



Recherche opérationnelle pour l'optimisation des flux de brancardiers à l'hôpital

PROJET TUTORÉ DE 5ÈME ANNÉE

CARBASSE Camille - DIAZ Quentin - EL HARCHAOUI Amine

Équipe



CARBASSE Camille



DIAZ Quentin



EL HARCHAOUI Amine

Sommaire

1

INTRODUCTION

2

MÉTHODE
HEURISTIQUE

3

MÉTHODE
EXACTE

4

MÉTHODE
MÉTAHEURISTIQUE

5

RÉSULTATS

6

AXES
D'AMÉLIORATION

7

GESTION DE
PROJET

8

CONCLUSION

Introduction



Enjeux et Objectif

Comparaison de différentes approches, méthodes et outils de recherche opérationnelle

Méthodes heuristiques

Programme fourni
par Atout Majeur
Concept

Méthodes mét-heuristiques

Outil OptaPlanner

Méthodes exactes

Programmation
linéaire
Utilisation de CPLEX

Brancardage

Etape transversale du flux patient

Les déplacements internes au sein du service (transferts entre les boxes et les salles d'attente)



Les transports liés aux plateaux médico-techniques (salles d'opération, imagerie, etc.)

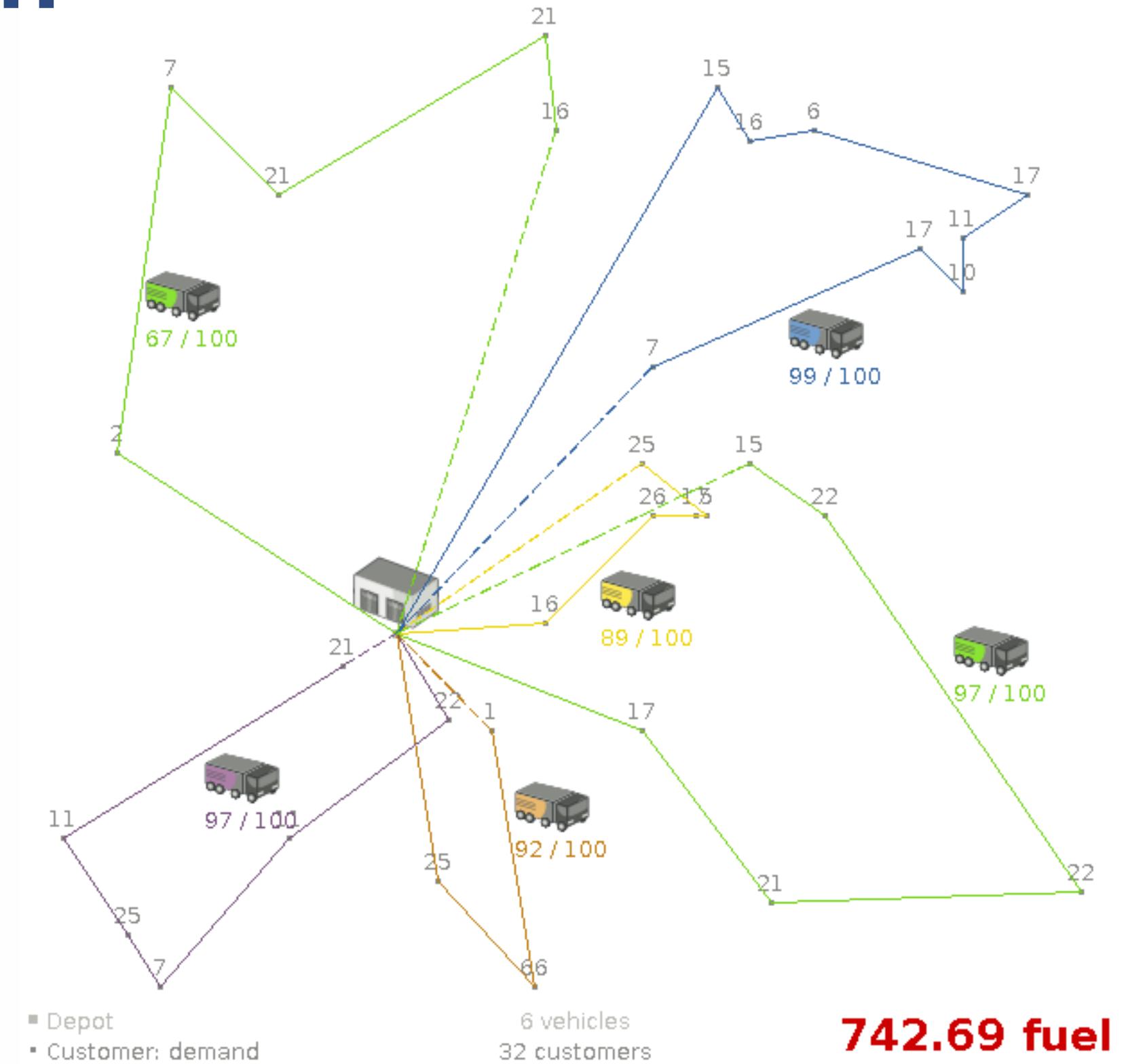
Les transports vers les services cliniques d'hospitalisation



Véhicle Routing Problem

« PROBLÈME DE TOURNÉES DE VÉHICULES ».

- Problème classique d'optimisation
- Applications dans de nombreux domaines
- Résolution du VRP est un défi complexe
- Résolution grâce à différentes méthodes



Grand principes à respecter

Mission

Caractéristiques des missions
(Heure de départ et d'arrivée,
durée, etc).

Plusieurs types de missions.

Priorité

Système

Synchroniser les missions à deux brancardiers.

Limiter les retards des brancardiers et leurs déplacements inutiles.

Imposer une exécution chronologique des missions.



Brancardier

Respecter les horaires des
brancardiers.

Respecter la charge de travail
journalière maximale des
brancardiers.

Homogénéiser l'affectation de
missions

Méthode heuristique



Méthode heuristique

Méthodes d'optimisation approximative

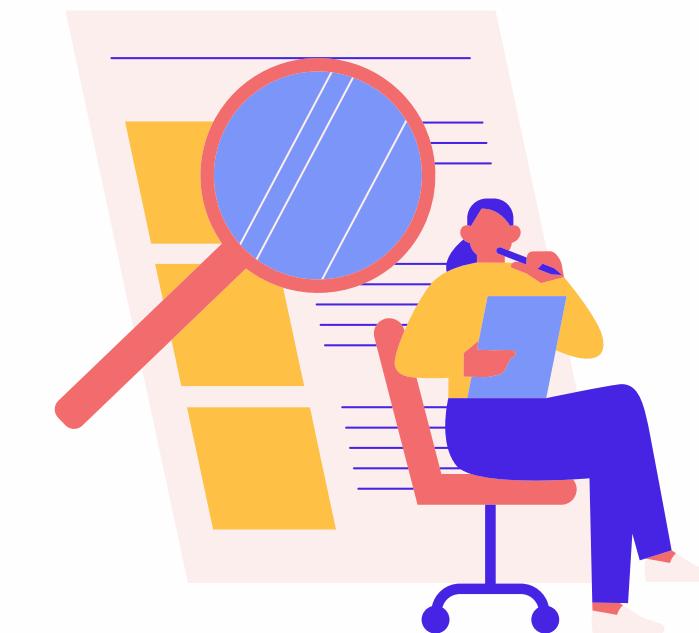
Solution existante utilisée par AMC



Codée en Python



Documentation



Respect des
principes

Méthode exacte



Programmation linéaire

Technique mathématique

Solution la plus optimale

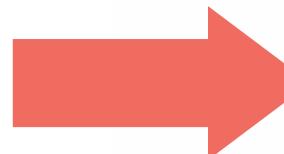
Situations de maximisation ou de minimisation d'une fonction

Repose sur des variables de décision

Améliorations apportées

Paramètres et variables

Paramètres	
N	Nombre de missions
N_1	Nombre de missions nécessitant 1 brancardier
N_2	Nombre de missions nécessitant 2 brancardiers
B	Nombre de brancardiers
D_i	Durée estimée de la mission i
$D_{moyen\ ij}$	Durée moyenne à vide entre la mission i et j
$Trdv_i$	Heure de rendez-vous de la mission i
$Hdeb_{mat_k}$	Heure de début de travail matinal pour le brancardier k
$Hdeb_{apm_k}$	Heure de début de travail de l'après-midi pour le brancardier k
$Hfin_{mat_k}$	Heure de fin de travail matinal pour le brancardier k
$Hfin_{apm_k}$	Heure de fin de travail de l'après-midi pour le brancardier k
Tra_{max_k}	Temps de travail maximal pour le brancardier k
M	Un grand nombre
R_i	Retard maximal accepté



Variables	
$Tdep_{ik}$	Heure de départ de la mission i pour le brancardier k
d_k	Temps de travail/nombre de missions attribuées au brancardier k
t_k^+	Pénalité positive sur le temps de travail d'un brancardier k
t_k^-	Pénalité négative sur le temps de travail d'un brancardier k
x_{ij}^k	1 sur le brancardier k effectue le trajet entre la mission i et la mission j, sinon 0
$late_i^k$	Le retard du brancardier k à la mission i

Paramètres		Domaine de définition
N	Nombre de missions	Entier
N_1	Nombre de missions nécessitant 1 brancardier	Entier
N_2	Nombre de missions nécessitant 2 brancardiers	Entier
B	Nombre de brancardiers	Entier
D_i	Durée estimée de la mission i	Entier
$D_{moyen\ ij}$	Durée moyenne à vide entre la mission i et j	Réel
Hrv_i	Heure de rendez-vous de la mission i	Entier
$Hdeb_{mat_k}$	Heure de début de travail matinal pour le brancardier k	Entier
$Hdeb_{apm_k}$	Heure de début de travail de l'après-midi pour le brancardier k	Entier
$Hfin_{mat_k}$	Heure de fin de travail matinal pour le brancardier k	Entier
$Hfin_{apm_k}$	Heure de fin de travail de l'après-midi pour le brancardier k	Entier
Tra_{max_k}	Temps de travail maximal pour le brancardier k	Réel
M	Un grand nombre	Entier
R_i	Retard maximal accepté pour la mission i	Réel

Variables	
$Hdep_{ik}$	Heure de départ de la mission i pour le brancardier k
x_{ij}^k	Brancardier k enchaîne la mission i et la mission j
y_i^k	Brancardier k effectue la mission i
$late_i^k$	Retard du brancardier k à l'heure prévue de début de mission i
m_k	Nombre de missions attribuées au brancardier k
$charge_k$	Charge temporelle du brancardier k
$charge\ moy$	Charge moyenne du pool des brancardiers
pST_i^k	Pointeur de sous tournée

Améliorations apportées

Fonction objectif

$$Z = \min \left(\sum_{i=0}^N \sum_{j=0}^N \sum_{k=1}^B D_{moyenij} * x_{ij}^k + \sum_{i=0}^N \sum_{k=1}^B (late_i^k) + \sum_{k=1}^B (t_k^- + t_k^+) \right)$$

$$Z = \min \left(\sum_{i=0}^N \sum_{k=1}^B (late_i^k) \right)$$

Améliorations apportées

Contraintes

$$Hdep_i^k \geq Hrdv_i \times y_i^k \&\& (Hdep_i^k \leq (Hrdv_i + R_i) \times y_i^k$$

Encadre l'heure de départ pour une mission i en tenant compte de la tolérance du retard maximal accepté pour cette mission.

Garantie qu'il n'y a pas de chevauchement entre deux missions consécutives, assurant ainsi un enchaînement sans interruption.

$$(Hdep_i^k + D_i + D_{moyenij}) + (x_{ij}^k - 1) * M \leq Hdep_j^k$$

CPLEX D'Ibm

Outil d'optimisation mathématique

The image shows a software interface with two tabs open. The left tab is titled "modele.mod" and contains OPL 22.1.0.0 Model code. The right tab is titled "data.dat" and contains OPL 22.1.0.0 Data code. Both tabs have a header with file name and creation date.

```
modele.mod x
1 ****
2 * OPL 22.1.0.0 Model
3 * Author: nrogers Carbasse Pingaud
4 * Creation Date: 17 avril 2023 at 10:59
5 ****
6
7 /*----- Set of parameters --
8
9 int NbMissions = ...; // Nombre de missions
10 int NbMissions1 = ...; // Nombre de missions à 1 brancardier
11 int NbBrancardiers = ...; // Nombre de brancardiers
12 range N = 0..NbMissions; // Ensemble des missions y compris la mission
13 range B = 1..NbBrancardiers; // Ensemble des brancardiers
14 range N0= 1..NbMissions;
15 range N1 = 1..NbMissions1; // ensemble des missions à un brancardier
16 range N2 = NbMissions1+1..NbMissions; // ensemble des missions à deux brancardiers
17 int Hrdv[N] = ...; // Heure de rdv des missions
18 int Duree[N] = ...; // Duree estimee des missions
19 int Hdeb_mat[B] = ...; // Duree minimale des missions
20 int Hfin_mat[B] = ...; // Duree maximale des missions
21 int Hdeb_apm[B] = ...; // Duree minimale des missions
22 int Hfin_apm[B] = ...; // Duree maximale des missions
23 float TraMax[B] = ...; // tps de travail maximal d'un brancardier
24 float Dist[N][N] = ...; // Duree moyenne entre les missions (matrice)
25 int M = 100000;
26 float R[N] = ...;
27
28 /*----- Set of variables -
29 // Variables de temps
30
31 dvar boolean v[N0][B]; // Choix du brancardier k pour une mission
```

```
data.dat x
1 ****
2 * OPL 22.1.0.0 Data
3 * Author: cacar
4 * Creation Date: 2 juin 2023 at 09:01:38
5 ****
6 SheetConnection donnees("missions.xlsx");
7 SheetConnection brancardiers("brancardiers.xlsx");
8 SheetConnection matrice("matrice.xlsx");
9 NbMissions from SheetRead(donnees, "G2"); // Nb de missions
10 NbMissions1 from SheetRead(donnees, "H2"); // Nb de missions à 1 brancardier
11 NbBrancardiers from SheetRead(donnees, "F2"); // Nb de brancardiers
12
13 Duree from SheetRead(donnees, "B2:B82"); // Duree moyenne estimee des missions
14 Hrdv from SheetRead(donnees, "C2:C82"); // heures de rdv des missions
15 R from SheetRead(donnees, "E2:E82"); // Retard max toléré
16
17 Hdeb_mat from SheetRead(brancardiers, "F2:F13"); // Heure de début d'un brancardier
18 Hfin_mat from SheetRead(brancardiers, "H2:H13"); // Heure de fin d'un brancardier
19 Hdeb_apm from SheetRead(brancardiers, "J2:J13"); // Heure de début d'un apm
20 Hfin_apm from SheetRead(brancardiers, "L2:L13"); // Heure de fin d'un apm
21 TraMax from SheetRead(brancardiers, "N2:N13"); // Charge max
22
23 Dist from SheetRead(matrice, "C3:CE83"); // tps de trajet à vide entre deux missions
24
25
26 SheetConnection result("Resultats_CPLEX.xlsx");
27 b to SheetWrite(result, "A1");
28 md to SheetWrite(result, "B1");
29 ti to SheetWrite(result, "C1");
30 hrdvi to SheetWrite(result, "D1");
31 ma to SheetWrite(result, "E1");
```

Résolution de problèmes de programmation linéaire à grande échelle

Vérification du modèle mathématique

Optimisation pour fournir des résultats rapides et précis

Méthode métaheuristique

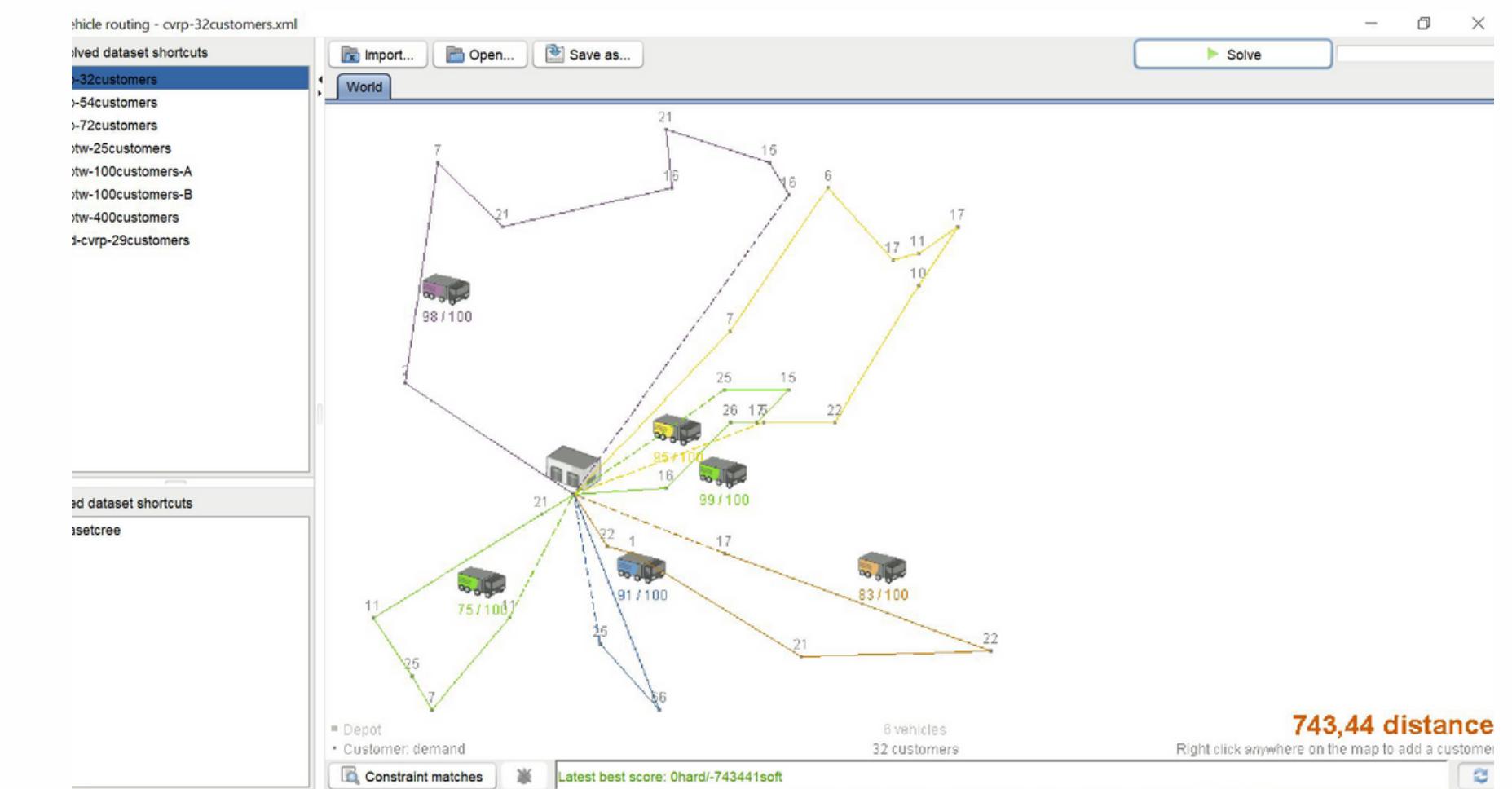


OptaPlanner 

Présentation d'Optaplanner



Interface graphique



Deux solutions : CVRP & CVRPTW

Les classes importantes

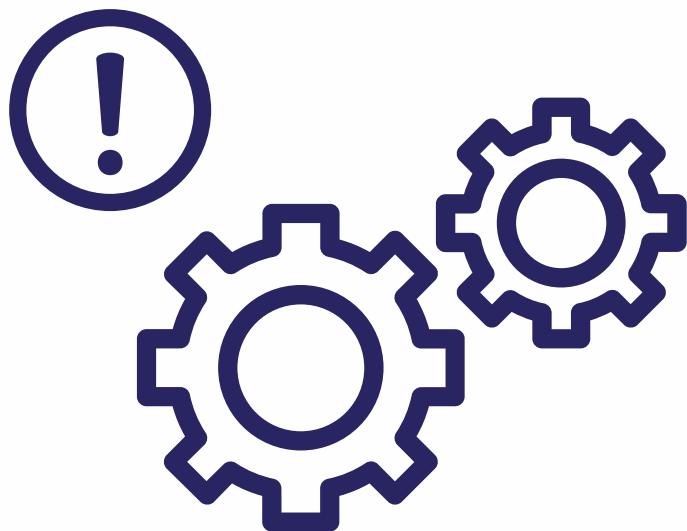
identification des éléments complexes du VRP



TimeWindowed
Customer



VehicleRouting
Solution

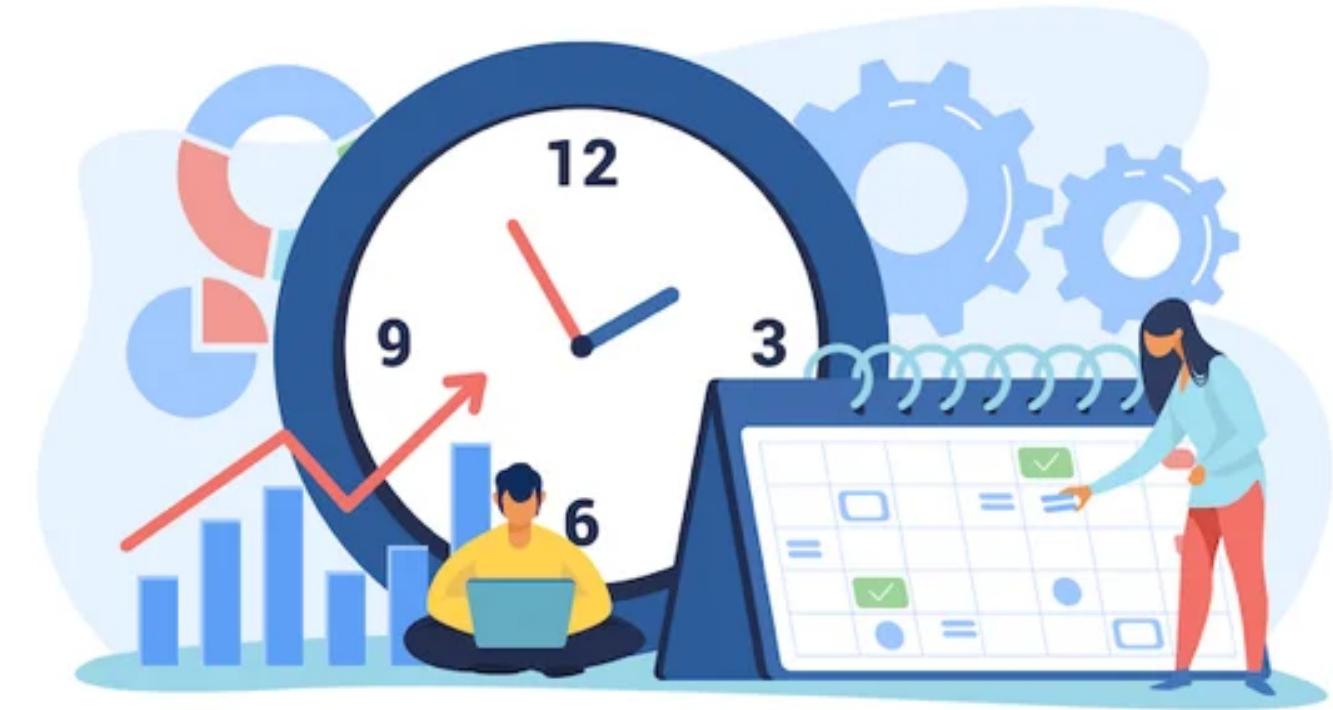


VehicleRouting
Constraint
Provider

Notion spécifique à l'outil

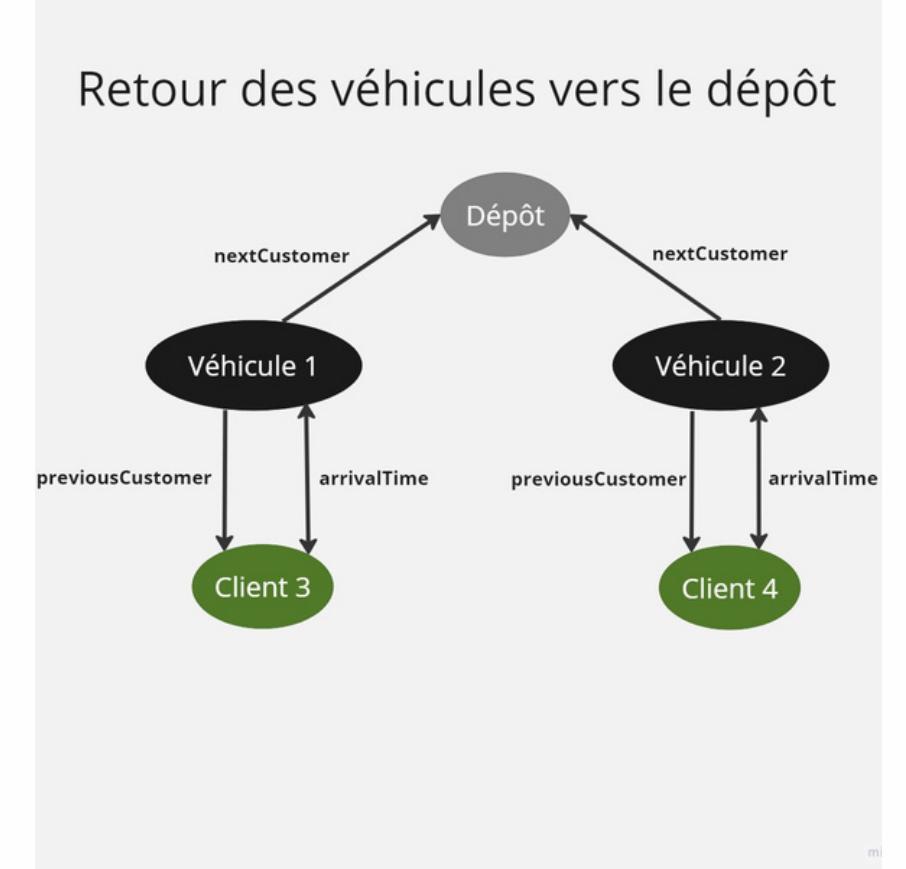
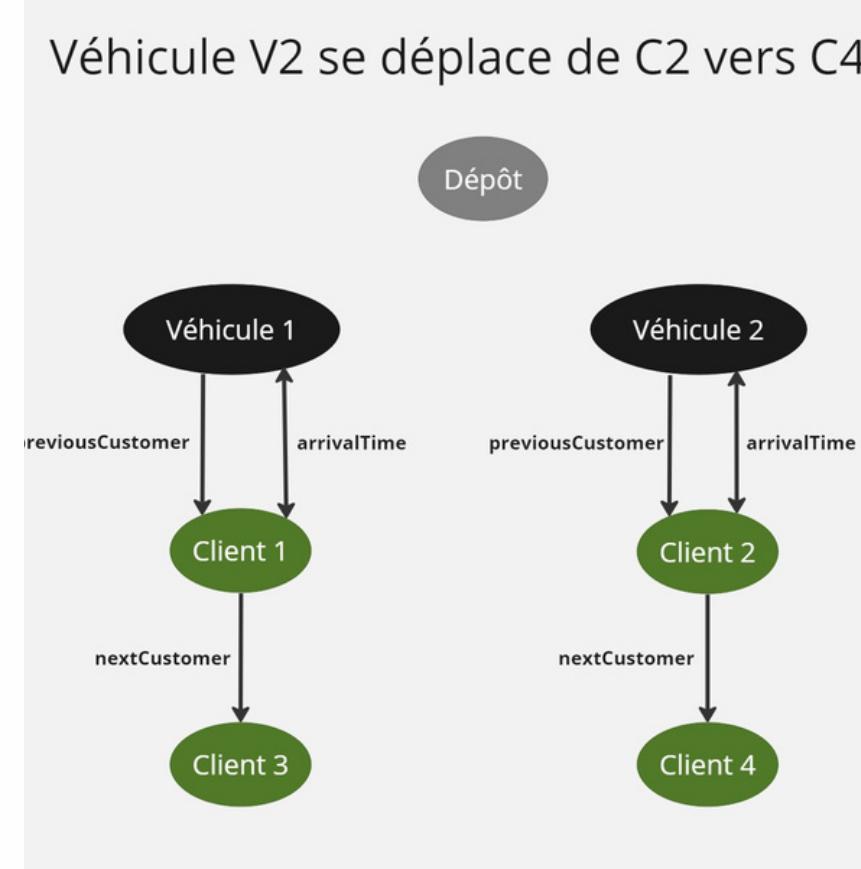
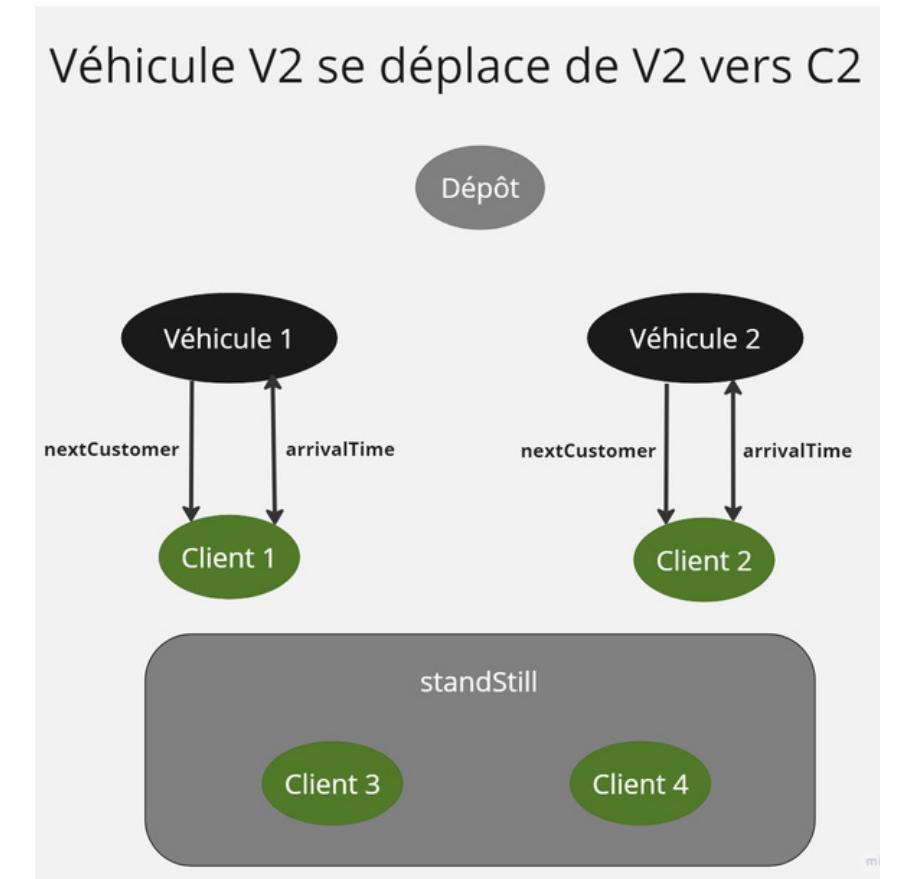
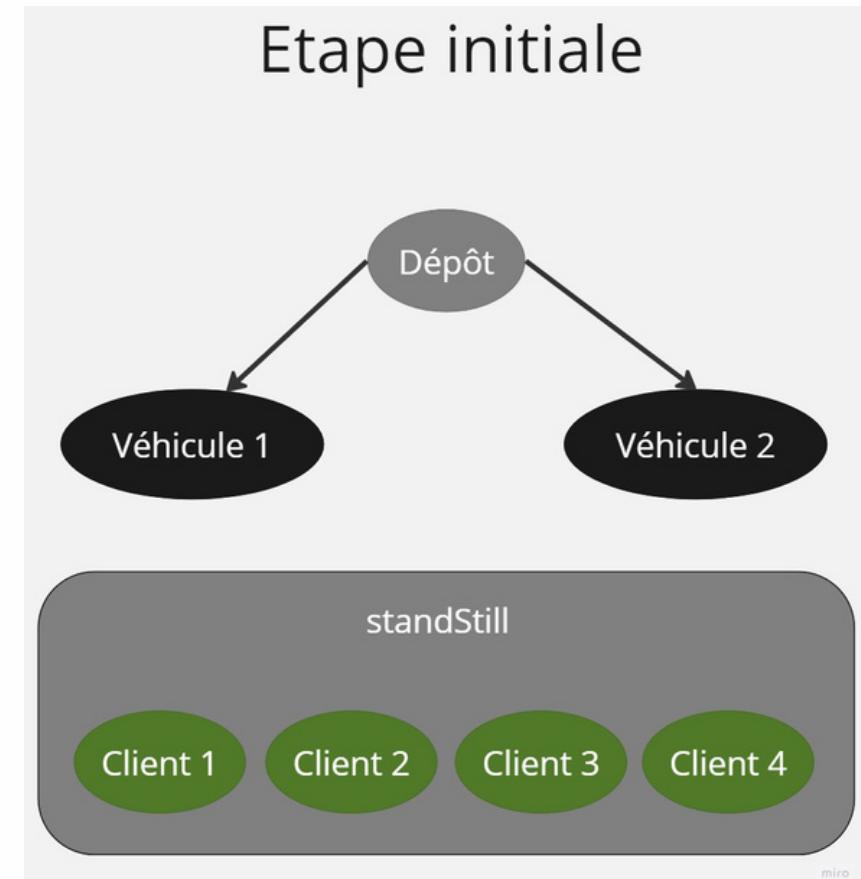
VARIABLE DE PLANIFICATION (PLANING VARIABLE)

- Algorithme d'optimisation
- Vise à améliorer la solution
- Recherche la configuration optimale



Notions spécifiques à l'outil

Planning variable exemple



Notions spécifiques à l'outil

Spécification des contraintes

Contrainte Hard

Respectée

Contrainte Medium

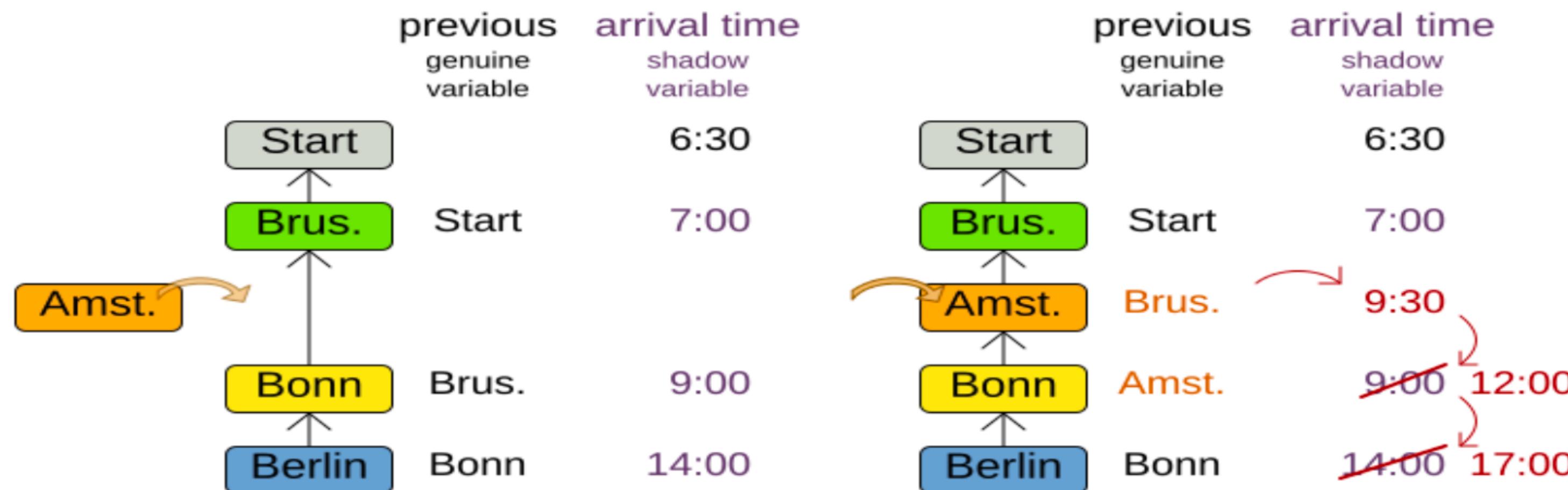
Flexibilité

Contrainte Soft

Peut ne pas être
respectée

Notion spécifique à l'outil

VARIABLE D'OMBRE (SHADOW VARIABLE)



Fonction objectif

- Contrainte de distance entre clients (soft)

```
softScore -= customer.getDistanceFromPreviousStandstill(); |
```

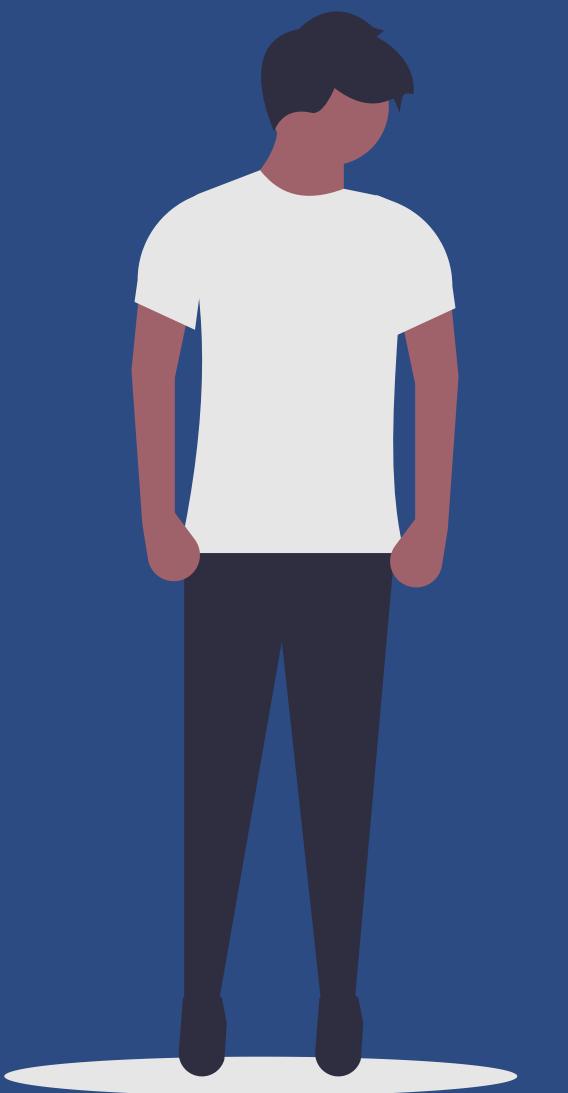
- Contrainte de distance avec le dépôt (soft)

```
if (customer.getNextCustomer() == null) {  
    // Score constraint distanceFromLastCustomerToDepot  
    softScore -= customer.getLocation().getDistanceTo(vehicle.getLocation());  
}
```

- Contrainte de temps (hard)

```
if (timeWindowed) {  
    TimeWindowedCustomer timeWindowedCustomer = (TimeWindowedCustomer) customer;  
    long dueTime = timeWindowedCustomer.getDueTime();  
    Long arrivalTime = timeWindowedCustomer.getArrivalTime();  
    if (dueTime < arrivalTime) {  
        // Score constraint arrivalAfterDueTime  
        hardScore -= (arrivalTime - dueTime);  
    }  
}
```

Résultats



Programmation linéaire

Comparaison avec les anciens résultats

Temps

Amélioration du temps
de calcul

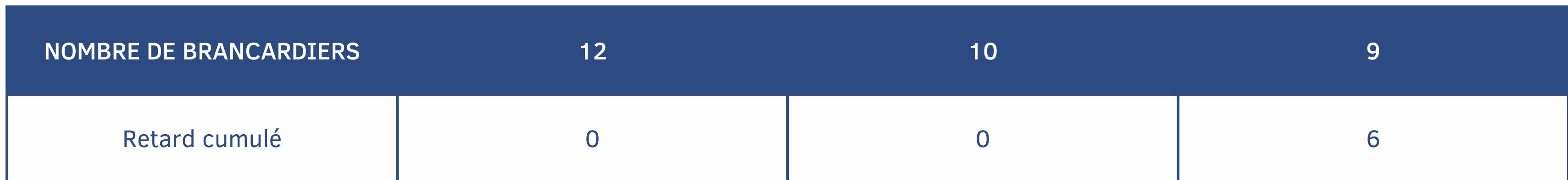
Résultats

Limite du modèle à 8 brancardiers

Résultats pour 9, 10 et 12 brancardiers

Programmation linéaire

Comparaison entre plusieurs jeux de données



Indicateurs de retard

Programmation linéaire

Comparaison entre plusieurs jeux de données

NOMBRE DE BRANCARDIER	12	10	9
Charge moyenne	180	211	228
Charge maximale	238	308	329
Charge minimum	113	159	135
Ecart moyen	30,4	32,9	35,8

Indicateurs sur la charge de travail temporel en minutes

Programmation linéaire

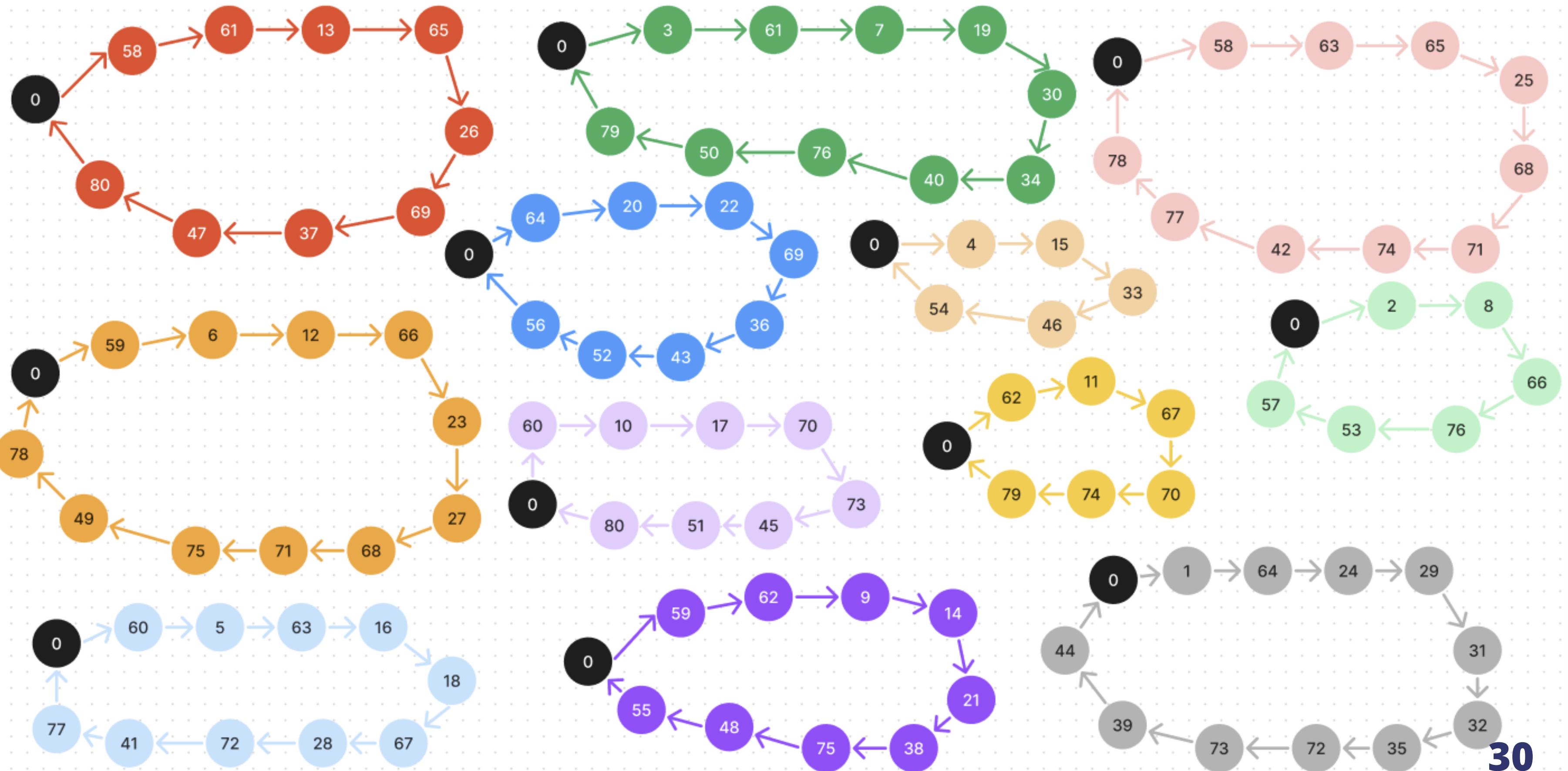
Comparaison entre plusieurs jeux de données

NOMBRE DE BRANCARDIER	12	10	9
Charge moyenne	8,5	10	11
Charge maximale	11	13	14
Charge minimum	5	6	9
Ecart moyen	1,7	2,1	1,2

Indicateurs sur la charge de travail en nombre de mission

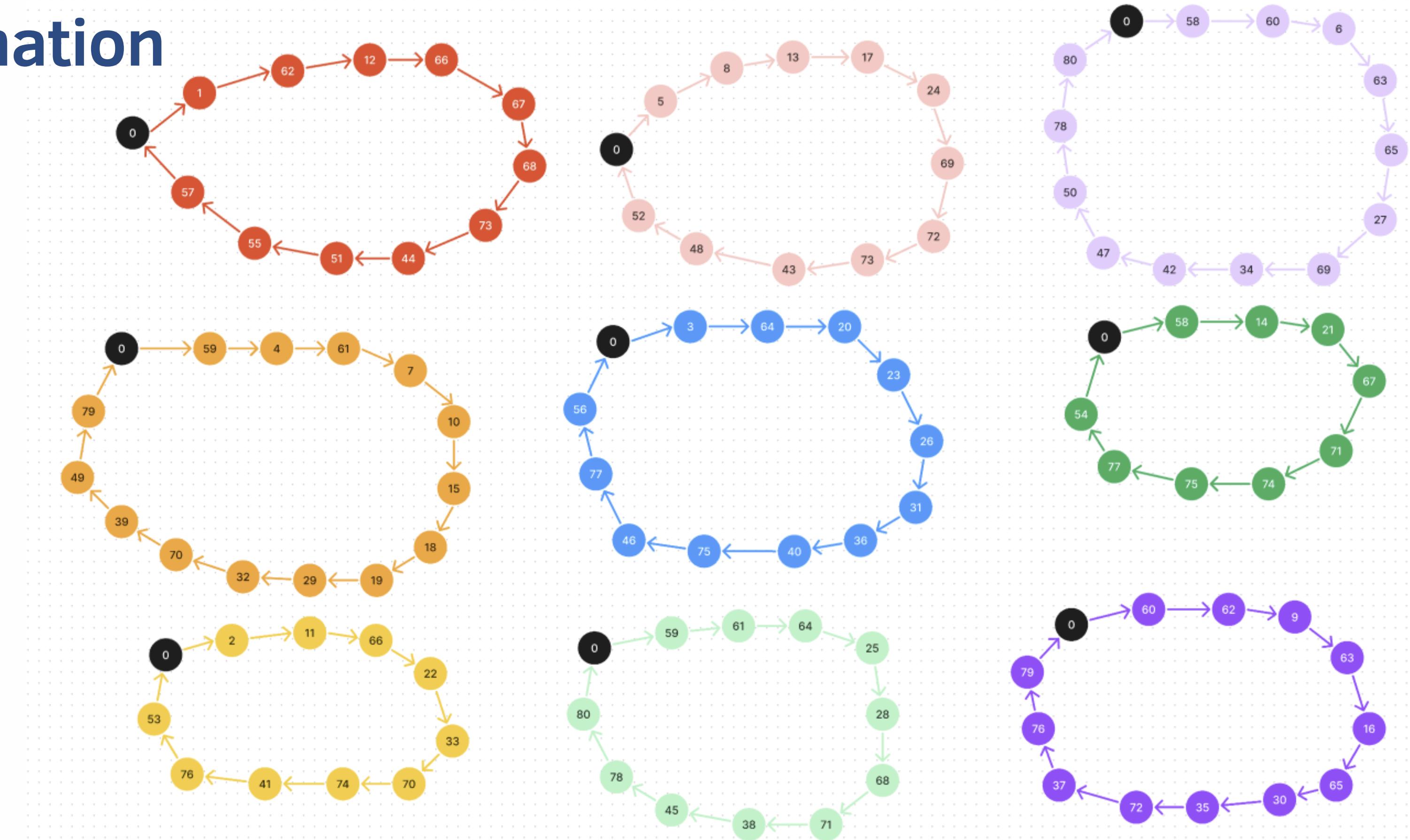
Programmation linéaire

Modélisation graphique



Programmation linéaire

Modélisation graphique



Méthode métaheuristique

Comparaison de plusieurs jeux de données sur OptaPlanner

- Même jeux de données que sur CPLEX
- Même fenêtre de temps

Temps

Entre 10 secondes et 5 minutes

Résultats

Résultats pour 12, 15 et 25 brancardiers
Limite à 10 brancardiers

Méthode métaheuristique

Contraintes techniques

Contraintes

- Notion de capacité : charge et demande
- Absence de coordonnées dans le jeu de données
- Aucune notion de double brancardiers

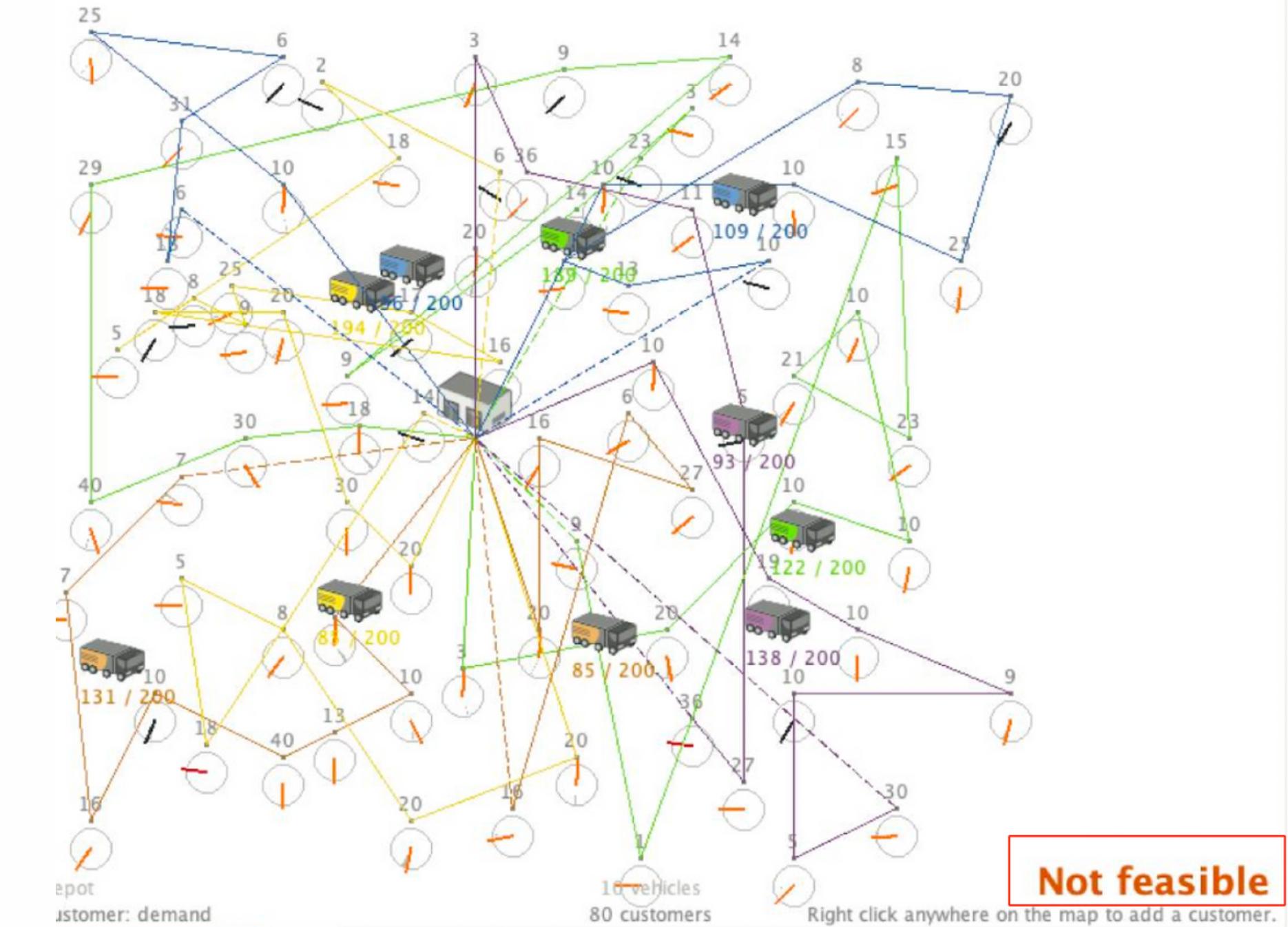
Solutions

- Baser la capacité sur un exemple d'OP
- Baser les coordonnées sur un exemple d'OP
- Uniquement des missions à 1 brancardier

Méthode métaheuristique

Simulation de 80 missions avec 10 brancardiers

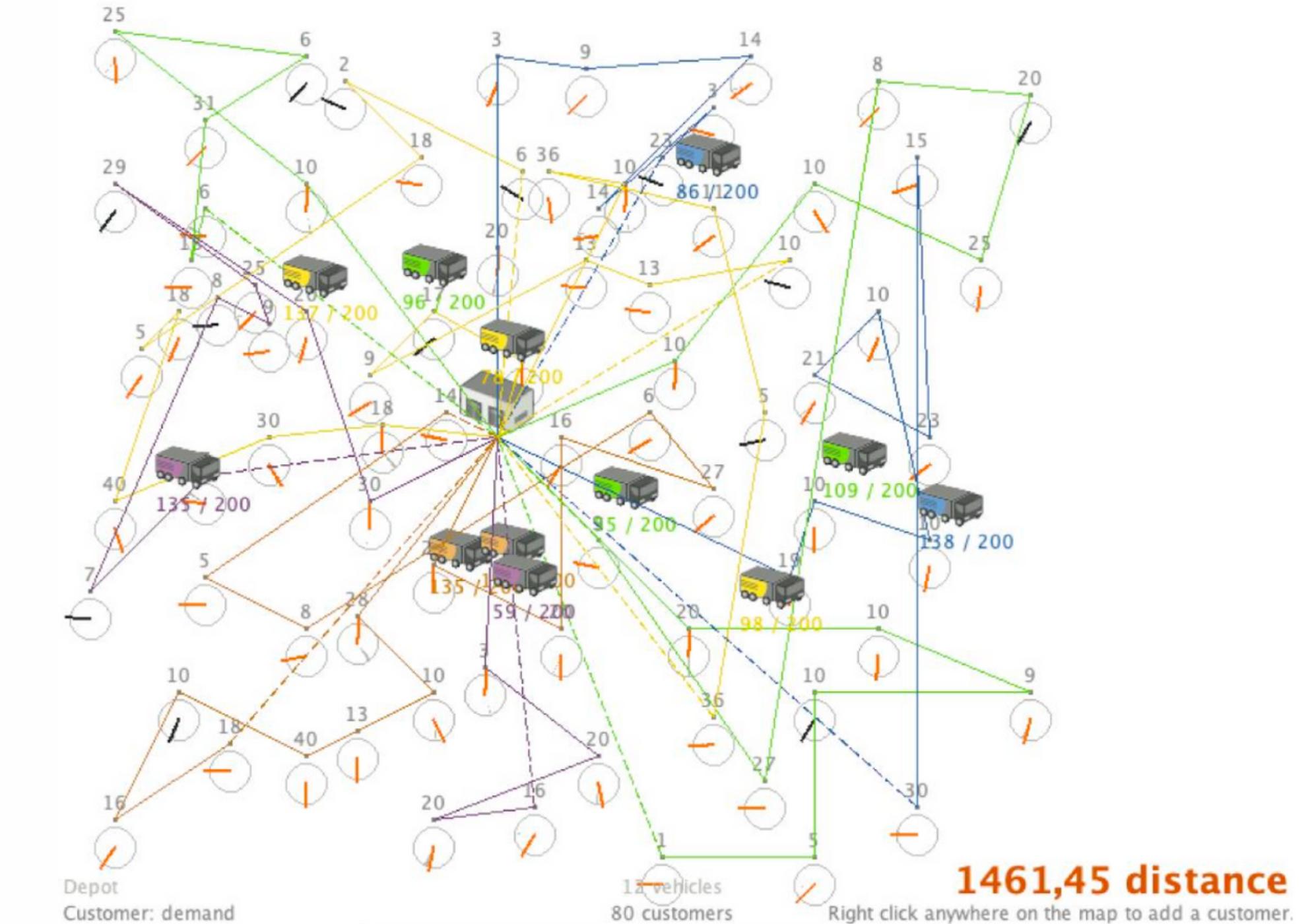
INDICATEURS	10 BRANCARDIERS
Retard	7
Moyenne charge	8
Ecart moyen	1
Score	1279,98



Méthode métaheuristique

Simulation de 80 missions avec 12 brancardiers

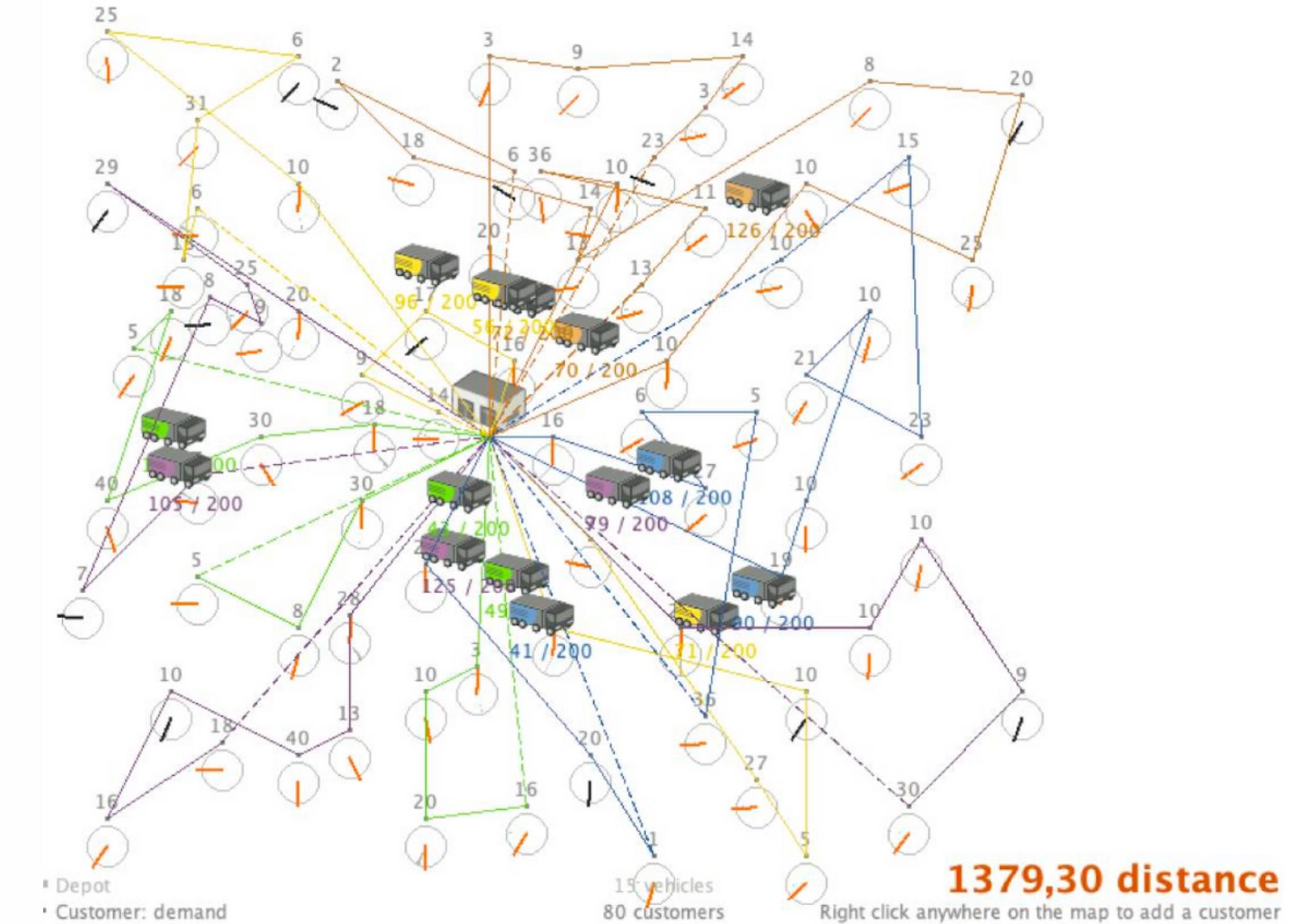
INDICATEURS	12 BRANCARDIERS
Retard	0
Moyenne charge	6.5
Ecart moyen	0.6
Score	1461,45



Méthode météuristicque

Simulation de 80 missions avec 15 brancardiers

INDICATEURS	15 BRANCARDIERS
Retard	0
Moyenne charge	5
Ecart moyen	1
Score	1379,30



Méthode métaheuristique

Comparaison des résultats



Indicateurs de retard

Méthode métaheuristique

Comparaison des résultats

NOMBRE DE BRANCARDIER	15	12	10
Retard cumulé	0	0	7
Charge moyenne	5	6,5	8
Charge maximale	7	8	10
Charge minimum	3	6	6
Ecart moyen	1	0,6	1
Score total	1379,30	1461,45	1279,98

Comparaison entre les résultats avec plusieurs indicateurs

Méthode heuristique

Résultats

Nombre de mission	1B	2B	Nombre de brancardier	Temps d'exécution (seconde)	Répartition (nb de missions)	Retard cumulé (seconde)
50	35	15	4	0.07	17,13,15,13	24000
50	35	15	5	0.12	17,11,15, 5, 9	16800
50	35	15	6	0.11	11,13,10, 7, 9,11	12600
80	57	23	4	0.14	30,24,21,23	51900
80	57	23	5	0.13	28,19,23,15,11	38400
80	57	23	6	0.12	22,20,13,18,13, 8	35700
80	57	23	7	0.13	21,11,20,13,11,10,13	25200
80	57	23	8	0.15	21,11,20,13,11,10,13,4	25200

Comparaison des résultats

Méthode heuristique

- Temps de calcul bas
- Mauvaise répartition de missions
- Gros retard cumulé
- 8 brancardiers max

Méthode mét-heuristique

- Temps de calcul bas-moyen
- Très bonne répartition de missions
- Aucun retard cumulé
- 12 brancardiers min

Méthode exacte

- Temps de calcul bas
- Bonne répartition de missions
- Aucun retard cumulé
- 9 brancardiers min

Axes d'amélioration



OptaPlanner

- Coder les contraintes restantes
- Coder des fonctionnalités supplémentaires
- Adapter les jeux de données avec coordonnées

CPLEX

Amélioration de l'exploitation du temps de travail maximal des brancardiers

NOMBRE DE BRANCARDIER	12	10	9
Charge moyenne	180	211	228
Charge maximale	238	308	329
Charge minimum	113	159	135

585

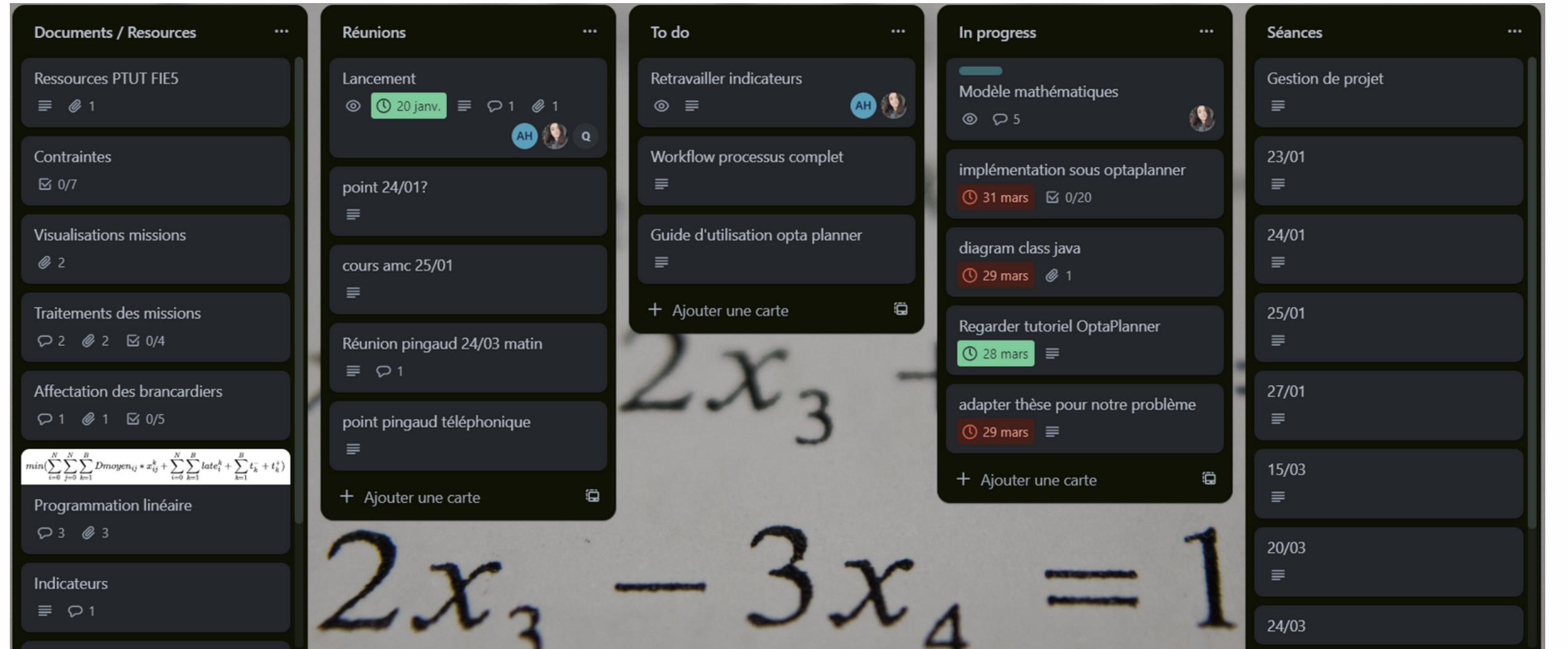
Charge de travail
maximal moyenne

Gestion de projet



Gestion du projet

Méthode agile



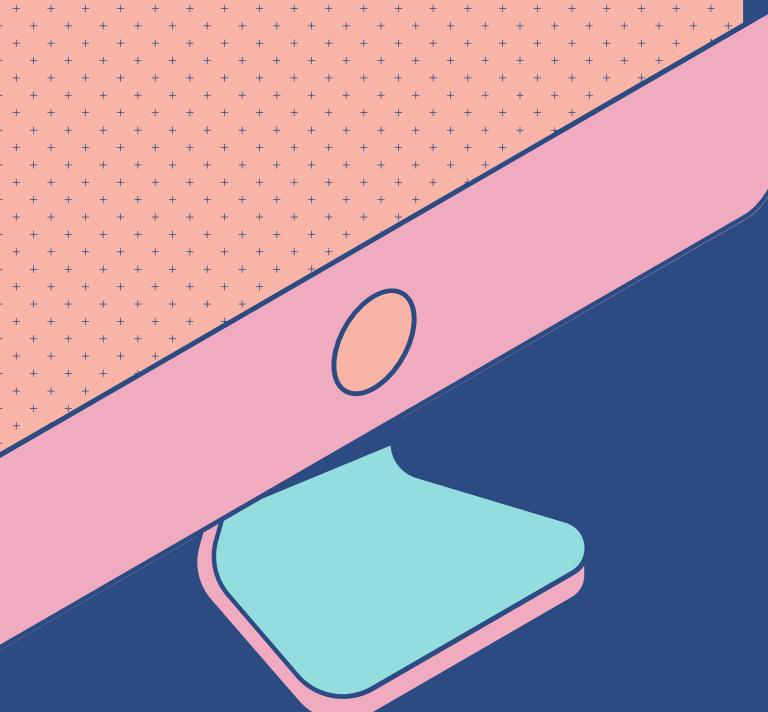
Utilisation de Trello

Répartition par compétences

Points réguliers

Conclusion

Démonstration



Merci pour votre attention

