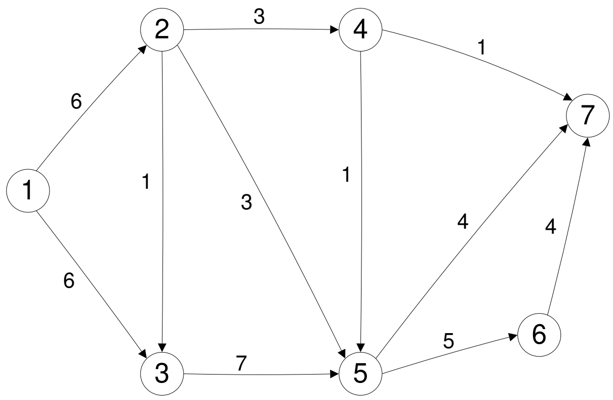
2.10 Maximum flow and minimum cut

Given the following network with capacities on the arcs



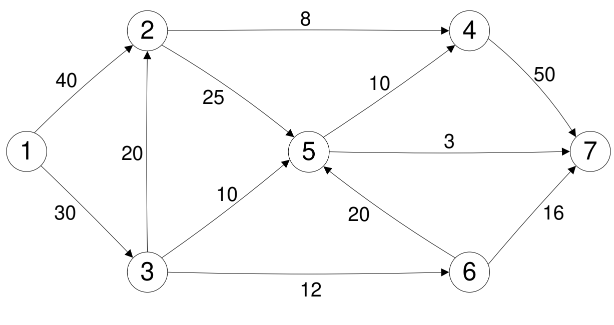
find a maximum (feasible) flow from node 1 to node 7, and determine a corresponding minimum (capacity) cut.

2.11 Maximum flow with node capacities

In maximum flow problems, how can we deal with capacities on both nodes and arcs? Find a maximum flow from node 1 to node 7 in the network of the previous exercise, with a node capacity of 2 on node 6.

2.12 Maximum flow with a strictly positive initial feasible flow

Given the following network with capacities on the arcs



find a maximum flow from node 1 to node 7, starting from the feasible flow of value 10 in which 10 units are sent along the path 1-3-6-5-4-7. Determine a corresponding minimum cut.

2.13 Indirect application of maximum flows

A software house has to handle 3 projects, P_1, P_2, P_3, over the next 4 months. The projects require, respectively, 8, 10, and 12 man-months. P_1 can only begin after month 1, and must be completed (at latest) by the end of month 3. P_2 and P_3 can begin from month 1, and must be completed, respectively, by the end of month 4 and 2, respectively. Each month, 8 engineers are available. Due to the internal structure of the company, no more than 6 engineers can work, at the same time, on the same project.

Determine whether it is possible to complete the three projects within the time constraints and, if it is possible, find a feasible workforce plan. Describe how this problem can be reduced to the problem of finding a maximum flow in an appropriate network.

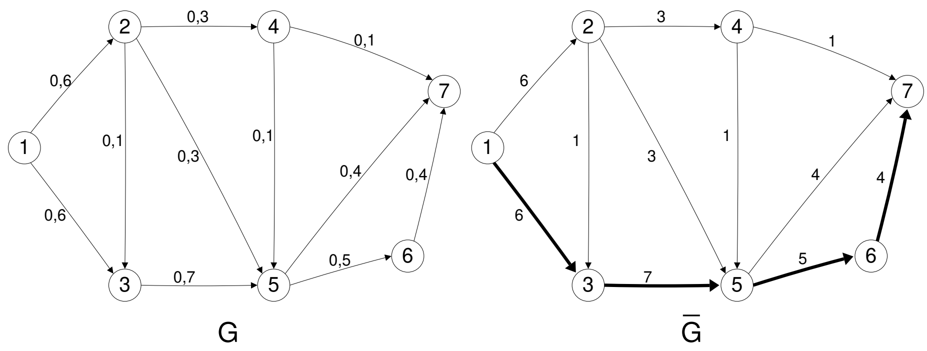
[*Hint*: Look for a feasible flow of value 30?]

SOLUTION

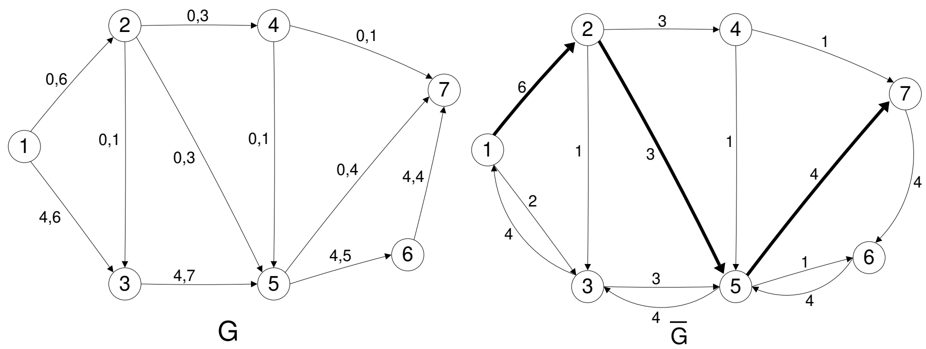
2.10 Maximum flow and minimum cut

We apply Ford-Fulkerson's algorithm. In the following figures, on the left we report the current feasible flow in the network G (with on each arc the quantity of product x_{ij} flowing through it and its capacity k_{ij}) and on the right the incremental (residual) network $\bar{G}$ associated to the current feasible flow.

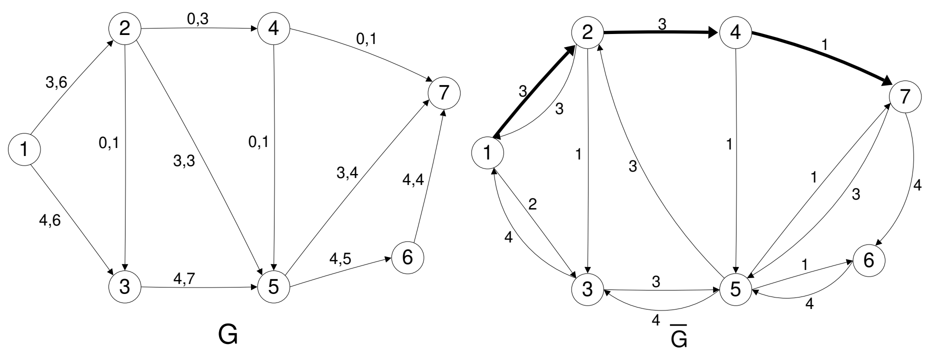
We start from the null feasible flow $\underline{x}_0 = \underline{0}$ of value $\varphi_0 = 0$. Since all arcs are empty, $\bar{G}$ is equivalent to G.



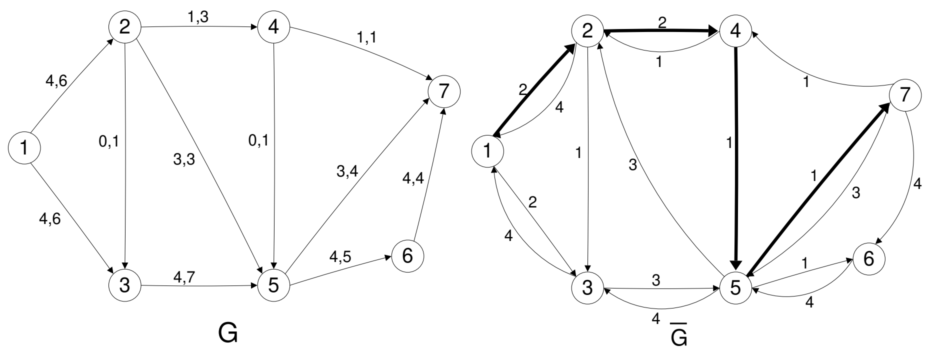
Along the augmenting path 1-3-5-6-7 we can send up to $\delta = 4$ additional units of product. δ is given by arc $(6, 7)$ which has the smallest capacity k_{ij} of all arcs on the path. Adding these $\delta = 4$ units of product to the feasible flow $\underline{x}_0$, we obtain the following feasible flow $\underline{x}_1$ (on the left) of value $\varphi_1 = 0 + 4 = 4$. The associated incremental (residual) network is reported on the right.



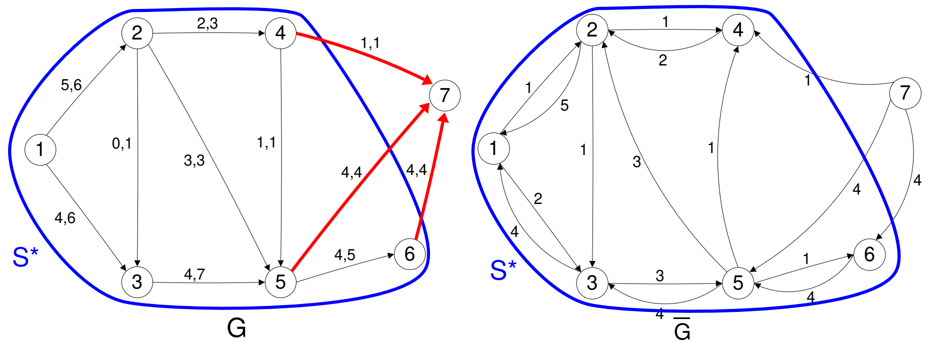
Along the augmenting path 1-2-5-7 we can send up to $\delta = 3$ additional units of product. δ is given by arc $(2, 5)$ which has the smallest residual capacity $\bar{k}_{ij}$ of all arcs on the path. Adding these $\delta = 3$ units of product to the feasible flow $\underline{x}_1$, we obtain the following feasible flow $\underline{x}_2$ of value $\varphi_2 = 4 + 3 = 7$. The associated incremental (residual) network is reported on the right.



Along the augmenting path 1-2-4-7 we can send up to $\delta = 1$ additional units of product, where δ is given by arc $(4, 7)$. We obtain the following feasible flow $\underline{x}_3$ (on the left) of value $\varphi_3 = 7 + 1 = 8$. The associated incremental (residual) network is reported on the right.



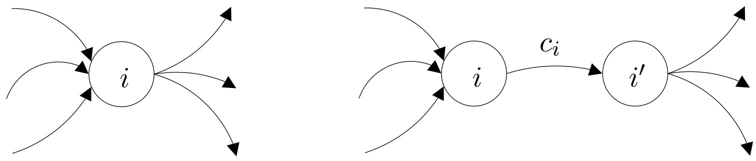
Along the augmenting path 1-2-4-5-7 we can send up to $\delta = 1$ additional units of product, where δ is given by arc $(4, 5)$ (or $(5, 7)$). We obtain the following feasible flow $\underline{x}_4$ (on the left) of value $\varphi_4 = 8 + 1 = 9$. The associated incremental (residual) network is reported on the right.



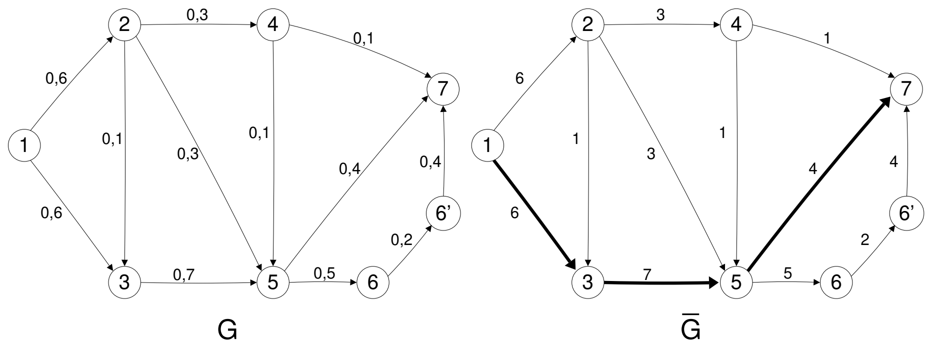
Since no other augmenting path exists (no path from node 1 to node 7 in the incremental/residual network $\bar{G}$), the algorithm halts. The set $S^* = \{1, 2, 3, 4, 5, 6\}$, highlighted in blue, contains all the nodes that can be reached from the source 1 in $\bar{G}$. S^* induces the cut $\delta(S^*)$ of minimum total capacity $k(S^*) = 9$ which is highlighted in red. Note that, according to strong duality, the value $\varphi_4 = 9$ of the feasible flow $\underline{x}_4$ is equal to the total capacity $k(S^*)$ of the cut $\delta^(S^*)$ induced by S^*.

2.11 **Maximum flow with node capacities**.

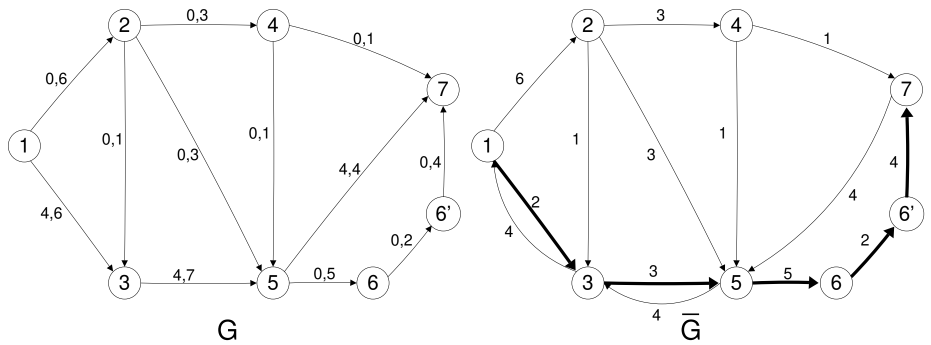
The capacities on nodes can be easily reduced to capacities on arcs. Indeed, each node i with a capacity c_i can be substituted with two auxiliary nodes, which are connected with an arc whose capacity is equal to c_i and where all entering arcs enter in the left node and all exiting arcs exit from the right node.



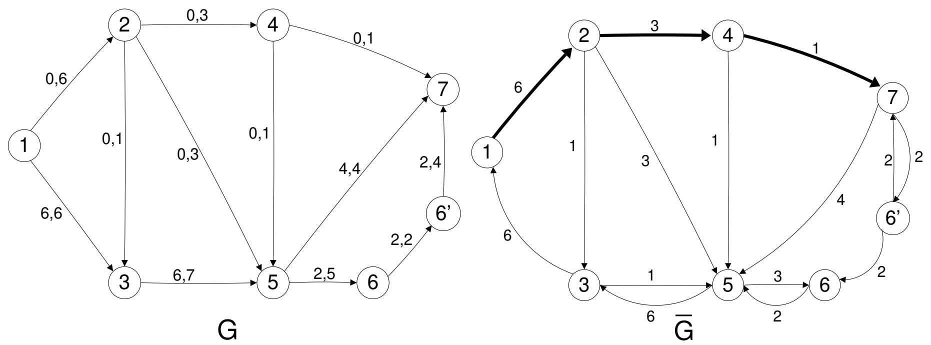
The network of exercise 2.11 is modified as follows.



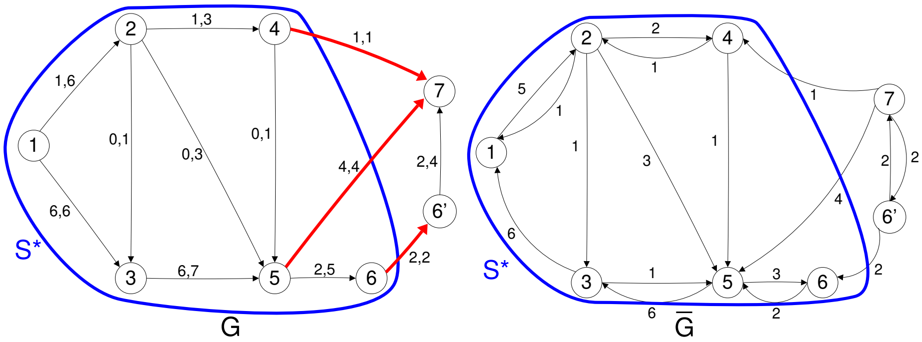
We send $\delta = 4$ additional units along the augmenting path 1-3-5-7.



We send $\delta = 2$ additional units along the augmenting path 1-3-5-6-6'-7.



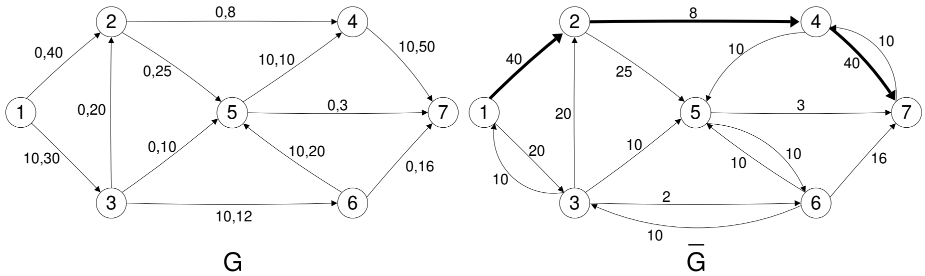
We send $\delta = 1$ additional units along the augmenting path 1-2-4-7.



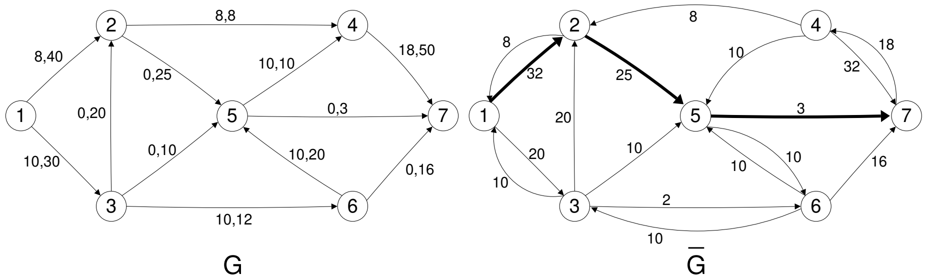
No other augmenting path exists. The minimum cut induced by $S^* = \{1, 2, 3, 4, 5, 6\}$ is highlighted in red. It has a total capacity $k(S^*) = \varphi = 7$.

2.12 **Maximum flow with a strictly positive initial flow**

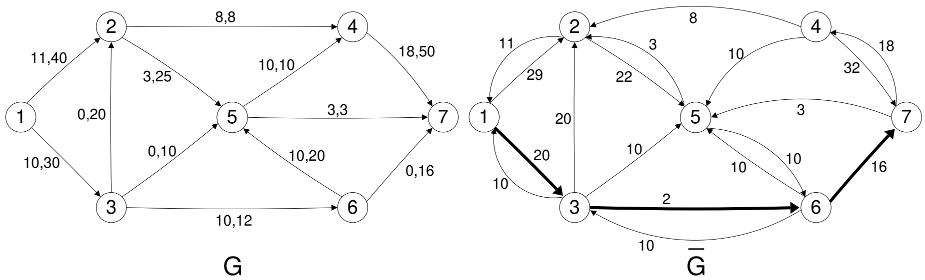
The initial feasible flow $\underline{x}_0$ of value $\varphi_0 = 10$ and the associated incremental (residual) network are as follows:



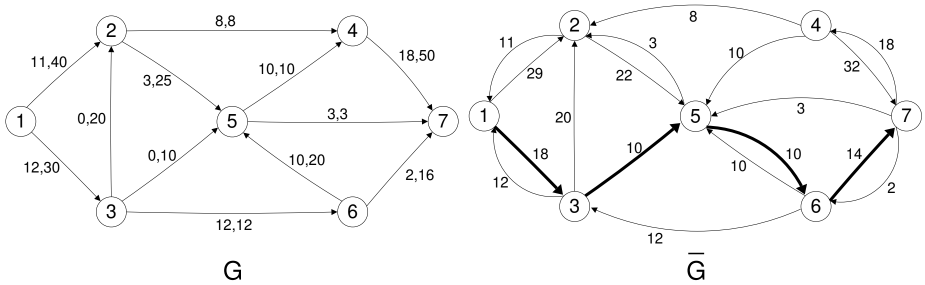
We send $\delta = 8$ additional units along the augmenting path 1-2-4-7.



We send $\delta = 3$ additional units along the augmenting path 1-2-5-7.

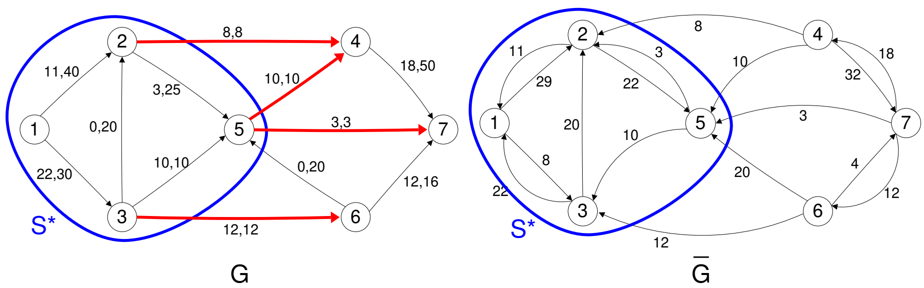


We send $\delta = 2$ additional units along the augmenting path 1-3-6-7.



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 send $\delta = 10$ additional units on path 1-3-5-6-7.

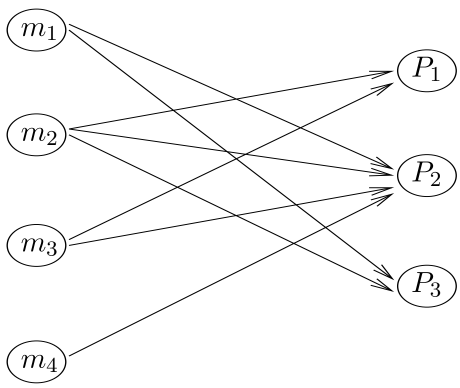


No other augmenting path exists. The minimum cut induced by $S^* = \{1, 2, 3, 5\}$ is highlighted in red. It has a total capacity $k(S^*) = \varphi = 33$.

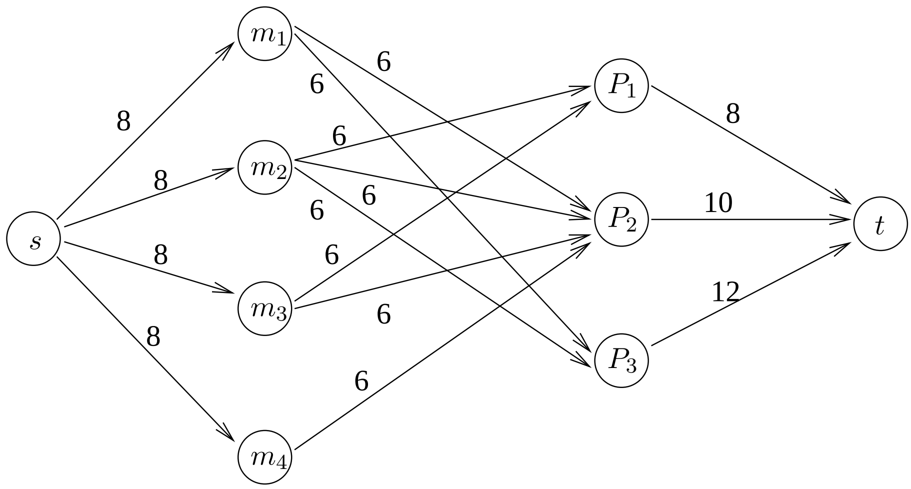
2.13 Indirect application of maximum flows

We build a network with nodes m_1, m_2, m_3, m_4 associated to the four months and nodes P_1, P_2, P_3 associated to the three projects.

For each pair of month-node m_i and project-node P_j, the arc (m_i, P_j) is included in the network if man-hours of month i can be allocated to project P_j. For instance, since project P_1 can only begin after month 1 and must be completed within month 3, there are only two arcs entering in P_1, namely (m_2, P_1) and (m_3, P_1). Thus we obtain the following bipartite oriented graph:



To turn this graph into a network and reduce the problem under consideration to a maximum flow problem, the idea is to consider a man-month as a unit of product (flow). Then we add a source s and a sink t, and connect them to the other nodes as follows:



All arcs outgoing from s have capacity 8, since there are 8 engineers available. All arcs connecting month-nodes to project-nodes have capacity 6, as no more than 6 engineers can work on the same project at the same time. All arcs entering in t have a capacity that is equivalent to the number of man-months needed to complete the corresponding project P_j.

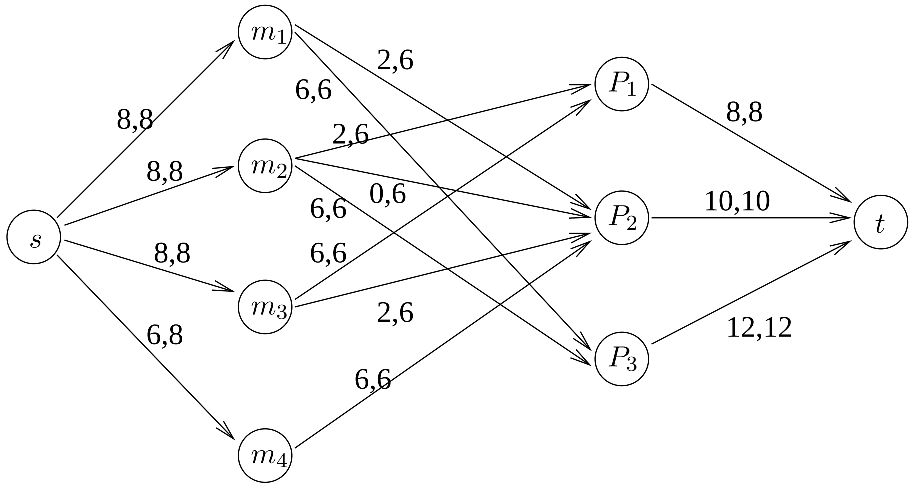
Since all capacities are integer, the maximum flow will be integer as well. To check whether all projects can be completed within the given time limits, it suffices to check whether the network admits a feasible flow of value $8 + 10 + 12 = 30$.

A feasible flow of maximum value 30 can be found by applying Ford-Fulkerson algorithm. We can start from the feasible flow $\underline{x}_0 = \underline{0}$ of value $\varphi_0 = 0$ and use the following augmenting paths:

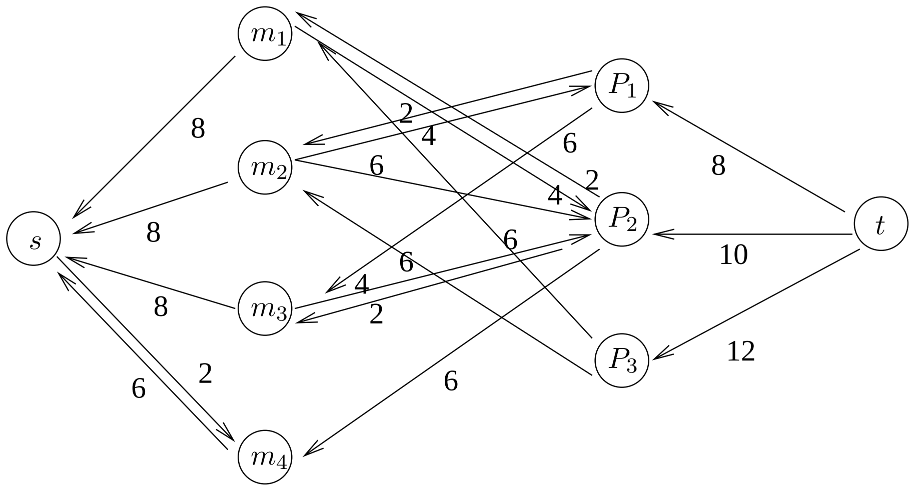
- $s - m_3 - P_1 - t$ with $\delta = 6$, yielding a feasible flow of value $\varphi_1 = 6$,
- $s - m_2 - P_1 - t$ with $\delta = 2$, yielding a feasible flow of value $\varphi_2 = 8$,
- $s - m_1 - P_3 - t$ with $\delta = 6$, yielding a feasible flow of value $\varphi_3 = 14$,
- $s - m_2 - P_3 - t$ with $\delta = 6$, yielding a feasible flow of value $\varphi_4 = 20$,

- $s - m_4 - P_2 - t$ with $\delta = 6$, yielding a feasible flow of value $\varphi_5 = 26$,
- $s - m_1 - P_2 - t$ with $\delta = 2$, yielding a feasible flow of value $\varphi_6 = 28$,
- $s - m_3 - P_2 - t$ with $\delta = 2$, yielding a feasible flow of value $\varphi_7 = 30$.

The resulting feasible flow $\underline{x}_7$ of value $\varphi_7 = 30$ is as follows



and the associated residual network $\overline{G}_7$ is



Note that in $\overline{G}_7$ only node m_4 can be reached from node s, hence $S^* = \{s, m_4\}$.

Since the cut $\delta_G(S^*) = \{(s, m_1), (s, m_2), (s, m_3), (m_4, P_2)\}$ has a total capacity of $k(S^*) = 8 + 8 + 8 + 6 = 30$ which is equal to the value $\varphi_7 = 30$ of the feasible flow $\underline{x}_7$, weak duality implies that the feasible flow $\underline{x}_7$ is of maximum value and the cut $\delta_G(S^*)$ is of minimum total capacity (among all the cuts separating the source s from the sink t).