an appropriate graph.

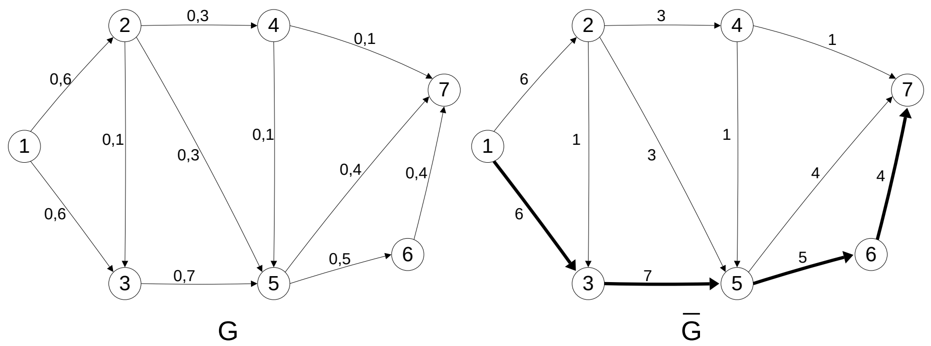
[*Hint*: can you find a feasible flow of value 30?]

Solution

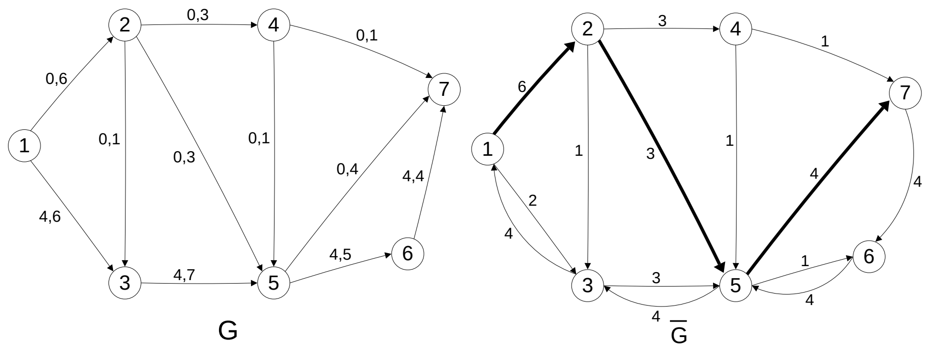
2.11 Maximum flow and minimum cut

We apply the Ford-Fulkerson algorithm. In the following figures, the network on the left reports the current feasible flow $\underline{x}$ in the graph G, indicating on each arc the quantity of product x_{ij} flowing through it, and the capacity k_{ij}. The network on the right indicates the incremental graph $\bar{G}$ for the current flow.

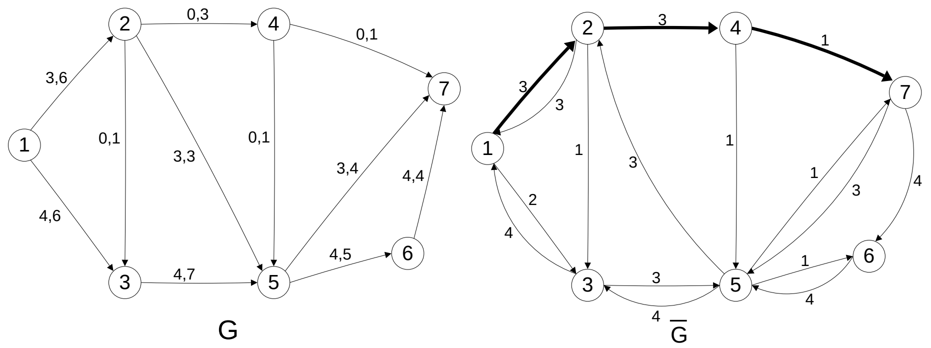
We start from the null flow $\underline{x} = \underline{0}$ of value $\varphi = 0$. Since all arcs are empty, $\bar{G}$ is equivalent to G.



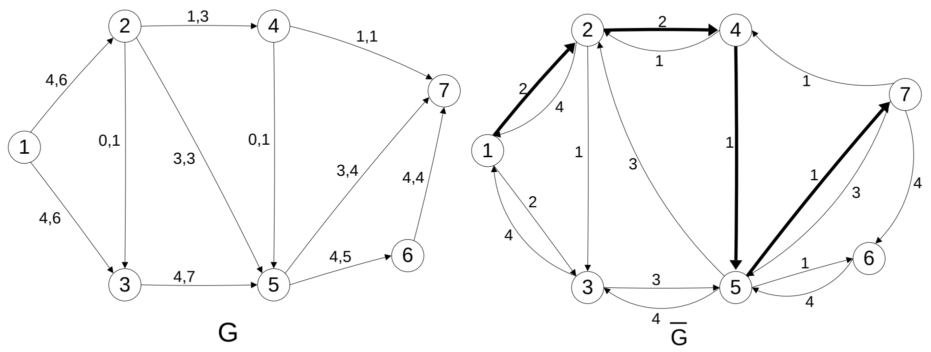
We sent $\delta = 4$ units on path 1-3-5-6-7. δ is given by arc $(6, 7)$, which has the smallest capacity k_{ij} of the path.



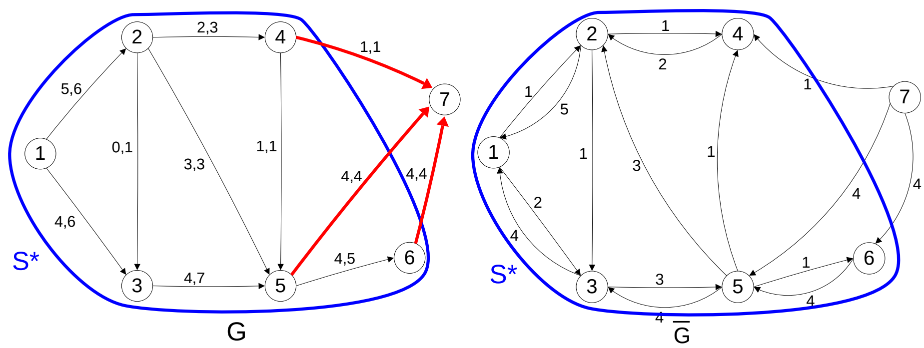
We sent $\delta = 3$ units on path 1-2-5-7. δ is given by arc $(2, 5)$, which has the smallest residual capacity $\bar{k}_{ij}$ of the path. We obtain a flow of value $\varphi = 7$.



We sent $\delta = 1$ units on path 1-2-4-7. δ is given by arc $(4, 7)$. We obtain a flow of value $\varphi = 8$.



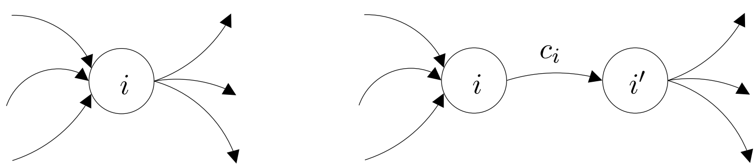
Se send $\delta = 1$ units on path 1-2-4-5-7. δ is given by arc $(4, 5)$ (or $(5, 7)$). We obtain a flow of value $\varphi = 9$.



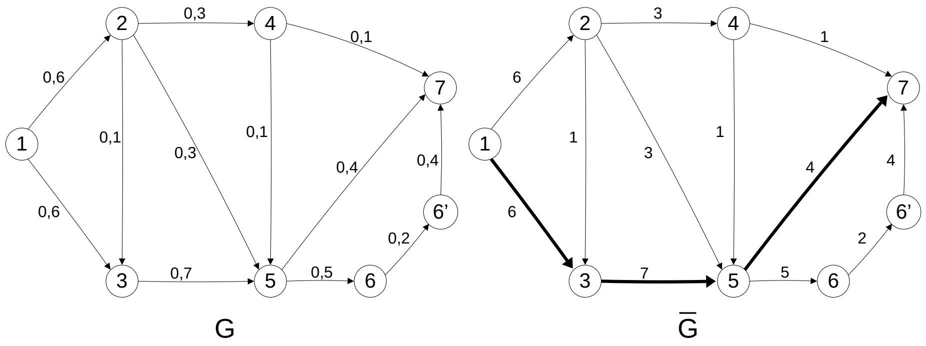
No augmenting paths can be found in $\bar{G}$ and the algorithm halts. The set $S^* = \{1, 2, 3, 4, 5, 6\}$, highlighted in blue, indicates the nodes that can be reached from the source 1 in $\bar{G}$. S^* induces the minimum cut corresponding to the flow $\underline{x}$, which is highlighted in red. Correctness can be checked by observing whether the value of the flow equals the total capacity of the cut. At any optimal solution, the two values must be equivalent.

2.12 **Node capacities**.

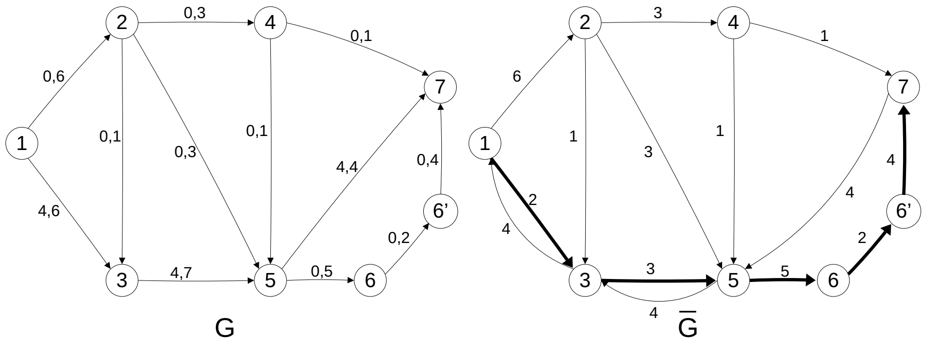
We substitute, for each node with a capacity, two auxiliary nodes, which are connected with an arc of capacity equivalent to that of the original node, as shown in the figure



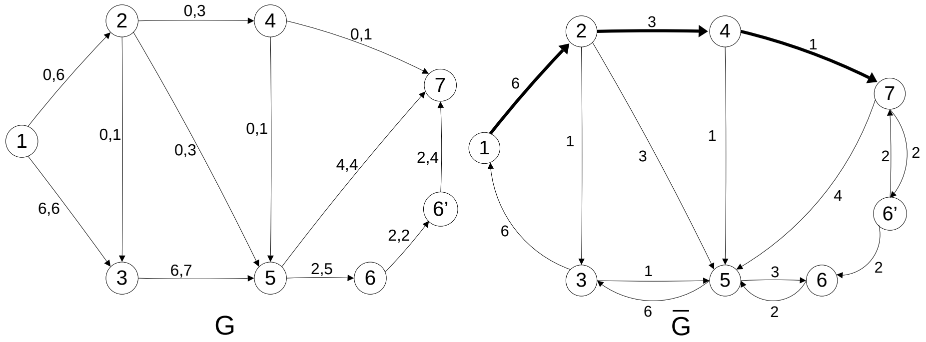
The network of exercise 2.11 is modified as follows.



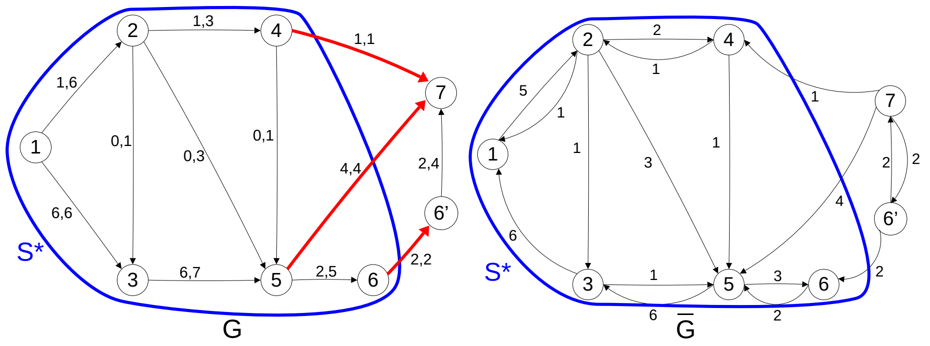
We sent $\delta = 4$ units, on path 1-3-5-7.



We sent $\delta = 2$ units, on path 1-3-5-6-6'-7.



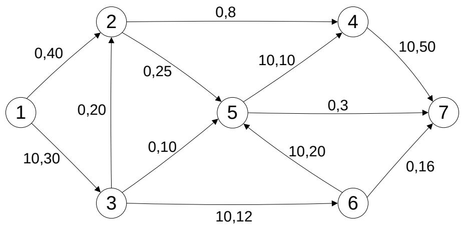
We sent $\delta = 1$ units, on path 1-2-4-7.



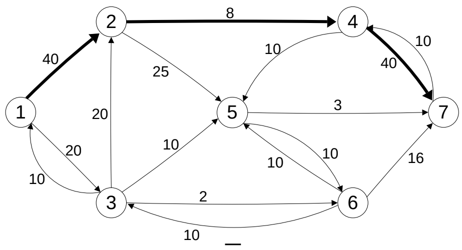
No more augmenting paths can be found. The minimum cut is highlighted in red, induced by $S^* = \{1, 2, 3, 4, 5, 6\}$. It has total capacity $k(S^*) = \varphi = 7$.

2.13 Maximum flow with a strictly positive initial flow

The original network and the incremental one, for the initial flow of value 10, are as follows

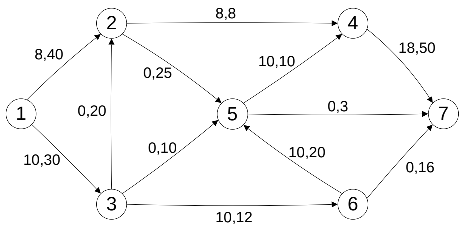


G

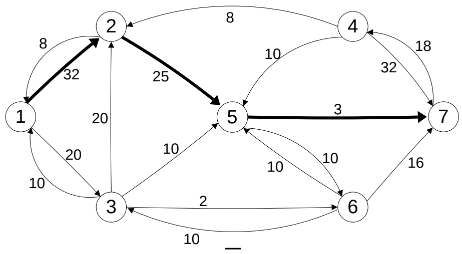


$\overline{G}$

We sent $\delta = 8$ units, on path 1-2-4-7.

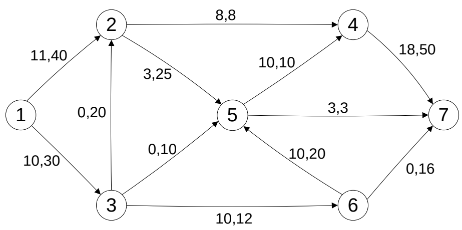


G

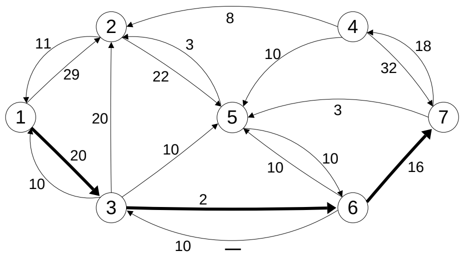


$\overline{G}$

We sent $\delta = 3$ units, on path 1-2-5-7.

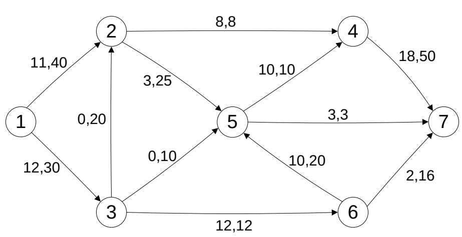


G

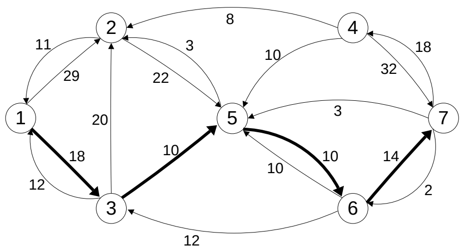


$\overline{G}$

We sent $\delta = 2$ units, on path 1-3-6-7.



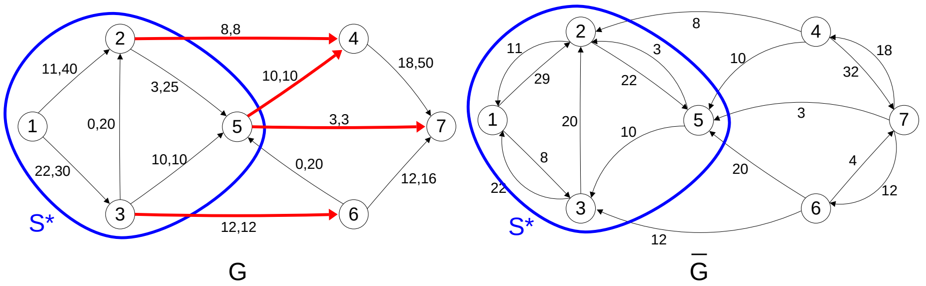
G



$\overline{G}$

Observe that all augmenting paths for this network use the backward arc $(5, 6)$. From the flow point of view, this amounts to unload arc $(6, 5)$, decreasing the amount of product flowing through it.

We sent $\delta = 10$ units, on path 1-3-5-6-7.

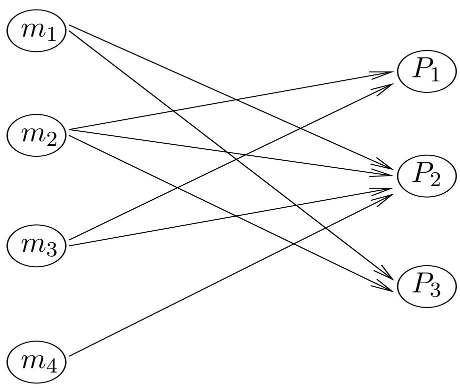


No more augmenting paths can be found. The minimum cut is highlighted in red, induced by $S^* = \{1, 2, 3, 5\}$. It has total capacity $k(S^*) = \varphi = 33$.

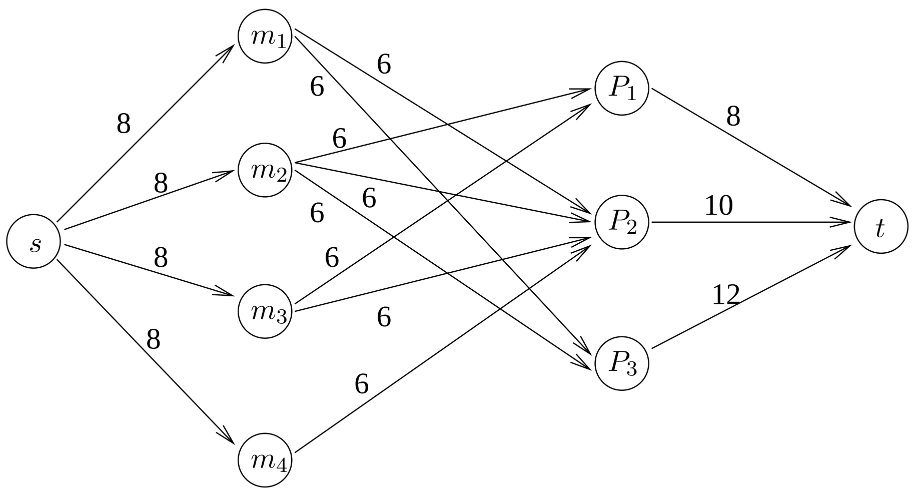
2.14 Software house

We build a product network where months and projects are represented by, respectively, month-nodes m_1, m_2, m_3, m_4 and project-nodes P_1, P_2, P_3.

Each (i, j) arc denotes the possibility of allocating man-hours of month i to project j. For instance, since project P_1 can only begin after month 1 and must be completed before month 3, only arcs outgoing from m_2, m_3 are incident in P_1.



We add to auxiliary nodes s, t, denoting the source and sink of the flow that represents the allocation of men-hours.



All arcs outgoing from s have capacity 8, equivalent to the number of available engineers per month. All arcs connecting month-nodes to project-nodes have capacity 6, as no more than 6 engineers can work on the same project in the same month. All arcs incident in t have capacity equivalent to the number of man-months needed to complete the project.

Since all capacities are integer, the maximum flow will be integer as well. To check whether all projects can be completed within the time limits, it suffices to check whether the network admits a feasible flow of value $8 + 10 + 12 = 30$.

As an exercise, find the maximum flow in the network and, if it has a value of 30, deduce the corresponding work plan.