

**Rapport de projet — Master**

# **Multi-Product Vehicle Routing Problem with Changeover Cost**

Présenté par :

Irené LOKOSSOU	10 %
Juste TOSSOU	10 %
Méschac HOUESSOU	10 %
Samuel AKABASSI	10 %
Mavic PATINVOH	10 %
Omella	10 %

Encadrant : Dr Ratheil HOUNJII

IFRI - UAC

Année académique : 2025–2026

# Table des matières

<b>Résumé</b>	<b>ii</b>
<b>Introduction</b>	<b>1</b>
<b>1 Présentation du problème</b>	<b>2</b>
1.1 Description générale . . . . .	2
<b>2 Modélisation du problème</b>	<b>3</b>
2.1 Données du problème . . . . .	3
<b>3 Résolution</b>	<b>4</b>
3.1 Présentation du solveur utilisé . . . . .	4
3.1.1 OR-Tools CP-SAT : Un solveur de pointe . . . . .	4
3.2 Architecture du code . . . . .	4
3.2.1 Structure modulaire en 5 parties principales . . . . .	4
3.3 Analyse des résultats par catégorie d’instances . . . . .	6
3.3.1 Instance <i>small</i> . . . . .	6
3.3.2 Analyse argumentée des résultats de l’instance . . . . .	7
3.4 Instance <i>medium</i> . . . . .	8
3.4.1 Caractéristiques principales . . . . .	8
3.4.2 Résolution avec OR-Tools CP-SAT . . . . .	8
3.4.3 Résultats opérationnels . . . . .	8
3.4.4 Analyses argumentées des résultats obtenus . . . . .	9
3.5 Instance <i>large</i> . . . . .	9
3.5.1 Caractéristiques principales . . . . .	9
3.5.2 Résolution avec OR-Tools CP-SAT . . . . .	10
3.5.3 Résultats opérationnels . . . . .	10
3.5.4 Analyses argumentées des résultats obtenus . . . . .	10
<b>Conclusion et perspectives</b>	<b>12</b>
<b>Bibliographie</b>	<b>13</b>

# Résumé

Cet rapport ...

# Table des figures

# Liste des tableaux



# Introduction

Le Vehicle Routing Problem (VRP) est un problème central en logistique...

# Chapitre 1

## Présentation du problème

### 1.1 Description générale

Le MPVRP-CC consiste à desservir un ensemble de clients avec un parc de véhicules, chaque client demandant potentiellement plusieurs produits...

# Chapitre 2

## Modélisation du problème

### 2.1 Données du problème

Notations principales ...

## Résolution

### 3.1 Présentation du solveur utilisé

#### 3.1.1 OR-Tools CP-SAT : Un solveur de pointe

OR-Tools (Google Optimization Tools) est une suite open-source développée par Google pour résoudre des problèmes d'optimisation complexes.

**Caractéristiques techniques :**

- Type : Solveur par programmation par contraintes (CP-SAT)
- Algorithme : Recherche arborescente avec propagation de contraintes
- Parallélisation : Support multi-threads (4 workers)
- Heuristiques : Incorporation de solutions initiales (Hints)
- Garanties : Trouve des solutions optimales prouvées

### 3.2 Architecture du code

#### 3.2.1 Structure modulaire en 5 parties principales

**Chargement des données (load\_instance)**

La première étape consiste à **charger l'instance du problème** depuis un fichier `.dat`.

Les données extraites sont :

- le nombre de **produits, stations, véhicules, dépôts** et **garages**
- les **coûts de transition entre produits**
- les **capacités des véhicules**
- les **demandes par station et par produit**
- les **coordonnées géographiques** des entités

**Analyse préalable de faisabilité**

Avant toute résolution, une **analyse de faisabilité** est réalisée :

- calcul de la **demande totale**
- calcul de la **capacité totale disponible**

Cette étape permet :

- de détecter rapidement les cas **impossibles**
- d'interpréter correctement les résultats du solveur
- d'expliquer d'éventuelles situations d'infaisabilité dans le rapport

### Construction d'une solution heuristique initiale

Afin d'améliorer les performances du solveur exact, une **solution heuristique initiale** est construite.

#### Principe de l'heuristique :

- chaque véhicule conserve son **produit initial**
- il dessert les **stations les plus proches** demandant ce produit
- il livre jusqu'à **épuisement de sa capacité**

#### Rôle de cette étape :

- produire une **solution faisable rapidement**
  - fournir une **borne supérieure initiale**
  - guider le solveur CP-SAT grâce au mécanisme de **hint**
- Cette solution n'est pas optimale, mais elle est réaliste et exploitable.

### Modélisation mathématique avec CP-SAT

#### Variables de décision

- **Variables de livraison** : quantité livrée par un véhicule à une station pour un produit donné
- **Variables binaires de visite** : indiquent si un véhicule visite une station

Ces variables permettent de modéliser à la fois :

- les quantités transportées
- la structure des tournées

#### Contraintes du modèle

Les principales contraintes sont :

1. **Satisfaction complète de la demande** : chaque demande (station, produit) doit être entièrement satisfaite par l'ensemble des véhicules
2. **Capacité des véhicules** : la somme des quantités livrées par un véhicule ne peut pas dépasser sa capacité
3. **Lien livraison–visite** : un véhicule est considéré comme visitant une station s'il y livre une quantité strictement positive

Ces contraintes garantissent la **faisabilité logistique** de la solution.

#### Fonction objectif

La fonction objectif vise à **minimiser le coût total**, composé de :

1. **Coût de transport** : distance entre le garage du véhicule et les stations visitées
2. **Coût de changement de produit (changeover)** : pénalité lorsqu'un véhicule livre un produit différent de son produit initial
3. **Coût de consolidation** : coût fixe par visite afin de limiter le nombre de stations visitées et favoriser des tournées plus compactes

### Intégration de la solution heuristique (Hint)

La solution heuristique construite précédemment est injectée dans le modèle sous forme de **point de départ**.

#### Avantages :

- accélération de la recherche
- amélioration de la qualité des premières solutions

- réduction du temps de calcul
- Le solveur reste libre d'améliorer ou de modifier cette solution.

### Résolution avec OR-Tools CP-SAT

- Le solveur est ensuite lancé avec :
- une **limite de temps**
  - une **recherche multi-threads**
  - un **critère d'écart relatif** pour accepter une solution quasi-optimale
- À l'issue de la résolution, le solveur retourne :
- une solution **optimale, faisable**, ou
  - un statut d'**infaisabilité** clairement identifié

### Analyse et validation des résultats

- La solution obtenue est analysée afin de :
- vérifier que **toutes les demandes sont satisfaites**
  - mesurer le **taux d'utilisation des véhicules**
  - évaluer le **coût total**
  - comparer la solution finale à la solution heuristique initiale
- Cette étape permet de justifier la qualité de la solution.

### Visualisation des résultats

- Une visualisation graphique est générée pour :
- représenter les **garages, dépôts et stations**
  - afficher les **routes par véhicule**
  - illustrer les **quantités livrées**

### Sauvegarde et synthèse finale

- Enfin, la solution est :
- sauvegardée dans un fichier de sortie
  - accompagnée d'un **résumé chiffré** :
    - coût total
    - nombre de routes
    - taux de satisfaction de la demande
    - taux d'utilisation des capacités

## 3.3 Analyse des résultats par catégorie d'instances

### 3.3.1 Instance *small*

Ici le texte est effectué sur de petite instance (small instance). La commande de test est :  
python solver.py instances/small.dat solution.dat

### Présentation des résultats de l'instance

**Description de l'instance** L'instance catégorie *small* se caractérise par une taille réduite, permettant de valider le bon fonctionnement du modèle et du solveur.

- Nombre de stations : 5
- Nombre de produits : 2
- Nombre de véhicules : 2

- Nombre de dépôts : 1
- Nombre de garages : 1

Cette instance représente un cas simple mais représentatif, dans lequel chaque station peut demander un ou plusieurs produits.

**Analyse de faisabilité a priori** Avant la résolution, une analyse de faisabilité est effectuée afin de vérifier l'adéquation entre la demande globale et la capacité totale disponible.

- Demande totale : 6 900 unités
- Capacité totale des véhicules : 9 000 unités

La capacité totale étant supérieure à la demande, l'instance est théoriquement faisable.

- Marge de capacité : 2 100 unités
- Ratio demande / capacité : 0,77

**Résolution avec OR-Tools CP-SAT** La résolution est effectuée à l'aide du solveur OR-Tools CP-SAT, basé sur la programmation par contraintes et la recherche SAT.

- **Variables de décision** : 2 véhicules  $\times$  5 stations  $\times$  2 produits = 20 variables principales
- **Contraintes** :
  - 7 contraintes de satisfaction de la demande
  - 10 contraintes liant livraison et visite
  - 2 contraintes de capacité des véhicules
- **Fonction objectif** : minimisation du coût total (distance + coûts de changement de produit), composée de 30 termes

La phase de modélisation est réalisée en 0,003 seconde, ce qui montre une construction efficace du modèle.

**Solution heuristique initiale** Une solution heuristique est construite afin de fournir un point de départ au solveur CP-SAT.

- **Véhicule 1 (Produit initial 1, capacité 5000)** : dessert 3 stations, capacité utilisée : 2 500 / 5 000
- **Véhicule 2 (Produit initial 2, capacité 4000)** : dessert 4 stations, capacité utilisée : 4 000 / 4 000

Le coût estimé de cette solution initiale est de **18 937**, et **7 hints** sont fournis au solveur.

**Résultats de la résolution** Le solveur CP-SAT atteint une solution **OPTIMALE**.

- Statut : OPTIMAL
  - Valeur de l'objectif : 13 418
  - Temps de résolution : 0,058 s
  - Temps total (modélisation + résolution) : 0,061 s
  - Amélioration par rapport à la solution heuristique initiale : 5 519 unités soit 29,1 %
- Toutes les demandes sont intégralement satisfaites.

### 3.3.2 Analyse argumentée des résultats de l'instance

#### Qualité de la solution obtenue

La solution trouvée est optimale et respecte l'ensemble des contraintes du problème. Le solveur parvient à réduire significativement le coût total par rapport à la solution heuristique initiale, ce qui démontre l'efficacité de l'exploration de l'espace des solutions par CP-SAT.

### Utilisation des capacités des véhicules

Les capacités des véhicules sont exploitées de manière équilibrée :

- Véhicule 1 : 64 % de capacité utilisée
- Véhicule 2 : 92,5 % de capacité utilisée

Cette répartition permet de satisfaire l'ensemble des demandes tout en limitant les surcharges inutiles.

## 3.4 Instance *medium*

L'instance *medium* du problème MPVRP-CC se caractérise par une taille intermédiaire, permettant d'évaluer à la fois la qualité de la modélisation et les performances du solveur. Pour la tester, on utilise la commande :

```
python solver.py instances/medium.dat solution.dat
```

### 3.4.1 Caractéristiques principales

- Stations : 12
- Produits : 3
- Véhicules : 4
- Dépôt : 1
- Garages : 2
- Demande totale : 19 100 unités
- Capacité totale des véhicules : 21 000 unités
- Ratio demande/capacité : 0,91

Cette configuration garantit la faisabilité théorique du problème tout en conservant une complexité suffisante pour tester l'efficacité du solveur.

### 3.4.2 Résolution avec OR-Tools CP-SAT

- Solveur utilisé : OR-Tools CP-SAT
- Temps limite : 30 s
- Temps effectif de résolution : 5,85 s
- Statut final : OPTIMAL
- Valeur fonction objectif : 27 396

Une solution heuristique initiale a été générée avant la résolution exacte :

- Coût heuristique initial : 103 951
- La solution optimale finale représente une amélioration de 73,6 %

### 3.4.3 Résultats opérationnels

- 100 % des demandes des stations sont satisfaites
- 4 routes générées, correspondant aux 4 véhicules
- Taux d'utilisation des capacités :
  - Véhicule 1 : 98,3 %
  - Véhicule 2 : 85,5 %
  - Véhicule 3 : 100 %
  - Véhicule 4 : 77,8 %

Une visualisation graphique des routes a été produite afin de faciliter l'interprétation spatiale de la solution.

### 3.4.4 Analyses argumentées des résultats obtenus

#### Qualité de la solution

La solution obtenue est **optimale**, ce qui démontre que la modélisation CP-SAT est **correcte et complète**. Toutes les contraintes (demande, capacité, compatibilité produits/véhicules) sont respectées sans exception. L'amélioration significative par rapport à la solution heuristique initiale montre que :

- l'heuristique joue efficacement son rôle de **point de départ**
- le solveur CP-SAT exploite cette base pour explorer rapidement des solutions de meilleure qualité

#### Performance du solveur OR-Tools CP-SAT

Le solveur atteint l'optimalité en moins de 6 secondes, bien en-dessous de la limite imposée (30 s). Cela met en évidence :

- la robustesse du moteur CP-SAT
- l'efficacité des stratégies internes (LNS, LP relaxation, local search, feasibility pump)

La diversité des sous-solveurs activés a permis une exploration rapide et équilibrée de l'espace de recherche.

#### Répartition des charges et routes

La charge est globalement bien répartie :

- Aucun véhicule n'est sous-utilisé de manière critique
- Un véhicule atteint même 100 % de capacité, indiquant une bonne exploitation des ressources

Les routes restent relativement courtes (2 à 4 stations par véhicule), ce qui contribue à :

- réduire les coûts
- simplifier l'exécution opérationnelle

## 3.5 Instance *large*

L'instance large du problème MPVRP-CC correspond à un cas de grande taille, destiné à évaluer la robustesse de la modélisation ainsi que les performances du solveur OR-Tools CP-SAT dans un contexte fortement combinatoire.

Pour le tester, on utilise la commande suivante :

```
python solver.py instances/large.dat solution.dat
```

### 3.5.1 Caractéristiques principales

- Stations : 20
- Produits : 2
- Véhicules : 6
- Dépôts : 2
- Garages : 3
- Demande totale : 24 210 unités
- Capacité totale des véhicules : 40 500 unités
- Ratio demande/capacité : 0,60

Cette configuration garantit une faisabilité confortable du problème, avec une marge de capacité importante, tout en introduisant une complexité élevée due au nombre de stations, de véhicules et aux livraisons multi-produits.

### 3.5.2 Résolution avec OR-Tools CP-SAT

- Solveur : OR-Tools CP-SAT
- Temps limite : 30 s
- Temps effectif de résolution : 2,83 s
- Temps total (modélisation + résolution) : 2,85 s
- Statut final : OPTIMAL
- Valeur de la fonction objectif : 45 772

Avant la résolution exacte, une solution heuristique initiale a été construite et injectée dans le solveur et son coût heuristique initial était : 123 836. La solution optimale finale représente une amélioration de 63,0 % par rapport à cette solution de départ.

### 3.5.3 Résultats opérationnels

- 100 % des demandes des stations sont satisfaites, pour l'ensemble des produits.
- 5 routes générées, correspondant à 5 véhicules effectivement utilisés sur les 6 disponibles.
- Taux d'utilisation des capacités :
  - Véhicule 1 : 66,9 %
  - Véhicule 2 : 38,6 %
  - Véhicule 3 : 92,9 %
  - Véhicule 5 : 99,8 %
  - Véhicule 6 : 58,2 %

Les routes desservent entre 1 et 6 stations, avec des livraisons mono-produit et multi-produits selon les stations. Une visualisation graphique des routes a été générée afin de faciliter l'analyse spatiale et la validation visuelle de la solution (*visualisation\_large.png*).

### 3.5.4 Analyses argumentées des résultats obtenus

#### Qualité de la solution

La solution obtenue est optimale, ce qui confirme la validité et la complétude de la modélisation CP-SAT du MPVRP-CC. Toutes les contraintes sont strictement respectées :

- satisfaction exacte des demandes,
- respect des capacités des véhicules,
- cohérence entre visite d'une station et livraison effective,
- prise en compte des coûts de changement de produit

L'amélioration significative par rapport à la solution heuristique initiale montre que :

- l'heuristique joue efficacement son rôle de point de départ faisable,
- le solveur CP-SAT exploite cette base pour réduire rapidement l'espace de recherche et atteindre une solution optimale.

#### Performance du solveur OR-Tools CP-SAT

Le solveur atteint l'optimalité en moins de 3 secondes, largement en dessous de la limite imposée de 30 secondes. Cette performance met en évidence :

- la robustesse du moteur CP-SAT face à une instance de grande taille,
- l'efficacité des stratégies internes activées automatiquement, notamment :
  - Large Neighborhood Search (LNS),
  - relaxations linéaires (default\_lp),
  - feasibility pump,
  - recherche locale et redémarrages contrôlés.

La diversité des sous-solveurs a permis une exploration rapide, équilibrée et progressive de l'espace de recherche, comme l'illustre la succession de 37 solutions améliorantes avant l'optimalité.

### Répartition des charges et routes

La répartition des charges n'est pas parfaitement équilibrée, mais reste cohérente d'un point de vue économique :

- certains véhicules sont fortement exploités (jusqu'à **99,8 %**),
- d'autres sont moins sollicités lorsque cela permet de réduire le coût global.

Un véhicule n'est pas utilisé, ce qui indique que le solveur préfère minimiser les coûts plutôt que mobiliser inutilement toute la flotte disponible. Les routes restent de taille raisonnable (jusqu'à 6 stations par véhicule), ce qui contribue à :

- limiter les distances et coûts cumulés,
- conserver une exécution opérationnelle réaliste,
- gérer efficacement les livraisons multi-produits.

# Conclusion et perspectives

Pour conclure ...

## Bibliographie