Algorithmic Discrete Mathematics
5. Exercise Sheet

TECHNISCHE UNIVERSITÄT DARMSTADT

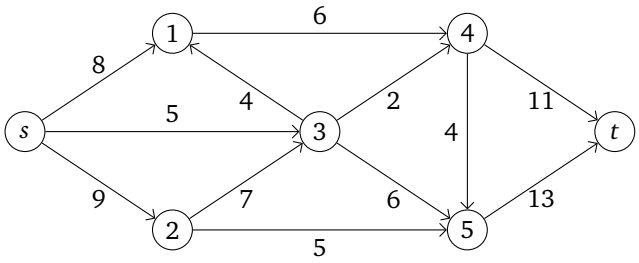
Department of Mathematics
Andreas Paffenholz
Silke Horn

SS 2013
12/13 June 2013

Groupwork

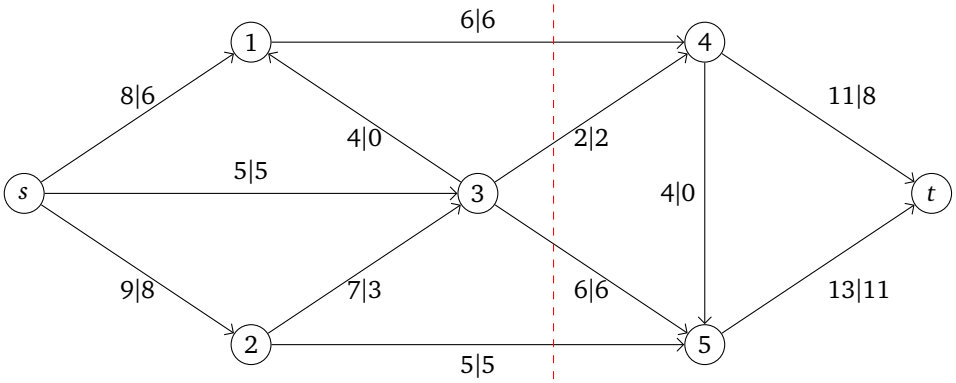
Exercise G1

Using the Ford-Fulkerson method, compute a maximal flow in the following network:



Also determine a minimal cut in G.

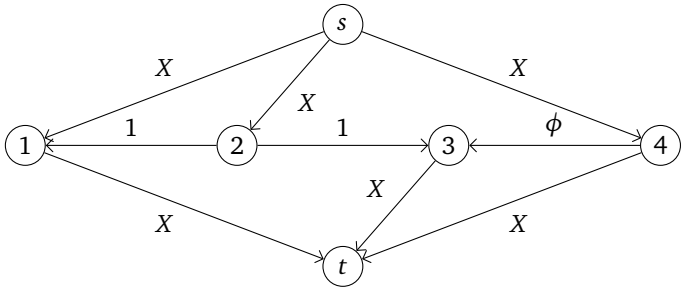
Solution: The following showas a maximal flow and a minimal cut of value 19:



Exercise G2

The goal of this exercise is to show that the Ford-Fulkerson method need not terminate if we allow irrational edge capacities.

Consider the following network with capacities c_e.



Here X is some large integral constant and $\phi = \frac{1}{2}(\sqrt{5}-1)$. (Note that $\phi^n = \phi^{n+1} + \phi^{n+2}$ for any $n \geq 0$.)

(a) Show by induction that for any integer $n \geq 0$ the residual capacities of the three horizontal edges can be brought to the values $\phi^n, 0, \phi^{n+1}$.

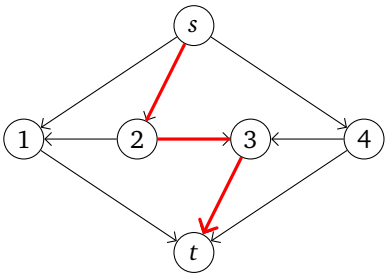
(b) Conclude that Ford-Fulkerson need not terminate on this network. Does it converge?

(c) Find a network where Ford-Fulkerson converges, but not to a maximal flow.

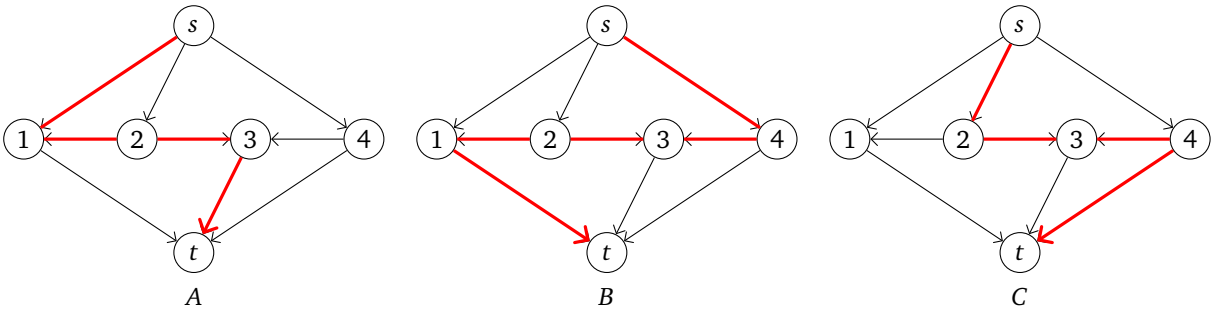
Solution:

(a) In the following we always consider the residual capacities of the three horizontal edges (in the direction of the edge). In the beginning they are $1, 1, \phi$.

First choose the following path and increase the flow by 1:



Now the residual capacities are $1, 0, \phi$. So now assume that the capacities are $\phi^{n-1}, 0, \phi^n$ for some $n \geq 1$. By increasing the flow along the following three paths, the capacities become $\phi^{n+1}, 0, \phi^{n+2}$:



(1) Augmenting along the path B, adding ϕ^n to the flow, the residual capacities are $\phi^{n+1}, \phi^n, 0$.

(2) Augmenting along the path C, adding ϕ^n to the flow, the residual capacities are $\phi^{n+1}, 0, \phi^n$.

(3) Augmenting along the path B, adding ϕ^{n+1} to the flow, the residual capacities are $0, \phi^{n+1}, \phi^{n+2}$.

(4) Augmenting along the path A, adding ϕ^{n+1} to the flow, the residual capacities are $\phi^{n+1}, 0, \phi^{n+2}$.

(b) If we choose the augmenting paths as above, the residual capacities of the horizontal edges will never reach 0 and we can always find another augmenting path.

The flow value converges to

$$1 + 2\sum_{n=1}^{\infty} \phi^i = 1 + \frac{2}{1-\phi} < 7.$$

The maximal flow value, however, is $2X + 1$.

In general the flow value does always converge since the sequence of flow values is monotone and bounded.

(c) See above.

Exercise G3

Let $(G = (V, E), s, t, c)$ be a network with integral capacities $c(e) \in \mathbb{Z}$ for all edges $e \in E$. Prove or refute the following assertions:

(a) If all capacities are even then there is a maximal $(s-t)$-flow f such that $f(e)$ is even for all $e \in E$.

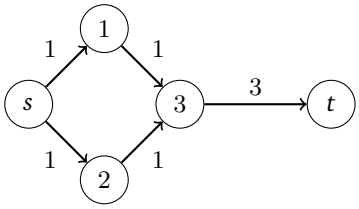
(b) If all capacities are odd then there is a maximal $(s-t)$-flow f such that $f(e)$ is odd for all $e \in E$.

Solution:

(a) One can divide all capacities by 2. On the modified graph there is an integral maximal flow. If we multiply this by 2 we get an even maximal flow on the original graph.

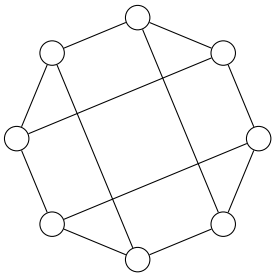
Alternatively, one can replace "integral" by "even" in the proof that Ford-Fulkerson is correct for integral edge weights.

(b) Counter example:

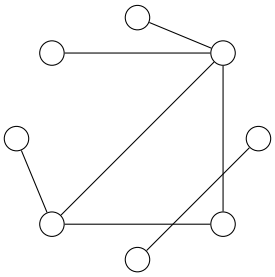


Exercise G4
Let $G = (V, E)$ be a graph. A subset $M \subseteq E$ is a *matching* in G if $m \cap m' = \emptyset$ for all $m, m' \in M$. A matching M is *perfect* if $2|M| = |V|$.
In each of the following graphs determine a perfect matching or show that no perfect matching exists.

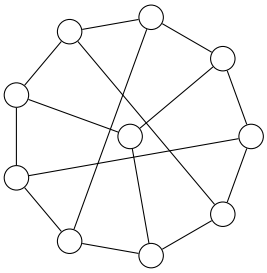
(a)



(b)

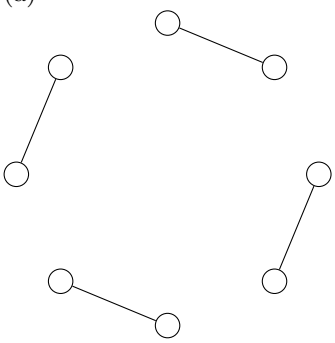


(c)

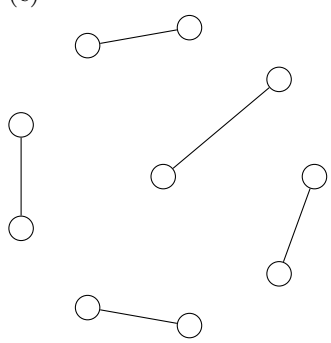


Solution:

(a)



(c)



For the graph in (b) there is no perfect matching.

Homework

Exercise H1 (5 points)
Let $(G = (V, E), s, t, c)$ be a network with integral capacities $c(e) \in \mathbb{Z}_+$ for all $e \in E$. Let f be a maximal integral flow in this network. We assume that the capacity of one edge e

(a) is increased by 1,

(b) is decreased by 1.

Describe an algorithm with complexity $\mathcal{O}(m + n)$ that determines a maximal flow in the new network. Improve your algorithm (or your analysis) to $\mathcal{O}(m)$.

Solution: First we note that in a connected graph $|E| \geq |V| - 1$ and hence both DFS and BFS run in time $\mathcal{O}(m)$. If the graph is not connected we only consider one connected component (the one with the source).

(a) If the capacity of one edge is increased by one, one only needs to look for an augmenting path exactly once since the value of the maximal flow can be increased by at most one. This can be done by BFS in time $\mathcal{O}(m)$.

(b) If the capacity of one edge is decreased by one, there are two cases:
   i. The maximal flow through this edge remains feasible: In this case there is nothing to do.
   ii. The flow in the edge exceeds its capacity by exactly one: Let (u, v) be the edge; we search for a path p_1 from s to u and a path p_2 from v to t. These two paths shall be disjoint and only contain edges with positive flow. By flow composition these paths exist or there is a cycle with positive flow. We can use BFS

to determine p_1 and p_2 in time $\mathcal{O}(m)$. If $p_1(u,v)p_2$ contains a cycle, we reduce the flow along this cycle by one to obtain a feasible (and maximal) flow.

Otherwise we reduce the flow along the path $p_1(u,v)p_2$ by one to obtain a feasible flow. Now we have to check whether we can increase the flow by one (along a different path). Thus, we look for an augmenting path and if applicable increase the flow along this path by one. Since this is also possible in time $\mathcal{O}(m)$ we obtain a total running time of $\mathcal{O}(m)$.

Exercise H2 (5 points)

(a) An edge e in a network $(G=(V,E),s,t,c)$ where t can be reached from s, is called *upwards critical* if increasing the capacity of e increases the value of the maximal flow. Does every network possess an upwards critical edge?

Describe an algorithm that finds all upwards critical edges and has a considerably better running time than solving m max flow problems.

(b) An edge e in a network $(G=(V,E),s,t,c)$ where t can be reached from s, is called *downwards critical* if decreasing the capacity of e decreases the value of the maximal flow. Does every network possess a downwards critical edge?

Describe an algorithm that finds all downwards critical edges and analyse its running time.

Solution:

(a) Not every graph possesses an upwards critical edge. Consider for instance a path of length ≥ 2 where every edge has equal capacity.

We first compute a maximal flow in our network. Then, with exercise H1 (a) above, the problem of finding all upwards critical edges can be solved in time $\mathcal{O}(m^2)$ (by running the algorithm described above for every edge).

Here is another way:

An edge is upwards critical if and only if it is contained in every minimal cut.

Let f be a maximal flow. Let S be the set of nodes that can be reached from s in G_f and let T be the set of nodes from which t can be reached in G_f. Then the set of upwards critical edges equals $\delta^+(S)\cap\delta^-(T)$.

The sets S and T, respectively $\delta^+(S)$ and $\delta^-(T)$ can be determined in time $\mathcal{O}(m)$ (*e.g.* using BFS or enumeration). Since $\delta^+(S)$ and $\delta^-(T)$ are of cardinality $\mathcal{O}(m)$ their intersection can easily be determined in $\mathcal{O}(m^2)$. By sorting the lists (with an appropriate sorting algorithm) we can do this in $\mathcal{O}(m\log m)$ or even in $\mathcal{O}(n^2)$ (which needs some more work).

(b) An edge is downwards critical if and only if it is contained in some minimal cut. Hence every graph with positive maximal flow possesses a downwards critical edge and the set of downwards critical edges need not equal the set of upwards critical edges.

Again, we need to compute a maximal flow in the network. Then, with exercise H1 (b) above, the problem of finding all downwards critical edges can be solved in time $\mathcal{O}(m^2)$.

Exercise H3 (5 points)

Let $G=(V,E)$ be an undirected graph and $s\neq t\in V$. A subset $F\subseteq E$ is $(s-t)$-*separating* if any $(s-t)$-path uses at least one edge of F. A collection $P_1,\ldots,P_k$ of $(s-t)$-paths in G is *edge-disjoint* if no pair $P_i,P_j, i\neq j$ have an edge in common.

(a) Prove the *edge version of the Theorem of Menger*:

The maximal number of edge-disjoint paths in G equals the minimal size of an $(s-t)$-separating edge set.

Hint: Apply the MaxFlow-MinCut Theorem to a suitable network.

(b*) (Bonus exercise – no points) A subset $U\subseteq V$ is $(s-t)$-*separating* if any $(s-t)$-path uses at least one node of U. Two $(s-t)$-paths are *internally disjoint* if they only share the nodes s and t.

Prove the *node version of the Theorem of Menger*:

Assume $\{s,t\}\notin E$. Then the maximal number of internally disjoint $(s-t)$-paths equals the minimal size of an $(s-t)$-separating node set.

Hint: Construct a directed graph as above, then replace each node $v\in V\setminus\{s,t\}$ by a pair v^-,v^+ and a directed edge (v^-,v^+). Again apply the MaxFlow-MinCut Theorem.

Solution:

(a) It is clear that the number of edge-disjoint paths is less than or equal to the size of any $(s-t)$-separating set since each $(s-t)$-path uses at least one edge from every $(s-t)$-separating set.

For the reverse inequality we construct a network with a directed graph $D=(V,A)$ by:

$$\begin{aligned} uv \in A \Leftrightarrow\ & \{u,v\} \in E \text{ and} \\ & (1)\ u = s \text{ or} \\ & (2)\ v = t \text{ or} \\ & (3)\ \{u,v\} \cap \{s,t\} = \emptyset. \end{aligned}$$

Moreover, we define the capacity $c : A \to \mathbb{R} : c(a) = 1$ for all $a \in A$. Then any $(s-t)$-cut and hence any maximal flow in D are integral. Thus, we have an integral path decomposition and each edge appears in at most one path.

On the other hand, any $(s-t)$-cut in D defines an $(s-t)$-separating set in G of the same size. Moreover, any integral flow yields a flow decomposition with at least as many paths as the flow value. Hence we get

$$\text{max number of edge-disjoint } (s-t)\text{-paths} \geq \text{max flow} = \text{min cut} \geq \text{min } (s-t)\text{-separating set.}$$

(b) Again, it is clear that the number of internally disjoint paths is less than or equal to the size of any $(s-t)$-separating set.

For the reverse inequality we again construct a network with a directed graph (V',A). We start with the directed graph as above. Then we replace every vertex $v \in V \setminus \{s,t\}$ by a pair v^-, v^+ and a directed edge (v^-, v^+). Moreover, we replace each edge $(u,v) \in \delta^-(v)$ by an edge (u, v^-) and each edge $(v,u) \in \delta^+(v)$ by an edge (v^+, u). Again we define the capacity $c : A \to \mathbb{R}_{\geq 0} : c(a) = 1$ for each $a \in A$.

We can now assume that a minimal $(s-t)$-cut consists only of edges of the form (v^-, v^+). In fact, assume that a minimal cut $P = (S,T)$ contains an edge (u^+, v^-) with $u^+ \in S, v_- \in T - \{t\}$. Now consider the cut $P' = (S \cup \{v^-\}, T - \{v^-\})$. Then $E_{P'} \subseteq E_P \cup \{(v^-, v^+)\} - \{(u^+, v^-\}$ and hence $c(P') \leq c(P)$.

Such a cut, however, corresponds to an $(s-t)$-separating node set of the same size.

The remainder of the proof is analogous to the edge version. Hence we get

$$\text{max number of internally disjoint } (s-t)\text{-paths} \geq \text{max flow} = \text{min cut} \geq \text{min } (s-t)\text{-separating set.}$$

Exercise H4 (5 points)
Let $G=(V,E)$ be a directed graph, $c : E \to \mathbb{R}_{\geq 0}$ a capacity and $f : E \to \mathbb{R}$ a flow on G. Prove or disprove the following statements:

(a) f is maximal $\Rightarrow f(e) = 0$ or $f(e) = c(e)$ for all $e \in E$.

(b) There is a maximal flow such that $f(e) = 0$ or $f(e) = c(e)$ for all $e \in E$.

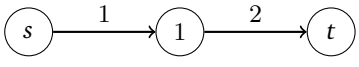
(c) A minimal cut is unique if all capacities are pairwise distinct.

(d) Multiplying all capacities $c(e)$ by a number $\lambda > 0$ does not change the minimal cuts.

(e) Adding a number $\lambda > 0$ to all capacities $c(e)$ does not change the minimal cuts.

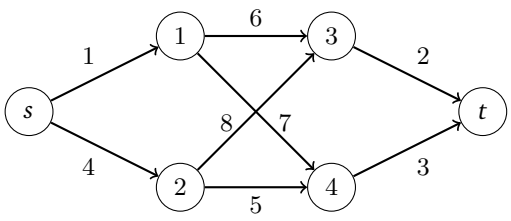
Solution:

(a) False, counter example:

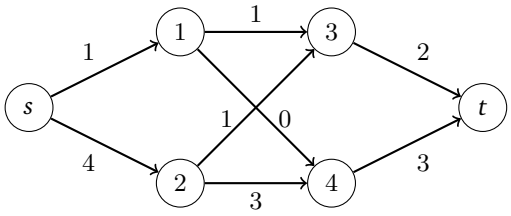


(b) False, see (a) for a counter example.

(c) False, counter example: Consider the following graph:



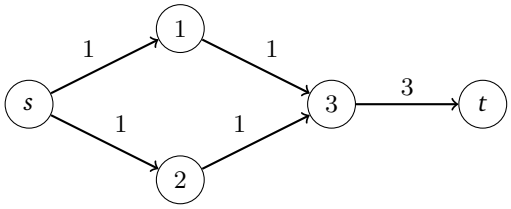
The value of a maximal (s,t)-flow (and hence the capacity of a minimal (s,t)-cut) is 5, as demonstrated by the following flow:



The cuts $(\{s\},\{1,2,3,4,t\})$ and $(\{t\},\{1,2,3,4,s\})$ both have capacity 5.

(d) True. Let P be a minimal cut in a graph G. Then $c_P \leq c_{P'}$ for any cut P' in G. But this is equivalent to $\lambda c_P \leq \lambda c_{P'}$ for any scalar $\lambda > 0$.

(e) False, counter example: Consider the following graph:



A minimal cut of value 2 is formed by $(\{s\},\{1,2,3,t\})$. After adding $\lambda = 2$ to all capacities, the (unique) minimal cut of value 5 is formed by $(\{t\},\{s,1,2,3\})$, while the value of the cut $(\{s\},\{1,2,3,t\})$ is 6.