Week III

Overview

- Flow Network;
- Cut;
- Maximum Flow;
- Minimal Cut.

Definitions

- A **flow network** is a **directed** graph where each edge has a **capacity** and each edge receives a **flow**, where the amount of flow allowed in each edge cannot surpass its capacity;
- A **cut** in graph theory corresponds to a **partition** of the nodes in a graph splitting them into **disjoint subsets**.

Problem (Maximum Flow Problem (MaxFlow))

Given a flow network represent as a digraph $G = (v, E)$ with capacities u and unique source and unique sink s and t respectively, such that $s, t \in V$.
The goal is to find an s-t-flow of ***maximum*** *value.*

Problem (Minimum Cut Problem (MinCut))

Given a flow network represent as a digraph $G = (v, E)$ with capacities u and unique source and unique sink s and t respectively, such that $s, t \in V$.
The goal is to find an s-t-cut of ***minimum capacity****.*

Ford-Fulkerson's

Algorithm 1: FORD-FULKERSON ALGORITHM

Input: digraph $G = (V, E)$, capacities
$u \colon E \longrightarrow \mathbb{Z}_+$, $s, t, \in V$
Output: maximal s-t-flow f

set $f(e) = 0$ for all $e \in E$
while *there exists f-augmenting path in G_f* **do**
  choose f-augmenting path P
  set $\Delta_f(P) = \min_{a \in E(P)} u_f(a)$
  augment f along P by $\Delta_f(P)$
  update G_f
return f

Emonds-Karp's

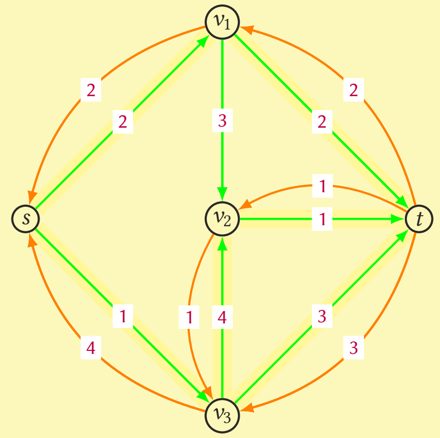
Algorithm 2: EDMONDS-KARP ALGORITHM

Input: digraph $G = (V, E)$, capacities
$u \colon E \longrightarrow \mathbb{R}_+$, $s, t, \in V$
Output: maximal s-t-flow f

set $f(e) = 0$ for all $e \in E$
while *there exists f-augmenting path in G_f* **do**
  choose f-augmenting path P with minimal number of edges
  set $\Delta_f(P) = \min_{a \in E(P)} u_f(a)$
  augment f along P by $\Delta_f(P)$
  update G_f
return f

Maximum Flow ILP

$$
\begin{aligned}
\max \quad & \sum_{e \in \delta^+(s)} f_e & & \text{(1a)}\\
\text{s.t.} \quad & \sum_{e \in \delta^-(v)} f_e - \sum_{e \in \delta^+(v)} f_e = 0 & v \in V \setminus \{s, t\} \quad & \text{(1b)}\\
& f_e \leq u(e) & e \in E \quad & \text{(1c)}\\
& f_e \geq 0 & e \in E \quad & \text{(1d)}
\end{aligned}
$$



Fernando Dias
fernando.dias@aalto.fi
Combinatorial Optimization
Aalto University