

Algorithme de Résolution du Problème de Routage de Véhicules (VRP)

1. Introduction

Le **Problème de Routage de Véhicules (VRP)** est un problème d'optimisation combinatoire qui a pour but de déterminer les itinéraires optimaux pour un ensemble de véhicules devant livrer des clients dispersés sur un territoire. Ce problème est soumis à des contraintes telles que le nombre de véhicules disponibles et la capacité des véhicules, et vise à minimiser la distance totale parcourue ou le coût associé aux trajets.

Ce document présente une solution basée sur un algorithme heuristique pour résoudre le VRP, prenant en compte la demande des clients et leur disposition géographique. L'objectif principal est d'affecter les clients aux véhicules et de déterminer un ensemble de trajets, en optimisant les déplacements tout en respectant les contraintes liées au nombre de véhicules.

2. Description de l'Algorithme

2.1. Approche de l'Algorithme

L'algorithme proposé suit une approche simplifiée de **routage des véhicules**, où les clients sont assignés aux véhicules de manière itérative. Les étapes principales incluent :

1. **Initialisation des routes** : Les véhicules sont initialisés avec des routes vides, prêtes à accueillir les clients.
2. **Mélange aléatoire des clients** : Une permutation aléatoire des clients est effectuée afin d'éviter les biais d'un traitement systématique et pour améliorer l'efficacité de la solution.
3. **Assignation des clients** : Chaque client est affecté à un véhicule, tant que la demande du client peut être satisfaite sans dépasser la capacité restante du véhicule.
4. **Changement de véhicule** : Si un client ne peut être ajouté à la route d'un véhicule en raison de contraintes, un autre véhicule est sélectionné pour recevoir ce client, et ainsi de suite jusqu'à ce que tous les clients soient affectés à des véhicules.

2.2. Fonctionnement des Routes

Chaque véhicule est associé à un itinéraire constitué de clients, et ces itinéraires sont calculés en fonction de la disponibilité et de la demande des clients. L'algorithme vise à minimiser les trajets en réorganisant la liste des clients de façon efficace, tout en assurant que les contraintes liées aux capacités sont respectées.

L'algorithme ne procède à aucune optimisation avancée des trajectoires (par exemple, par des méthodes de type *2-opt* ou *3-opt*). Il se contente d'une affectation simple basée sur un critère de demande et de capacité.

3. Détails Techniques de l'Algorithme

3.1. Structure des Données

Les clients sont représentés sous forme d'un dictionnaire où chaque clé correspond à l'identifiant d'un client, et la valeur associée contient la demande ainsi que les coordonnées géographiques du client.

Les véhicules sont représentés par des listes qui contiennent les clients affectés à chaque véhicule. Ces listes sont mises à jour au fur et à mesure que les clients sont affectés à un véhicule.

3.2. Calcul des Distances

Les distances entre deux clients sont calculées en utilisant la formule de la distance euclidienne, qui permet d'évaluer la longueur du trajet entre deux points dans un espace à deux dimensions. Ce calcul est essentiel pour la construction du graphe de VRP, car il permet de mesurer le coût des trajets entre les clients.

Formule de la distance euclidienne :

$$d(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

où $d(a,b)$ représente la distance entre les points a et b , et x et y sont les coordonnées des points respectifs dans le plan.

3.3. Affichage du Résultat

Une fois que les clients ont été attribués aux véhicules, un graphique représentant le réseau de transport est généré. Ce graphique visualise les itinéraires des véhicules, ainsi que les distances entre les différents clients. Les clients sont représentés par des nœuds, et les trajets entre eux par des arêtes, où le poids de chaque arête correspond à la distance entre les clients. Ce graphique est généré à l'aide de la bibliothèque NetworkX et de matplotlib.

Le résultat final est affiché à l'utilisateur sous la forme d'un graphe qui illustre l'optimisation des trajets, avec les distances associées aux arcs du graphe.

4. Complexité de l'Algorithme

L'algorithme proposé est un algorithme de type **greedy** (glouton), qui ne garantit pas nécessairement la solution optimale, mais offre une solution suffisamment bonne dans des délais raisonnables pour des instances de taille modérée.

- **Complexité en temps** : L'algorithme itère sur les clients, en les affectant à des véhicules. Pour chaque client, une vérification de la capacité du véhicule est effectuée, ce qui donne une complexité de l'ordre de $O(n^2)$, où n est le nombre de clients.
- **Complexité en espace** : L'espace nécessaire pour stocker les routes des véhicules est proportionnel au nombre de clients et de véhicules, soit $O(n+v)$, où n est le nombre de clients et v le nombre de véhicules.

Bien que l'algorithme ne garantisse pas une solution optimale, il reste efficace pour des problèmes de taille modérée et peut être amélioré par des techniques de recherche locale ou des heuristiques plus sophistiquées.

Cette approche heuristique offre une méthode simple et efficace pour résoudre le **Problème de Routage de Véhicules** avec des contraintes de capacité. Bien qu'elle ne prenne pas en compte des optimisations complexes, elle constitue une base solide pour des applications pratiques dans des scénarios réels de logistique et de transport.