Algorithmics and complexity

TD 3/7 – Flow graphs

Training exercises

Note: The solution elements given here are not complete. Their purpose is only to guide you. We encourage you to write a proper answer, as you would do for the exam. If any question remains, feel free to ask your tutorial supervisor for help.

Exercice 1 : Routing

A storage server S is connected to a terminal T through a network deployed upon four routers A, B, C, D. The link capacities, expressed in Mbps, are given below:

	A	B	C	D	T
S	2	6	4		
A		3		7	
B				2	5
C		1			
D	3				6

A user at the terminal T has to download a voluminous file stored at the server S. Through which links the file should be transmitted to shorten the user waiting time.

Question 1

Which well-known problem corresponds to this routing problem? Which algorithm could be used to solve it?

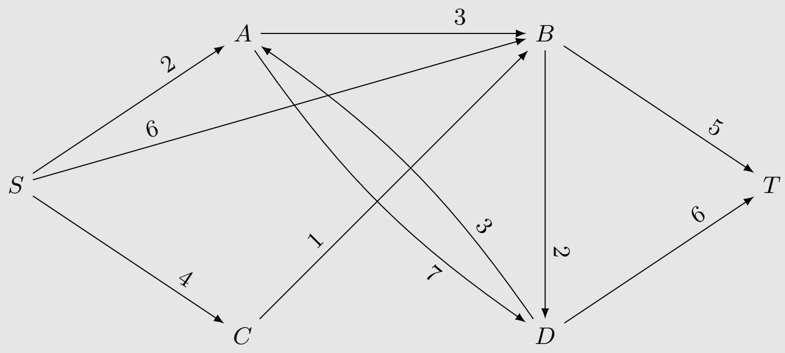
Solution elements :

- This is an optimization problem: Maximum Flow problem on a flow graph (routed network here).
- Ford Fulkerson Algorithm.

Question 2

Model this instance of the problem, and run this resolution algorithm step by step. Give the minimum cut.

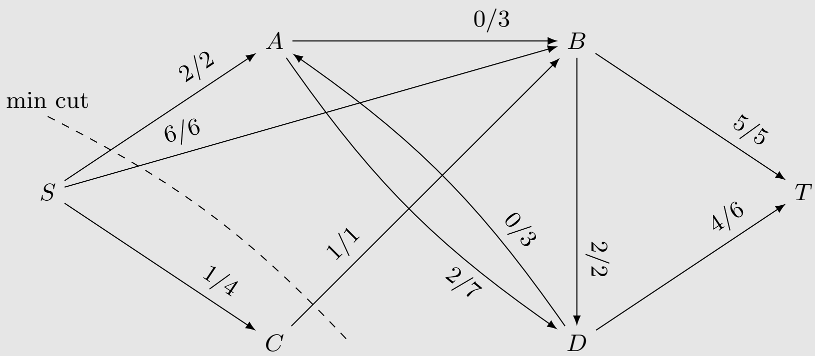
Élements de correction :



We apply for example a DFS at each step of the algorithm to find an augmenting path:

- $S \to A \to B \to T : 2$
- $S \to B \to T : 3$
- $S \to B \to D \to T : 2$
- $S \to B \leftarrow A \to D \to T : 1$
- $S \to C \to B \leftarrow A \to D \to T : 1$
- the last call to find an augmenting path visits S and C and stops

The max flow is 9, the minimum cut is $\{S, C\}$, and we obtain the following flow graph :



Question 3

What is the throughput of the routing found? How much time does it take to transfer a 100MB file ($1MB = 2^{23} bits$)?

Solution elements :

Maximum flow = $9Mbits/s$ according the the previous question. Sending the file will require:

$$\frac{100 \times 2^{23}[\text{bits}]}{9 \cdot 2^{20}[\text{bits/s}]} = 88{,}8 \text{ [s]}.$$

Exercice 2 : Graph cut

Let $G = (V, E)$ be a non-oriented connected graph and let s and p be two vertices of the graph.

A sub-set of edges $S \subseteq E$ is an (s, p)**-separator** if any path between s and p go necessarily through an edge of S.

A set of paths between s and p are **edge-disjoint** if each pair of paths P_i and P_j with $i \neq j$ has no edges in common.

Menger's theorem: The maximum number of edge-disjoint paths that can be found between s and p in G is equal to the size of the minimum (s, p)-separator of G

Question 1

Prove the Menger's theorem.

Solution elements :

You need to turn G into a flow graph and then use the min-cut theorem presented in the lecture.

Each non-directed edge $\{u, v\}$ will give rise to two directed arcs (u, v) and (v, u), except those issuing from s or entering p. These edges will only be turned into a single arc directed "in the right direction". **Write this transformation properly!**

We set the capacity of each arc to 1. We can observe that:

- The maximum number of *edge-disjoint* chains between s and p is equals to the maximum flow, since all capacities have been set to 1 (each edge can only be used once).
- The size of the minimum (s, p)-separator in G is equal to the capacity of the minimum (s, p)-cut in G'.

Using the *min-cut* theorem, we have:

$$\begin{aligned}\text{max number of } \textit{edge-disjoint} \text{ chains} &= \text{max flow} \\ &= \text{min cut} \\ &= \text{size of the smallest } (s,p)\text{-separator.}\end{aligned}$$