Projet Modélisation 2018

**Aucune entrée d'index n'a été trouvée.**

# Présentation du problème

Dans ce projet nous devions implémenter les algorithmes qui nous permettent de mettre en lumière les propriétés d’un graphe : connexité, cycle, cycle à valuations négatives. Un algorithme très pratique pour ce problème est le DFS. Il parcourt tout le graphe et note ainsi tous les arcs et leur direction. Il pourra donc détecter la présence d’un cycle et d’une composante connexe dans le graphe.

Grâce à ces propriétés, nous avons pu écrire les algorithmes de plus court chemin suivants : Dijkstra et Ford.

Dijkstra permet de trouver le plus court chemin (avec un coût par arc ou un temps, c’est au choix) grâce aux labels sur les arcs et sur les sommets. Cependant le graphe a pour contrainte de ne pas avoir de valuation négatives et pas de cycle non plus. C’est en cela que DFS est pratique. Un simple test permettra de savoir si l’utilisation de Dijkstra est possible ou non.

Dans la vie courante, ces algorithmes économiseraient beaucoup de temps et d’argent à ceux qui partent en voyage, par exemple, ou bien pour des sociétés de transports qui souhaitent livrer leur marchandise à moindre coût et/ou dans un laps de temps le plus petit possible.

# Les choix liés à la programmation

Pour créer le graphe, nous avons réutilisé les classes que nous avons étudié dans le cours de Représentations des Connaissances. La structure d’un graphe est élémentaire : il est composé d’une liste de sommets et d’une liste d’arcs, tous deux héritant de la classe Maillon<> qui est une classe patron permettant de créer des chaînes.

Dans Maillon<> la donnée membre valeur est un pointeur sur l’élément de la liste qui est étudié (Sommet ou Arc). La donnée membre suivant est un pointeur sur l’élément qui vient juste après, qui est un maillon du même ordre que le reste de la liste.

Mais ces classes Graphe, Arcs et Sommets ne permettent pas de lire les fichiers gpr qui nous ont été fournis. C’est la classe ChargerGraphe qui fournit ce travail. Une instance de cette classe va lire le fichier fourni grâce à son chemin et les données récupérées ligne par ligne vont conduire à la création d’un arc ou d’un sommet, le tout donnant bien sûr le graphe résultant de cette analyse. Rien de bien compliqué jusque là.

Les choix que nous avons fait pour l’implémentation des algorithmes DFS, Dijkstra et Ford sont un peu plus poussés. Les algorithmes fournis en cours sont facile à faire sur le papier mais les transformer en code concordant avec les classes déjà existantes dans le projet demandaient un peu plus de réflexion.