Week II

Overview

- Tree, Paths and Cycles;
- Shortest Path;
- Minimum Spanning Tree;
- Dijkstra's, Prim's and Kruskal's.

Definitions

- Shortest Path (SP): is the problem of finding a path between nodes in a graph such that the sum of the weights of its edges is minimized.
- Minimum Spanning Tree (MST): is a subset of the edges of a connected, weighted undirected graph that connects all nodes together, without cycles and with the minimum possible total edge weight.

Shortest Path ILP

$$\min \sum_{(u,v)\in E} f_{uv}x_{uv} \tag{1a}$$

subject to:

$$\sum_{(s,v)\in E} x_{sv} = 1, \tag{1b}$$

$$\sum_{(u,t)\in E} x_{ut} = 1, \tag{1c}$$

$$\sum_{(u,v)\in E} x_{uv} - \sum_{(v,w)\in E} x_{vwi} = 0, \tag{1d}$$

$$x_{uv} \in \{0,1\}, \qquad \forall (u,v) \in E \tag{1e}$$

Dijkstra's

Algorithm 1: DIJKSTRA'S ALGORITHM

Input: undirected, connected graph G, weights $c\colon E(G) \to \mathbb{R}$, nodes V, source s

```
d_v distance to reach node v
p_v node predecessor to node v
Q ← ∅ set of "unkown distance" nodes.
for each node v in V do
    d_v ← ∞
    p_v ← FALSE
    add v in Q
d_s ← 0 while Q ≠ ∅ do
    u ← node in Q with min d_u
    remove u from Q
    for each neighbor v of u still in Q do
        d ← d_u + c_uv
        if alt < d_v then
            d_v ← alt
            p_v ← u
```

Prim's

Algorithm 2: PRIM'S ALGORITHM

Input: undirected, connected graph G, weights $c\colon E(G) \to \mathbb{R}$

Output: spanning tree T of minimum weight

```
choose v ∈ V(G)
set T := ({v}, ∅)
while V(T) ≠ V(G) do
    choose an edge e ∈ δ_G(V(T)) of minimum
     weight
    set T := T + e
return T
```

Kruskal's

Algorithm 3: KRUSKAL'S ALGORITHM

Input: undirected, connected graph G, weights $c\colon E(G) \to \mathbb{R}$

Output: spanning tree T of minimum weight

```
sort edges such that c(e_1) ≤ c(e_2) ≤ ... ≤ c(e_m)
set T := (V(G), ∅)
for i := 1 to m do
    if T + e_i contains no cycle then
        set T := T + e_i
return T
```

MST ILP

$$\min \sum_{(u,v)\in E} f_{uv}x_{uv} \tag{2a}$$

subject to: (2b)

$$\sum_{(u,v)\in E} x_{uv} = n - 1, \tag{2c}$$

$$y^k_{uv} + y^k_{vu} = x_{uv}, \qquad (u,v) \in E, k \in V \tag{2d}$$

$$\sum_{k\in V\setminus\{(u,v)\}} y^v_{uk} + x_{uv} = 1, \qquad \forall (u,v) \in E \tag{2e}$$

$$x_{uv}, y^k_{uv}, y^k_{vu} \in \{0,1\}, \qquad \forall (u,v) \in E, k \in V \tag{2f}$$

Fernando Dias

fernando.dias@aalto.fi

Combinatorial Optimization

Aalto University