Exercise 7, Discrete Mathematics for Bioinformatics

Sascha Meiers, Martin Seeger

Winter term 2011/2012

7.1 Network Flow (Niveau II)

a) $\Rightarrow$ b) This is the same as the Theorem on pp. 5006-5007 in the lecture since the existence of an augmenting path is equivalent to the reachability of t from s in the residual network G_f. If there is such a path then f is not a maximum.

b) $\Rightarrow$ c) Assume there is no augmenting path in G_f from s to t. Let

$$S = \{v \in V | \exists \text{ path } s \rightsquigarrow t \text{ in } G_f\}, \; T = V \setminus S.$$

Then (S, T) is a cut because $s \in S$ and $T \notin S$.
It is easy to see that this cut is saturated by f: pick $(u, v) \in S \times T$. If $(u, v) \in E$, hence $(u, v) \in E \cap S \times T$ then if $f(u, v) < \text{cap}(u, v)$, v would become reachable from u and hence also from s. Therefore $f(u, v) = \text{cap}(u, v)$. On the other hand, assume $(v, u) \in E$, i.e. $(u, v) \in E \cap T \times S$. Then if $f(u, v) > 0$, we would have $v \in S$ which means that really $f(u, v) = 0$. Therefore f saturates (S, T). By the Lemma on p. 5006, $\text{val}(f) = \text{cap}(S, T)$.

c) $\Rightarrow$ a) We have by the Lemma on p. 5006 that for any flow f and cut (S, T)

$$\text{val}(f) \leq \text{cap}(S, T).$$

Therefore a flow which saturates this inequality must be maximal.

In summary, we have given an construction for a minimum cut that satisfies the Max-Flow Min-Cut Theorem given a maximum flow resp. its residual network.

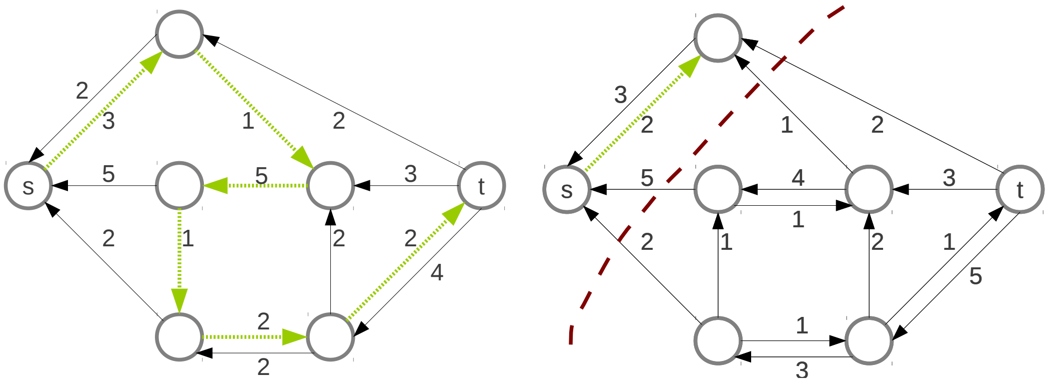
7.2 Network Flow (Niveau I)

Replace each node v with its associated *node capacity* c_v by two new nodes v_{in}, v_{out} and an edge (v_{in}, v_{out}) with *edge capacity* c_v. Then all ingoing edges $(u, v) \in E$ point to v_{in}, the outgoing edges $(v, u) \in E$ start from v_{out}.

7.3 Ford-Fulkerson

The first figure shows the residual network according to the current flow. An augmenting path is marked in green.

The Ford-Fulkerson algorithm uses this path to increase the network flow by 1. After that the residual network as seen on the right remains. There is no augmenting path left, as a breadth-first search can decide. The same breadth-first search (see green arrow) is used to find a minimum cut (see red line), which has a capacity of 10.



7.4 Bipartite matching

The figure below shows the augmenting-path method to find a optimal matching in a bipartite graph. The set of edges concerned for the matching are marked black and in each step we label a remaining augemting path in blue.