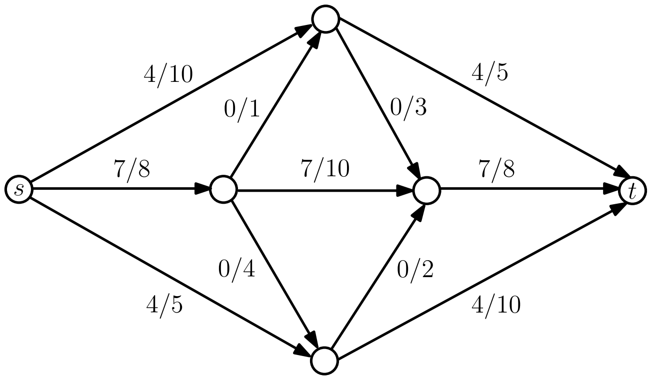
Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
M. Ahmadi, O. Saukh, A. R. Molla
December 16, 2015

Algorithm Theory, Winter Term 2015/16
Problem Set 7 - Sample Solution

Exercise 1: Ford-Fulkerson (5+1+2 points)

Consider the following flow network, where for each edge, the capacity (second larger number) and a current flow value (first smaller number) are given.



a) Find a maximal flow in the given network with the help of the Ford Fulkerson algorithm. Draw the residual graph with all the residual capacities in all steps.

b) Is the following statement true or false? *If all edges in a flow network have distinct capacities, then there is a unique maximum flow.* Justify your answer with a (short) proof or give a counterexample.

c) You are given a (connected) directed graph $G = (V, E)$, with positive integer capacities on each edge, a designated source $s \in V$, and a designated sink $t \in V$. Additionally you are given a current maximum $s - t$ flow $f : E \to \mathbb{R}_{\geq 0}$.

   Now suppose we increase the capacity of one specific edge $e_0 \in E$ by one unit. Show how to find a maximum flow in the resulting capacitated graph in time $\mathcal{O}(|E|)$.

Solution:

a) The Ford-Fulkerson algorithm starts by looking for an augmenting $s-t$-path in the residual graph, and after finding such a path it updates the flow and residual cpacity values of the edges in the residual graph. This procedure is repeated until no more $s - t$-path is found in the residual graph. At the beginning of the algorithm, we have initial flow and then algorithm starts with the corresponding residual graph. Next we search for an $s - t$-path in the residual graph, and update the flow values of all edges. On each iteration of the algorithm, we search for some bottleneck on the augmenting $s - t$-path, update the flow and change the capacities of the residual graph. The set of figures 1-3 show the resulting graph step-by-step. The selected path in each step is shown in red.

   Step 1: Path is $p_{s,1,3,2,4,t}$. The bottleneck on the path is 3, see figure 1.

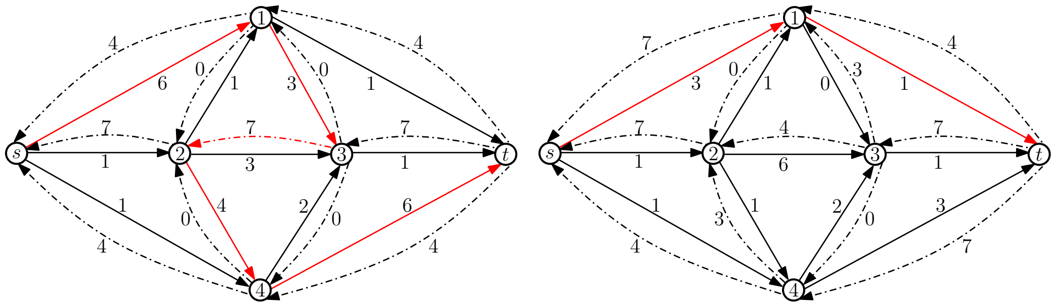


Figure 1: Residual graphs at steps 1 and 2. Currently considered paths are $p_{s,1,3,2,4,t}$ and $p_{s,1,t}$.

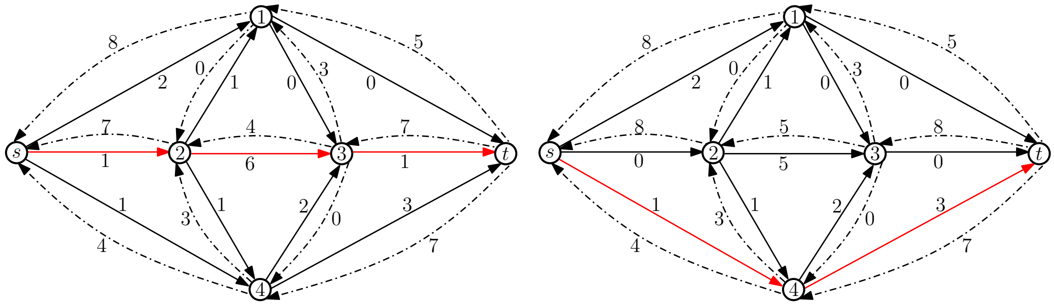


Figure 2: Residual graphs at steps 3 and 4 of the algorithm. Considered paths are $p_{s,2,3,t}$ and $p_{s,4,t}$.

Step 2: Path is $p_{s,1,t}$. The bottleneck on the path is 1.

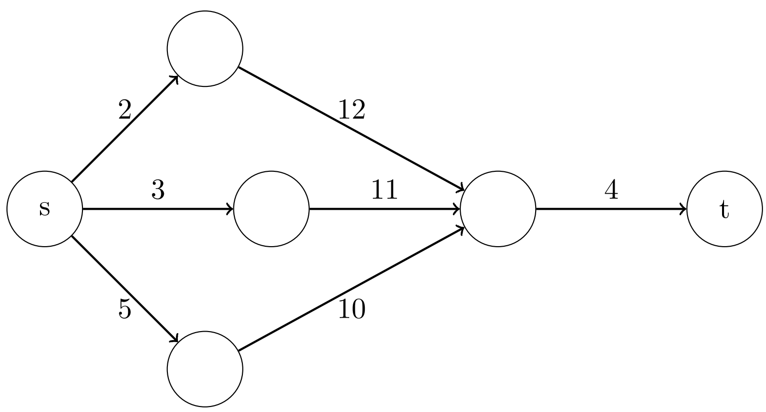
Step 3: Path is $p_{s,2,3,t}$. The bottleneck on the path is 1, see figure 2.

Step 4: Path is $p_{s,4,t}$. The bottleneck on the path is 1, see figure 3.

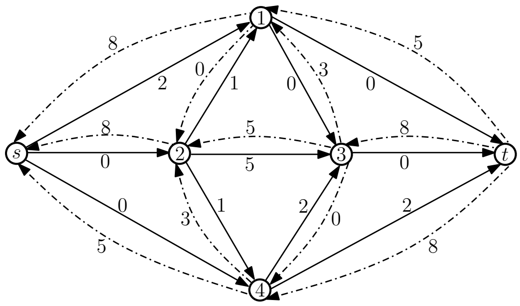
After Step 4 (in the left graph of figure 3) there are no more $s-t$-paths left in the graph; thus the current flow is a maximum flow. See the right graph on figure 3 as flow network with the maximum flow, which we computed.

b) In the following network all edges have distinct capacities and the max flow value is 4 (cut the node t from the graph). There are several ways of obtaining a flow of 4, e.g.,

- send 4 over the lowest path or
- send 1 on the lowest path and 3 on the path in the middle.



c) Let $m = |E|$ denote the number of edges and $\tilde{G}$ denote the flow network with the increased capacity. We cannot compute a maximum flow from scratch in $\tilde{G}$ in time $\mathcal{O}(m)$; thus we have to reuse the existing maximum flow f_{max}. The residual network of f_{max} in $\tilde{G}$ and one (!) augmenting path (if it exists) can be computed in time $\mathcal{O}(m)$. Because all values are integers it is sufficient to prove the following claim:



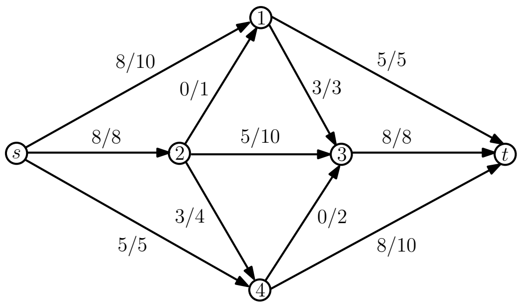


Figure 3: Residual graph after step 4 of the algorithm with no augmenting path (left). Final flow graph at the end of the algorithm (right).

Claim: The maximum flow in $\tilde{G}$ is smaller or equal to $|f_{max}| + 1$.

This claim can be proven with the max-flow min-cut Theorem. There is a min-cut $C_G = (S, V \backslash S)$ in G with value $|f_{max}|$. Now there are two cases for the corresponding cut $C_{\tilde{G}}$ in $\tilde{G}$.

- e_0 is an edge from S to $V \backslash S$. Then the value of the cut $C_{\tilde{G}} = (S, V \backslash S)$ is the value of C_G plus 1, that is $|f_{max}| + 1$. By the max-flow min-cut Theorem any flow in $\tilde{G}$ must have a value smaller or equal to $|f_{max}| + 1$.
- e_0 is not an edge from S to $V \backslash S$. Then the value of $C_{\tilde{G}}$ is $|f_{max}|$ and the claim holds with the max-flow min-cut Theorem.

So as a max flow in $\tilde{G}$ must have a value smaller or equal $|f_{max}| + 1$ we are done after at most one augmenting path.

Exercise 2: Seating arrangement (4 points)

A group of international students go out to eat dinner together. To increase their social interaction, they would like to sit at tables so that no two students from the same country are at the same table. Show how to formulate finding a seating arrangement that meets this objective as a maximum flow problem. Assume that the students are from p different countries and there are s_i students from the i^{th} country. Also assume that q tables are available and that the j^{th} table has a seating capacity of t_j.

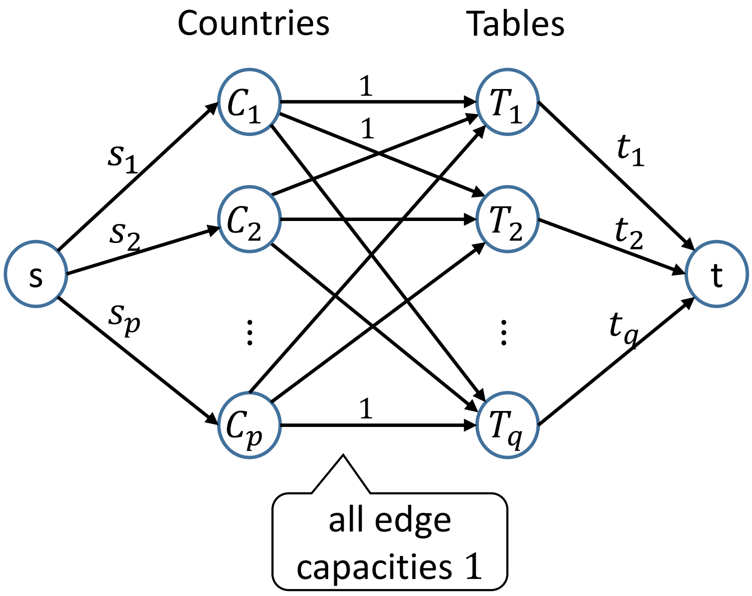
Solution

Construct a complete bipartite graph where, each country represents a vertex in one partition, and each table represents a vertex in another partition. There is a unit capacity edge between a country vertex and a table vertex. There is a source vertex s and a destination vertex t. The source s has an edge to every country, and there is an edge from every table to the destination t. Capacity of source and i^{th} country edge is s_i for all i, and the capacity of every edge between table j and destination is t_j. See the figure below.
Formally, the capacitated graph $G(V, E)$ is defined below:

$$
\begin{aligned}
V &= \{s, t\} \cup \{C_i \mid 1 \leq i \leq p\} \cup \{T_j \mid 1 \leq j \leq q\} \\
E &= \{(s, C_i) \mid 1 \leq i \leq p\} \cup \{(C_i, T_j) \mid 1 \leq i \leq p, 1 \leq j \leq q\} \cup \{(T_j, t) \mid 1 \leq j \leq q\}
\end{aligned}
$$

$capacity(s, C_i) = s_i$
$capacity(C_i, T_j) = 1$
$capacity(T_j, t) = t_j$

An integral solution to the max-flow problem gives an assignment of students to tables. A flow from C_i to T_j is either 0 or 1: with 1 meaning that a student of country i sits in table j. The maximum flow maximizes the number of student that can be seated under the stated constraints. A seating



arrangement is possible iff the maximum flow equals the sum of number of students from all the countries.