

TP5 - Making a Graph Eulerian

Renaud Chicoisne

24 Février 2023

Étant donné un graphe $G = (V, E)$ l'objectif de ce TP est d'implémenter en C une routine permettant de rendre G Eulérien (c.f. Algorithme 1).

Algorithm 1: Eulerianize

Data: A graph $G = (V, E)$ with edge weights $w \in \mathbb{R}^m$

Result: An Eulerian graph $G := (V, E')$ with $E \subseteq E'$ and $w(E')$ is minimum

```

1  $\mathcal{O} \leftarrow$  set of odd-degree nodes in  $G$ ;
2 for  $(u, v) \in \mathcal{O}^2$  do
3    $\mathcal{P}_{uv} \leftarrow$  shortest  $(u, v)$ -path in  $G$ ;
4    $l_{uv} \leftarrow l(\mathcal{P}_{uv})$ ;
5  $\mathcal{K} \leftarrow$  complete graph on  $\mathcal{O}$  with edge weights  $l$ ;
6  $\mathcal{M} \leftarrow$  minimum weight perfect matching in  $(\mathcal{K}, l)$ ;
7  $E' \leftarrow E$ ;
8 for  $(u, v) \in \mathcal{M}$  do
9    $E' \leftarrow E' \cup \mathcal{P}_{uv}$ ;
10 return  $G' = (V, E')$ ;
```

Un code source vous est fourni sur l'ent qui comporte deux fonctions d'intérêt pour ce TP:

- La fonction `dijkstra` qui calcule un arbre de plus courts chemins à partir d'un certain noeud (fourni)
- La fonction `min_weight_perfect_matching` qui calcule un couplage parfait de poids minimum (implémenté avec CPLEX, à terminer)

Calculer un couplage parfait de poids minimum utilisera en interne la librairie CPLEX. Étant donné un graphe complet $\mathcal{K} = (\mathcal{N}, \mathcal{A})$ et des poids d'arête l_{ij} , trouver un couplage parfait de poids minimum dans ce graphe est équivalent à résoudre le problème linéaire en nombres entiers suivant:

$$\begin{aligned}
& \min_x \sum_{(i,j) \in \mathcal{A}} x_{ij} l_{ij} \\
& \text{s.t.} \quad \sum_{j: (i,j) \in \mathcal{A}} x_{ij} = 1, & \forall i \in \mathcal{N} \\
& x_{ij} \in \{0, 1\}, & \forall (i,j) \in \mathcal{A}
\end{aligned}$$

Votre code devra lire un fichier d'instance qui contient toutes les informations du graphe considéré sous le format suivant:

1. La première ligne **n,m** contient le nombre de noeuds $n = |V|$ et d'arêtes $m = |E|$
2. Chacune des m lignes suivantes **i,j,c** contient l'information d'une arête $(i,j) \in E$ et son poids $c_{ij} = c$

Le code source fourni sur l'ent se compile simplement à l'aide de la commande **make**. Votre code devra retourner les informations relatives au Graphe Eulerien trouve dans un fichier **.csv** contenant en premiere ligne le nombre de noeuds et d'aretes n,m' et chacune des m' lignes suivantes les arcs du graphe $G' := (V, E')$: **i,j,c**.

Testez vos implémentations sur les réseaux donnés en instance sur l'espace de cours.