

**Prosit 4 : pas de déjeuner gratuit**

# Mots clés

- Métaheuristique : Une métaheuristique est une méthode d'optimisation itérative qui guide et modifie le processus de recherche d'une solution pour des problèmes complexes, souvent combinatoires. Elle est conçue pour trouver de bonnes solutions en un temps raisonnable, sans garantir la solution optimale.

Hill Climbing : Le Hill Climbing est une méthode de recherche locale qui part d'une solution initiale et explore les solutions voisines pour trouver une meilleure solution. L'objectif est de toujours minimiser ou maximiser la solution. Elle intensifie la recherche autour des solutions actuelles mais peut se retrouver coincée dans des optima locaux. Pour éviter cela, la diversification peut être utilisée en permettant des déplacements vers des solutions non optimales.

Heuristique Glouton :Un heuristique glouton (ou gourmande) est une méthode de construction de solutions qui fait des choix optimaux locaux à chaque étape dans l'espoir de trouver une solution optimale globale. Elle ne garantit pas toujours une solution optimale mais est souvent efficace et rapide.

Étape d’intensification**:** L'étape d'intensification vise à explorer plus profondément une région prometteuse de l'espace de recherche, souvent en concentrant la recherche autour des meilleures solutions trouvées jusqu'à présent.

Test statistique :Un test statistique est une méthode utilisée pour déterminer si les résultats d'une expérience ou d'une analyse sont significatifs. Il permet de vérifier des hypothèses et de prendre des décisions basées sur des données.

Étape de diversification :L'étape de diversification vise à élargir la recherche pour explorer de nouvelles régions de l'espace de recherche, en évitant de se concentrer uniquement sur les solutions locales déjà trouvées, afin de ne pas rester bloqué dans un optimum local.

Optimum local :Un optimum local est une solution qui est meilleure que toutes les solutions voisines mais qui n'est pas nécessairement la meilleure solution globale (optimum global).

Modélisation du voisinage :La modélisation du voisinage consiste à définir un ensemble de solutions voisines pouvant être atteintes à partir de la solution actuelle par de petites modifications. C'est une étape clé dans les méthodes de recherche locale.

Métaheuristique par trajectoire ou par population :Les métaheuristiques par trajectoire (comme le Hill Climbing, le Recuit Simulé) suivent une seule solution à travers l'espace de recherche, tandis que les métaheuristiques par population (comme les Algorithmes Génétiques, les Colonies de Fourmis) maintiennent et mettent à jour un ensemble de solutions simultanément.

Algorithme génétique :Un algorithme génétique est une métaheuristique inspirée de la sélection naturelle et de l'évolution biologique. Il utilise des opérations comme la sélection, le croisement (crossover) et la mutation pour évoluer une population de solutions vers une meilleure solution.

Méthode par construction :Les méthodes par construction construisent une solution étape par étape en ajoutant des éléments à une solution partielle jusqu'à obtenir une solution complète.

Méthode de recherche perturbative :Une méthode de recherche perturbative modifie une solution existante de manière significative pour explorer de nouvelles régions de l'espace de recherche.

Méthode de recherche locale :Une méthode de recherche locale améliore une solution initiale en explorant les solutions voisines dans l'espoir de trouver une solution meilleure. Elle se concentre sur les déplacements locaux dans l'espace de recherche.

Algorithme du tabou :L'algorithme du tabou est une méthode de recherche locale qui utilise une mémoire pour éviter de revisiter des solutions déjà explorées (tabou) et pour guider la recherche vers de nouvelles régions de l'espace de recherche.

Recuit simulé :Le recuit simulé est une méthode d'optimisation inspirée du processus de recuit en métallurgie. Elle permet des mouvements vers des solutions moins bonnes au début (pour échapper aux optima locaux), puis réduit progressivement cette tolérance au fil du temps.

GRASP (Greedy Randomized Adaptive Search Procedure) :GRASP est une métaheuristique qui combine des constructions gloutonnes aléatoires et des améliorations locales pour trouver des solutions de haute qualité.

VRP (Vehicle Routing Problem) géométrique :Le VRP géométrique est une variante du problème de tournées de véhicules où les distances et les trajets sont basés sur des relations géométriques, souvent utilisées pour des problèmes de logistique et de transport.

# Colonie de fourmi : La colonie de fourmi est une métaheuristique inspirée du comportement des fourmis cherchant de la nourriture. Elle utilise des agents (fourmis) pour explorer des solutions et utilise des phéromones pour guider les futures explorations.

# Rugosité du voisinage : La rugosité du voisinage décrit la complexité et la variabilité de l'espace de recherche autour d'une solution. Un voisinage rugueux a beaucoup de variations locales, ce qui peut rendre difficile la recherche de l'optimum global.

# Paramètre de génération : Les paramètres de génération sont des variables utilisées pour contrôler la création de nouvelles solutions dans des algorithmes comme les algorithmes génétiques (par exemple, taux de mutation, taille de la population).

# Plan d’expérience : Un plan d'expérience est une méthode structurée pour concevoir et réaliser des expériences de manière à analyser les effets de différents facteurs et à optimiser les résultats.

# Contexte

On a abandonné la programmation linéaire nous essayons de résoudre le problème à l’aide des métaheuristiques. Nous recherchons une méthode métaheuristique en vue de résoudre le problème.

# Problématique

**Comment concevoir une méthode de résolution à l’aide des métaheuristiques pour le problème du TSP ?**

# Contrainte

Résolution rapide

# Généralisation

Algorithme métaheuristique

Plan d’expérimentation

# Livrable

Choix de la méthode métaheuristique

Programme de résolution du problème

Plan d’expérimentation

Pistes de solution :

Trouver le ratio précision – temps d’exécution en faisant varier la solution

Déterminer un critère de précision

Comparer les solutions entre elles

Calculer borne inférieur et supérieur (utilisation solveur), pire et meilleure solution optimale

# Plan d’action

Lister les métaheuristiques (description, différence, …)

Choisir la plus adapter (génétique, fourmi, glouton)

Justifier le choix (source, biblio, …)

Implémenter la méthode

Plan d’expérimentation (comportement de l’algorithme, test , évaluation de la méthode, qualité et temps d’exécution).

Génération d’instance

# Réalisation du plan d’action

En reprenant la recherche taboue et le worksop j’ai pu établir un script python dispo sur mon git (tabou.py).

# Cours

## Introduction aux Métaheuristiques

Les métaheuristiques sont des algorithmes d'optimisation qui peuvent trouver des solutions approximatives à des problèmes complexes en un temps raisonnable. Elles sont souvent utilisées lorsque les méthodes exactes sont impraticables (trop longues).

Nous allons voir quatre types de métaheuristiques : les algorithmes génétiques, les colonies de fourmis, les algorithmes gloutons et l'ascension de colline (hill climbing), en mettant l'accent sur la résolution du problème du voyageur de commerce (TSP).

## 1. Algorithmes Génétiques

Les algorithmes génétiques (AG) sont inspirés de la théorie de l'évolution biologique. Ils utilisent des mécanismes tels que la sélection, le croisement et la mutation pour évoluer vers des solutions optimales.

### Étapes principales :

Initialisation : Génération d'une population initiale de solutions.

Évaluation : Calcul de la qualité (fitness) de chaque solution.

Sélection : Choix des meilleures solutions pour la reproduction.

Croisement : Combinaison de paires de solutions pour créer une nouvelle génération.

Mutation : Modification aléatoire de certaines solutions pour maintenir la diversité génétique.

### Avantages :

Capacité à explorer un vaste espace de solutions.

Évite les minima locaux grâce à la diversité de la population.

### Inconvénients :

Peut nécessiter beaucoup de temps de calcul.

La qualité des solutions peut stagner si les paramètres ne sont pas bien réglés.

## 2. Algorithme de la Colonie de Fourmis

L'algorithme de la colonie de fourmis (ACO) s'inspire du comportement des fourmis cherchant de la nourriture et déposant des phéromones pour guider les autres fourmis.

### Étapes principales :

Initialisation : Placement des fourmis sur les nœuds de départ.

Construction de solutions : Chaque fourmi construit une solution en suivant des règles probabilistes basées sur les phéromones et l'heuristique locale.

Mise à jour des phéromones : Les fourmis déposent des phéromones proportionnellement à la qualité de leurs solutions.

Évaporation des phéromones : Réduction des traces de phéromones pour éviter la convergence prématurée.

### Avantages :

Capacité à trouver de bonnes solutions dans des environnements dynamiques.

Bon équilibre entre exploration et exploitation.

### Inconvénients :

Sensible aux paramètres de phéromones et aux heuristiques locales.

Peut être coûteux en termes de temps de calcul.

## 3. Algorithme Glouton

Un algorithme glouton construit une solution de manière séquentielle en choisissant à chaque étape l'option locale optimale.

### Étapes principales :

Initialisation : Départ d'un point de départ choisi.

Construction : Ajout itératif des éléments à la solution actuelle en choisissant l'option la plus prometteuse à chaque étape.

Finalisation : Obtention de la solution finale lorsque toutes les étapes sont complétées.

### Avantages :

Simplicité et rapidité d'exécution.

Bonne performance pour certains problèmes spécifiques.

### Inconvénients :

Risque élevé de coincer dans des minima locaux.

Ne garantit pas une solution optimale globale.

## 4. Ascension de Colline (Hill Climbing)

L'ascension de colline est une méthode itérative qui commence par une solution initiale et effectue des modifications successives pour améliorer la solution actuelle.

### Étapes principales :

Initialisation : Choix d'une solution initiale.

Recherche locale : Modification de la solution pour améliorer la valeur de l'objectif.

Boucle d'amélioration : Répétition des modifications jusqu'à ce qu'aucune amélioration supplémentaire ne soit possible.

### Avantages :

Simple à implémenter et rapide.

Peut fournir de bonnes solutions rapidement pour des problèmes de grande taille.

### Inconvénients :

Peut facilement se coincer dans des minima locaux sans techniques additionnelles comme la recherche tabou.

Hill Climbing et Recherche Tabou pour le TSP

Pour la résolution du TSP, l'ascension de colline combinée avec la recherche tabou se révèle souvent très efficace. La recherche tabou ajoute une mémoire à court terme pour éviter de revisiter les mêmes solutions et sortir des minima locaux.

## Étapes de la recherche tabou :

Initialisation : Départ d'une solution initiale.

Recherche locale avec mémoire : Modification de la solution en respectant les restrictions de la liste tabou.

Mise à jour de la liste tabou : Enregistrement des mouvements interdits temporairement.

## Avantages combinés :

Évite les minima locaux grâce à la mémoire tabou.

Rapide et efficace pour les grandes instances du TSP.

## Conclusion :

Pour le problème du voyageur de commerce, l'ascension de colline, surtout lorsqu'elle est combinée avec la recherche tabou, offre un excellent compromis entre simplicité, rapidité et qualité des solutions. Tandis que les algorithmes génétiques et les colonies de fourmis peuvent explorer un plus grand espace de solutions, ils nécessitent souvent plus de temps de calcul et un réglage précis des paramètres. Les algorithmes gloutons, bien qu'efficaces pour certains problèmes, manquent souvent de flexibilité pour le TSP.