

# BLOC Algorithmique avancée

# Contexte :

Après avoir prouvé que notre problème est NP complet, on cherche maintenant une methode de résolution sachant qu’il est impossible de produire une solution optimale en temps polynomial.

# Mots à définir / mots Clés : (lien avec recherche opérationnelle)

La "Recherche Opérationnelle" est une discipline qui utilise des méthodes mathématiques et informatiques pour prendre des décisions optimales dans des situations complexes. Elle vise à maximiser les avantages ou minimiser les coûts, en prenant en compte des contraintes.

Le "Simplexe" est une méthode de résolution de problèmes de programmation linéaire. Il consiste à déplacer un point à travers les sommets d'un polyèdre afin de trouver le point optimal qui maximise ou minimise une fonction objective.

Une "Métaheuristique" est une méthode d'optimisation qui guide la recherche d'une solution dans un espace de recherche complexe. Contrairement aux algorithmes exacts, les métaheuristiques n'offrent pas de garantie de trouver la meilleure solution, mais elles sont souvent plus rapides et peuvent être appliquées à des problèmes difficiles.

La "Programmation Linéaire et Dynamique" est une méthode qui combine la programmation linéaire avec des éléments de la programmation dynamique pour résoudre des problèmes d'optimisation. Elle est utilisée dans des situations où les décisions doivent être prises séquentiellement et où les actions futures dépendent des actions précédentes.

L'"Approche Adaptative" est une méthode qui ajuste automatiquement ses stratégies ou ses paramètres en fonction des changements dans l'environnement ou des performances observées. Elle est utilisée dans de nombreux domaines, y compris l'optimisation, pour s'adapter à des situations variables et améliorer les résultats au fil du temps.

# Problématique :

**Comment choisir une méthode de résolution algorithmique pour résoudre le problème**

# Contraintes :

-Problème de grande taille

-Facile à implémenter

# Généralisation :

Recherche Opérationnelle

Methode de résolution de problème d’optimisation difficile

# Livrable :

Choisir l’algorithme

Implémenter

Tester deux algorithmes

# Hypothèses / Pistes de solution :

Représenter le problème d’optimisation comme un ensemble de sous problèmes

Programmation linéaire et pas dynamique

Système d’équation et inéquation

# Plan d’actions :

Se renseigner sur les différents algorithme métaheuristique et simplexe

Découper le problème en petit problème

Faire une methode qui donne une valeur exacte à petite échelle puis une autre methode à grande échelle.

Tester les deux algorithmes

# Réalisations du plan d’actions :

Rappel VDC :

* N villes
* Passer une seul et unique fois par cille tout en minimisant la distance totale parcourue

Graphe 1 orienté (n=6) :

Une image contenant ligne, horloge

Description générée automatiquement

## Modélisation :

Soit un graphe orienté G=(V,U) V=sommets (villes), U=arcs (routes)

Une image contenant texte, Police, capture d’écran, blanc

Description générée automatiquement

Xij=1 si l’arc (i,j) est dans le circuit hamiltonien

Xij=0 sinon

Contraintes 1 : Dans chaque ville, i il rentre un arc (le voyageur rentre dans i)

Contraintes 2 : Dans chaque ville i, il sort un arc (le voyageur sort de i)

Les contraintes 1 et 2 ne sont pas optimales car elles peuvent donner des sous-tours, si on reprend le graphe 1 :

Une image contenant ligne, Police

Description générée automatiquement

On peux donc poser l’inégalité suivante pour résoudre le problème :

Une image contenant texte, Police, blanc, ligne

Description générée automatiquement

Soit :

Une image contenant Police, écriture manuscrite, diagramme, ligne

Description générée automatiquement

## Implémentation python :

J’ai donc essayé de faire une implémentation python du VDC qui fonctionne sur 4 villes. En m’inspirant du workshop

Une image contenant texte, capture d’écran

Description générée automatiquement

Ca fonctionne sur 4 villes mais pas sur 5, sans que je ,ne sache pourquoi.

Code dispo sur mon github