# Prosit 2 : Sans Complexe

## Mots à définir / mots Clés :

* Algorithme heuristique : Méthode de résolution de problèmes qui utilise des techniques basées sur l'expérience ou l'observation pour trouver une solution, souvent rapidement mais pas nécessairement optimale.
* NP : (non déterministe polynomial) : s'il existe un algorithme pour vérifier qu'une solution donnée convient en un temps polynomial.
* P : s'il existe un algorithme pour le résoudre en temps polynomial.
* NP complet : si la résolution de ce problème en temps polynomial entraîne la résolution en temps polynomial de tout problème NP.
* Complexité exponentielle / spatiale / temporelle / asymptotique : Mesures de la performance d'un algorithme. La complexité exponentielle indique que le temps ou l'espace requis par l'algorithme augmente exponentiellement avec la taille de l'entrée, tandis que la complexité temporelle asymptotique décrit le comportement de l'algorithme lorsque la taille de l'entrée tend vers l'infini.
* Machine de Turing : Un modèle abstrait de calcul utilisé en informatique théorique pour décrire le fonctionnement des algorithmes. Une machine de Turing peut effectuer des opérations de lecture, d'écriture et de déplacement sur une bande infinie selon un ensemble de règles prédéfinies.
* Problème du voyageur de commerce : Problème d'optimisation combinatoire qui consiste à trouver le plus court chemin permettant de visiter chaque ville d'un ensemble donné exactement une fois et de revenir au point de départ.
* Optimisation combinatoire : Domaine de l'informatique qui étudie les méthodes pour trouver la meilleure solution parmi un ensemble fini de solutions possibles.
* Algorithme de certificat / optimal : Un algorithme qui peut être utilisé pour vérifier la validité d'une solution proposée à un problème, ou qui peut trouver la meilleure solution possible à ce problème.
* Réduction polynomiale : Une transformation d'un problème en un autre de sorte que la solution du premier problème puisse être utilisée pour résoudre le second, avec une complexité polynomiale.
* Problème de décision :
* Calculateur moderne : Un ordinateur ou une machine utilisée pour effectuer des calculs et traiter des données, généralement basée sur une architecture électronique.

## Contexte :

Nous devons optimiser les tournées de livraisons à plus grande échelle avec plus de contraintes sur l’appel de l’ADEME.

## Problématique :

* Comment déterminer la complexité du problème ? (Itinéraire du projet)
* Comment certifier l’appartenance à NP ?

## Contraintes :

* Type de complexité
* Pas de code
* **Démontrer** l’appartenance à un type de complexité
* Avoir une tournée en sortie

## Généralisation :

* Démontrer et calculer une complexité algorithmique
* Classes de problèmes

## Livrable :

Démonstration du type de complexité

## Pistes de solution :

* Passer par chaque route != passer par chaque ville ?
* Même complexité que le pb du voyageur de commerce ?
* Représentation pas eulérienne ni hamiltonienne ?
* Comprendre la tournée en sortie

## Plan d’actions :

* Lire des articles scientifiques sur le problème du VDC
* Comprendre les différentes complexités
* Faire la corbeille
* Déterminer le type d’algo
* Déterminer le type de complexité

## Démonstration :

Base du problème : On cherche à passer par chaque ville une et une seul fois en effectuant le trajet le plus court possible il nous faut également revenir au point de départ.

Données : Graphe G, arrête/value, et k l’entier représentant la somme du parcours minimale

Question : Existe-t-il un circuit passant une fois et une seul fois pour chaque sommet de G dont la somme des arrêtes et au plus égale à k ?

Qu’elle est la valeur finale de k ?

On cherche à montrer que le problème du Voyageur de Commerce et NP-complet :

Pour nous aider dans notre résolution, on peut voir que le VDC se rapproche d’un cycle hamiltonien, la seul différence est que le graphe doit être complet et que chaque arrête possède une valeur.

On se servira donc du cycle hamiltonien qui est un problème NP pour le réduire au VDC, On peut définir G :

Une image contenant cercle, ligne

Description générée automatiquement

La réduction vers le VDC serait :

Ajout du graphe complet :

Une image contenant cercle, ligne

Description générée automatiquement

Ajout des poids sur chaque arrêtes ( comme les branches orange ne sont pas existé nous allons mettre une « grosse valeur » pour être sur de ne pas passer par là nous mettrons au reste des branches) :

Une image contenant cercle, ligne, diagramme

Description générée automatiquement

L’algorithme de résolution serait :

* Vérifier que la suite de sommets soit correcte : que les arrêtes existe
* Si elle passe par une unique fois chaque sommet
* Vérifier si le cout du circuit est bien inférieur à k

La solution se résout en temps linéaire P, comme on sait que le circuit hamiltonien est de complexité NP-complet.

On a alors NP-Complet pour le VDC, on ne peut pas trouver de solution optimale en temps polynomial.