

ÉCOLE POLYTECHNIQUE DE L'UNIVERSITÉ FRANÇOIS RABELAIS DE TOURS  
DEPARTEMENT INFORMATIQUE  
64, Avenue Jean Portalis  
37200 TOURS, FRANCE  
Tél. +33 (0)2 47 36 14 18  
[www.polytech.univ-tours.fr](http://www.polytech.univ-tours.fr)



**POLYTECH<sup>®</sup>**  
**TOURS**

Département Informatique



**UNIVERSITÉ**  
**FRANÇOIS - RABELAIS**  
**TOURS**

# Rapport de Projet Fin d'Etude 2011/2012

---

## *Problème dynamique de transport à la demande*

---

Etudiant :

Fabien Touchard

[fabien.touchard@etu.univ-tours.fr](mailto:fabien.touchard@etu.univ-tours.fr)

Encadrants :

Ahmed Atahran

[ahmed.atahram@etu.univ-tours.fr](mailto:ahmed.atahram@etu.univ-tours.fr)

Christophe Lente

[christophe.lente@univ-tours.fr](mailto:christophe.lente@univ-tours.fr)

Vincent T'Kindt

[vincent.tkindt@univ-tours.fr](mailto:vincent.tkindt@univ-tours.fr)

# Table de matières

|                                                                   |    |
|-------------------------------------------------------------------|----|
| Introduction .....                                                | 5  |
| Le transport à la demande .....                                   | 6  |
| Généralités.....                                                  | 6  |
| Définition .....                                                  | 7  |
| Définition générale .....                                         | 7  |
| Le problème statique .....                                        | 7  |
| Le problème dynamique .....                                       | 8  |
| Modélisation .....                                                | 10 |
| Les véhicules .....                                               | 10 |
| Les requêtes des clients .....                                    | 10 |
| Les tournées.....                                                 | 10 |
| Les contraintes .....                                             | 11 |
| Les fonctions objectif .....                                      | 11 |
| Les données .....                                                 | 11 |
| Le fichier des véhicules .....                                    | 11 |
| Le fichier des requêtes statiques .....                           | 12 |
| Le fichier des requêtes dynamiques.....                           | 12 |
| Le fichier des solutions initiales .....                          | 12 |
| L'heure de soumission .....                                       | 13 |
| Heuristique par insertion .....                                   | 14 |
| Description générale.....                                         | 14 |
| Ordonnancement des requêtes .....                                 | 14 |
| 1 <sup>er</sup> composant : La largeur des fenêtres de temps..... | 14 |
| 2 <sup>ème</sup> composant : Le temps maximum de parcours.....    | 15 |
| 3 <sup>ème</sup> composant : Le nombre de clients.....            | 15 |
| Coefficient global .....                                          | 15 |
| Ordonnancement des tournées .....                                 | 15 |
| Calcul de $m(ri)k$ .....                                          | 16 |
| Calcul de $d(ri)k$ .....                                          | 17 |
| Faisabilité d'une insertion.....                                  | 18 |
| Vérification de la capacité.....                                  | 18 |
| Vérification des fenêtres de temps .....                          | 19 |
| Positions d'insertion .....                                       | 21 |
| Initialisation avec l'heure de soumission .....                   | 21 |
| Dépassement de la capacité entre i (inclus) et j.....             | 21 |
| Front de Pareto.....                                              | 22 |

|                                                          |    |
|----------------------------------------------------------|----|
| Distance de Chebyshev .....                              | 22 |
| Optimal Timing .....                                     | 23 |
| Algorithme .....                                         | 24 |
| Méthode arborescente .....                               | 25 |
| Description générale .....                               | 25 |
| L'état initial .....                                     | 25 |
| Le schéma de branchement .....                           | 26 |
| La règle d'affectation .....                             | 27 |
| La règle d'ordonnancement .....                          | 27 |
| Les filtres .....                                        | 28 |
| L'évaluation .....                                       | 28 |
| La sélection .....                                       | 29 |
| La récupération .....                                    | 30 |
| Opérateur de voisinage .....                             | 30 |
| Comparaison voisins .....                                | 31 |
| Tests .....                                              | 32 |
| Méthodologie .....                                       | 32 |
| Statistiques sur l'heuristique par insertion .....       | 33 |
| Evolution du temps de calcul .....                       | 33 |
| Position de la meilleure tournée .....                   | 33 |
| Statistiques sur la méthode arborescente .....           | 34 |
| Taux de réussite et d'insertion .....                    | 34 |
| Taille du front de Pareto .....                          | 35 |
| Temps d'exécution .....                                  | 36 |
| Heuristique Insertion vs RBS .....                       | 36 |
| Heuristique vs NSGAI vs RBS .....                        | 38 |
| Conclusion .....                                         | 41 |
| Bibliographie .....                                      | 42 |
| Annexes .....                                            | 43 |
| Diagramme de classe de l'heuristique par insertion ..... | 44 |
| Diagramme de classe de la RBS .....                      | 45 |
| Planning prévisionnel .....                              | 46 |
| Planning modifié .....                                   | 47 |
| Fichier de données « véhicules » .....                   | 48 |
| Fichier de données « requêtes statiques » .....          | 49 |
| Fichier de données « requêtes dynamiques » .....         | 50 |
| Fichier de données « solutions » .....                   | 51 |
| Résultats heuristique par insertion pour T=0 .....       | 52 |

|                                                     |    |
|-----------------------------------------------------|----|
| Résultats heuristique par insertion pour T=200..... | 53 |
| Résultats heuristique par insertion pour T=400..... | 54 |
| Résultats heuristique par insertion pour T=600..... | 55 |
| Résultats heuristique par insertion pour T=800..... | 56 |
| Résultats RBS pour T=0 .....                        | 57 |
| Résultats RBS pour T=200 .....                      | 58 |
| Résultats RBS pour T=400 .....                      | 59 |
| Résultats RBS pour T=600 .....                      | 60 |
| Résultats RBS pour T=800 .....                      | 61 |
| Comparaison heuristique – RBS pour T=0.....         | 62 |
| Comparaison heuristique – RBS pour T=200.....       | 63 |
| Comparaison heuristique – RBS pour T=400.....       | 64 |
| Comparaison heuristique – RBS pour T=600.....       | 65 |
| Comparaison heuristique – RBS pour T=800.....       | 66 |
| Comparaison heuristique – NGAll.....                | 67 |
| Comparaison RBS – NGAll.....                        | 68 |
| Rapport de Projet Fin d’Etude 2011/2012 .....       | 69 |

# Introduction

---

Ce document est un rapport d'activité sur le projet de fin d'étude « *problème dynamique de transport à la demande* » effectué au sein de l'école polytechnique de Tours dans le cadre de ma 5<sup>ème</sup> année d'études.

L'encadrant principal de ce projet est Ahmed Atahran, doctorant au sein de l'équipe OC (Ordonnancement et Conduite des systèmes de production). Il effectue sa thèse sur le problème du transport à la demande. Le projet est également encadré par Vincent T'kindt et Christophe Lente, enseignants-chercheurs à Polytech Tours.

Ce projet s'inscrit dans une suite de travaux (notamment d'autres projets de fin d'étude), menés l'année dernière et cette année sur le thème du transport à la demande. Même si ce projet est relativement indépendant, nous verrons dans la suite de ce document que nous avons été amenés à utiliser les résultats des projets précédents pour nous fournir des données ou pour vérifier la qualité de nos propres résultats.

L'objectif du PFE est de spécifier, développer puis évaluer une méthode heuristique simple et une méthode arborescente de type RBS (Recovering Beam Search) pour le problème du transport à la demande dynamique.

Dans un premier temps, nous allons aborder d'une manière générale le problème du transport à la demande, en évoquant de manière rapide la famille des problèmes de type « statique », puis plus en détails celle des problèmes de type « dynamique ». Après la modélisation du problème, nous aborderons en détails les travaux entrepris durant ce PFE : une heuristique par insertion et une méthode arborescente. Dans une troisième partie, nous présenterons la campagne de test effectuée pour évaluer la qualité de ces deux méthodes et en commenterons les résultats.

# Le transport à la demande

## Généralités

Le transport à la demande est un système de transport collectif qui diffère des systèmes de transports habituels de part une certaine souplesse. En effet, dans un réseau de bus classique, comme on en trouve dans la majorité des villes, nous pouvons identifier les caractéristiques suivantes :

- Arrêts de bus (départ/arrivées) fixes
- Horaires fixes
- Ordre de passage aux arrêts (tournées) fixe

Ce système rigide est mal adapté aux zones peu densément peuplées. Le transport à la demande quand à lui se veut plus souple, le but étant de s'adapter à la demande du client et non l'inverse. Tout d'abord implanté dans les zones rurales, il se développe maintenant dans les zones péri-urbaines pour répondre à un besoin de déplacement des habitants de ces zones.

Dans ces zones, les réseaux de transport classiques ne sont pas viables économiquement pour l'exploitant, tout simplement parce que la demande n'est pas assez forte pour remplir un bus à horaire régulier et à cadence régulière. La solution mise en place généralement est une diminution de la fréquence de passage des bus (par exemple deux fois par jour). Mais cette solution est loin d'être idéale, car elle est trop contraignante pour l'utilisateur qui se retrouve avec des horaires de passage non adaptés à ses besoins.

Les réseaux fonctionnant en transport à la demande (TAD) tentent donc de proposer une alternative en trouvant un compromis entre le coût d'exploitation et la qualité du service proposée. Pour cela, un TAD a les caractéristiques suivantes :

- Une zone de desserte définie (ex : 25 kms autour de la ville)
- Des plages horaires de fonctionnement (ex : 7j/7, de 8h à 20h)
- Réservations à l'avance (TAD statique) ou le jour même (TAD dynamique)
- Ordre de passage aux arrêts NON fixe
- Horaire de passage aux arrêts fixe ou non (si l'horaire est libre, l'utilisateur spécifie sa préférence)
- Arrêts de bus (départ/arrivées) fixes ou spécifiés par l'utilisateur (porte à porte)

Pour être considéré comme un transport à la demande, un réseau doit laisser libre au moins l'un des deux derniers points.

|                                       | Ligne régulière | Transport à la demande |                                             | Taxi |
|---------------------------------------|-----------------|------------------------|---------------------------------------------|------|
|                                       |                 | Ligne virtuelle        | TAD <i>stricto sensu</i>                    |      |
| Zone de desserte                      | Oui             | Oui                    | Oui                                         | Non  |
| Plage(s) horaire(s) de fonctionnement | Oui             | Oui                    | Oui, en général                             | Non  |
| Tracé                                 | Oui             | Théorique              | Non                                         | Non  |
| Réservation préalable                 | Non             | Oui                    | Oui                                         | Oui  |
| Horaires                              | Oui             | Oui                    | Au moins un des 3 critères n'est pas imposé | Non  |
| Point de montée imposé                | Oui             | Oui                    |                                             | Non  |
| Point de descente imposé              | Oui             | Oui                    |                                             | Non  |

Figure 1 : Comparaison des modes de transport

Il existe deux variantes de transport à la demande : statique ou dynamique. Dans la version statique, les clients doivent réserver leur trajet à l'avance (par exemple la veille). Les tournées sont alors calculées et aucune modification ne pourra y être apportée. Cette méthode procure l'avantage de donner du temps à l'exploitant pour organiser au mieux sa journée, connaissant en plus toutes les réservations. Dans la version dynamique, les clients peuvent réserver leur voyage jusqu'au dernier moment, on peut supposer une limite de 30 minutes avant l'heure de passage demandée. Ce système offre une plus grande souplesse et permet d'attirer une clientèle ayant des besoins plus imprévisibles, comme des étudiants par exemple. A l'inverse, cela oblige à modifier les tournées, dont certaines sont déjà en cours de service, et donc ajoute de fortes contraintes à l'exploitant qui ne pourra pas (dans la plupart des cas) organiser sa tournée de manière optimale.

## Définition

---

### Définition générale

---

Dans ce PFE, nous allons étudier de nouvelles méthodes pour un problème précis de transport à la demande dynamique (DTAD). Ce problème fait partie de la famille des problèmes de type DARP (dial a ride problem) dynamique. Les problèmes de type DARP concernent tous les problèmes où l'on essaye de répartir dans un ensemble de véhicules des objets (ou personnes) en les prenant en charge à un point A pour les emmener à un point B tout en essayant de satisfaire leurs demandes.

Le DARP est une extension des problèmes de type PDP (pickup and delivery problems), car ils demandent de satisfaire le client, en plus d'assurer la livraison d'un objet entre deux points. Il a été montré que ce problème est NP-difficile. Nous allons donc essayer de mettre en place des méthodes non-exactes pour résoudre ce problème dans des temps plus raisonnables.

Si on analyse plus en détails le problème du transport à la demande, on se rend compte qu'il est en fait composé de trois sous problèmes eux-mêmes complexes à résoudre :

- Un problème d'affectation : chaque requête doit être assignée à un seul véhicule
- Un problème d'ordonnancement des requêtes affectées à un véhicule
- Un problème d'optimisation multicritères

### Le problème statique

---

Dans notre version du problème, nous avons à traiter un ensemble de requêtes. Chaque requête est une demande pour transporter depuis un arrêt A vers un arrêt B un ensemble X de personnes. L'heure de passage à l'arrêt A et à l'arrêt B est fixée par le client. Pour cela, il fournit pour chacun des deux arrêts une fenêtre de temps (heure au plus tôt, heure au plus tard).

Afin de prendre en charge ces clients, nous disposons d'un ensemble de véhicules hétérogènes. C'est-à-dire que chaque véhicule a ces propres caractéristiques : capacité, coût kilométrique et émission de CO<sub>2</sub>/km. Il existe dans la littérature des problèmes où les véhicules possèdent plusieurs capacités (par exemple le nombre de places handicapés). Pour simplifier les choses, nous ne prenons en compte qu'une seule capacité. Tous ces véhicules sont présents dans un dépôt unique en début de journée, et y retournent après avoir achevé leur tournée en fin de journée. Enfin une requête doit être prise en charge entièrement par un véhicule (pas de changement de véhicule pour les passagers). Contrairement à d'autres études existantes, nous ne prenons pas en compte la longueur

du trajet ou le temps maximum pour l'accomplissement d'une tournée. La gestion des chauffeurs ne fait pas partie de notre problème.

L'objectif est alors de planifier un ensemble de tournées prenant en charge l'ensemble des requêtes connues, en essayant de respecter au maximum les contraintes d'horaires fournies par les clients et en minimisant à la fois le coût pour l'exploitant et la quantité de rejet de CO2. Ces tournées doivent bien sûr respecter les contraintes de capacités des bus. On voit bien alors qu'il s'agit d'un problème multicritères, en l'occurrence il possède trois fonctions objectifs:

- La satisfaction des clients
- Le coût pour le transporteur
- L'impact sur l'environnement

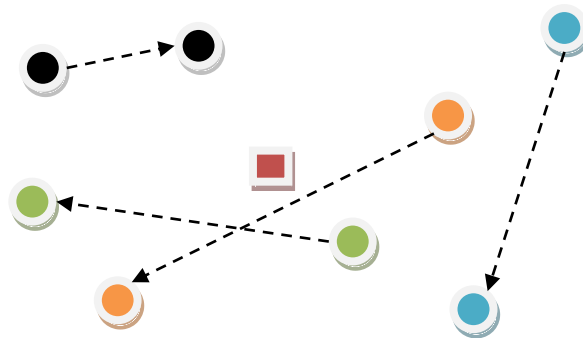


Figure 2 : Etat initial

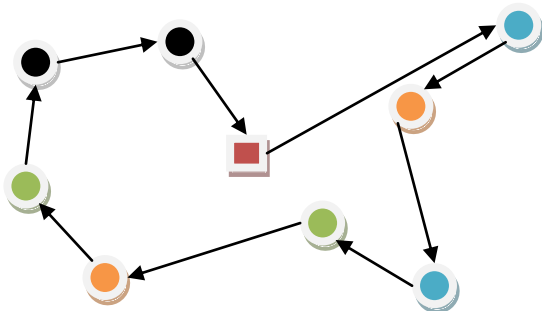


Figure 3: solution avec un véhicule

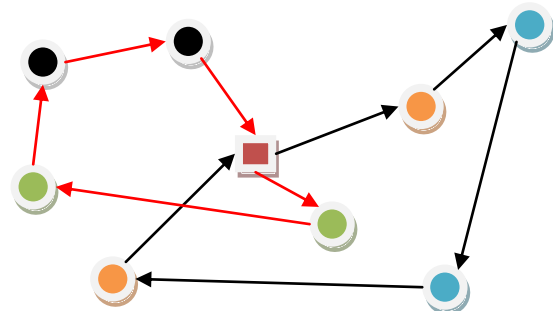


Figure 4 : solution avec deux véhicules

## Le problème dynamique

Notre sujet porte sur la partie dynamique du problème. C'est-à-dire que l'on permet à des clients d'appeler dans la journée pour demander une prise en charge le jour même. Les requêtes entrantes sont alors (si possible) ajoutées aux tournées déjà programmées avec les réservations précédentes. L'état initial de notre programme est en fait l'état final du programme traitant la version statique du problème. Etant possible qu'à l'heure de l'appel, les véhicules soient déjà en service, il existe alors deux possibilités :

- modifier la structure des tournées
- insérer la nouvelle requête dans les tournées existantes sans modifier l'ordonnancement

De plus, même si on modifie la structure, on ne peut la modifier que sur la partie qui n'est pas encore servie. Si on prend une solution de base comme celle-ci :

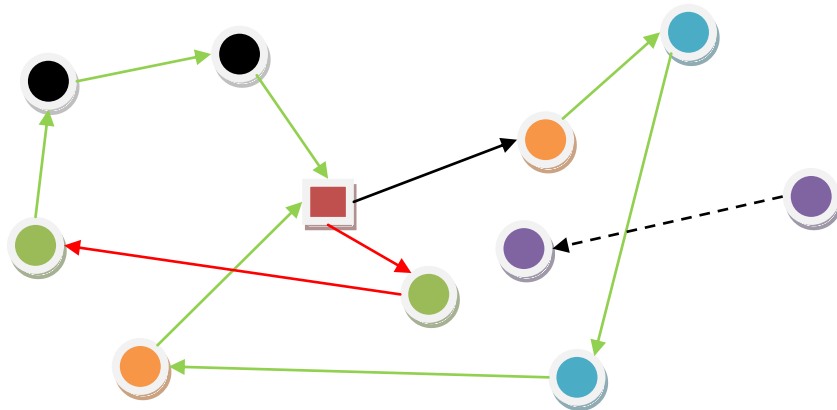


Figure 5 : Etat des tournées au temps T (en vert, les tronçons modifiables)

On veut insérer la nouvelle requête en violet. Il existe déjà deux routes qui sont en cours de service, et vu l'heure courante, nous ne pouvons modifier que la partie en vert de ces routes. Nous pouvons alors obtenir deux genres de solutions :

- Sans modifier les structures, on insère simplement les deux arrêts de la requête violette dans la tournée noire.

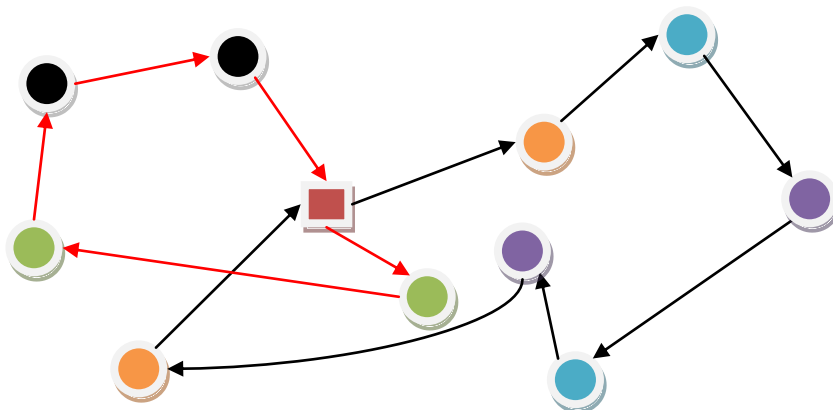


Figure 6 : solution sans modification des tournées existantes

- En modifiant la structure, on affecte maintenant la requête bleue à la tournée rouge et la nouvelle requête à la tournée noire.

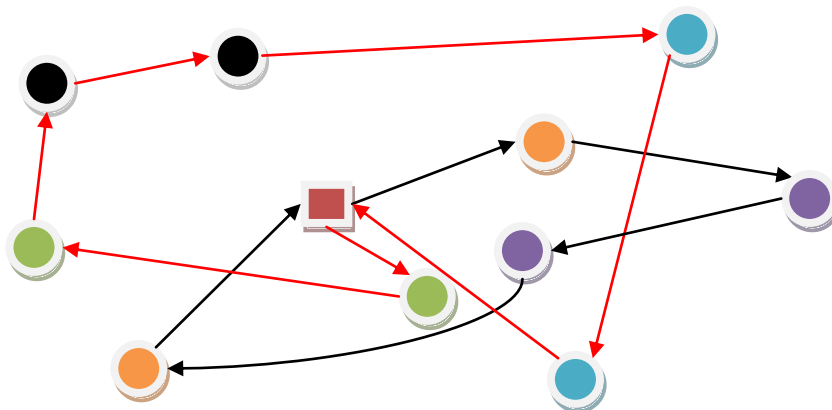


Figure 7 : solution en modifiant les tournées existantes

Attention, on ne peut changer l'affectation d'une requête que si aucun des deux sommets n'a été servi. Dans le cas précédent, il serait par exemple impossible d'affecter la requête orange à la tournée rouge, alors que le véhicule a déjà desservi l'arrêt de montée orange. Cela reviendrait à changer les clients de véhicule au cours de leur trajet. On parle alors de requête en cours de traitement. De plus, on ne peut pas modifier l'affectation ni l'ordonnancement de la requête verte, car elle a déjà été entièrement servie. On parle alors de requête terminée.

## Modélisation

---

Les choix de modélisation du problème sont hérités des PFE de l'année précédente. Nous avons donc conservé les mêmes notations à l'exception de la notation des lieux liées aux requêtes. Dans l'ancienne notation, une requête  $r_i$  desservait les lieux  $V_i$  et  $V_{i+N}$ ,  $N$  étant le nombre de requêtes totales. Dans notre problème dynamique, nous ne pouvons pas savoir combien de requêtes seront traitées dans la journée. Nous avons donc abandonné cette notation. Dorénavant, une requête  $r_i$  sert les arrêts  $V_{2i}$  et  $V_{2i+1}$ .

### Les véhicules

---

$V_0$  : Le dépôt commun de l'ensemble des véhicules.

$M$  : Le nombre total de véhicules disponibles.

$Q_k$  : La capacité du véhicule  $k$ , avec  $k \in \{1, \dots, M\}$ .

$B_k$  : Le coût kilométrique de l'utilisation du véhicule  $k$ , avec  $k \in \{1, \dots, M\}$ .

$H_k$  : Emission de  $CO_2$  du véhicule  $k$  en gramme par kilomètre, avec  $k \in \{1, \dots, M\}$ .

### Les requêtes des clients

---

$r_i$  : La requête  $i$

$V_{2i}$  : Le lieu de départ (origine)

$V_{2i+1}$  : Le lieu d'arrivée (la destination).

$[e_i, l_i]$  : La fenêtre de temps au sommet  $V_i$ , avec  $e_i$  l'heure au plus tôt, et  $l_i$  l'heure au plus tard à laquelle les passagers de la requête  $r_i$  souhaitent être servis.

$\alpha_i$  : Pénalité si le véhicule arrive en avance au sommet  $V_i$

$\beta_i$  : Pénalité si le véhicule arrive en retard au sommet  $V_i$

$q_i$  : Le nombre de personnes de la même requête  $r_i$ , on suppose que toutes les personnes  $q_i$  doivent être embarquées à un seul point de départ et débarquées à un seul point d'arrivée.

$s_i$  : Le temps de service associé au sommet  $V_i$ .

$D_{ij}$  : La distance qui sépare les deux sommets  $V_i$  et  $V_j$ .

### Les tournées

---

$X_{ij}^k$  : Vrai si le véhicule  $K$  visite le nœud  $V_j$  juste après le nœud  $V_i$

$A_i^k$  : Heure d'arrivée du véhicule  $K$  au nœud  $V_i$

$W_i^k$ : Temps d'attente du véhicule K avant de servir le nœud  $V_i$

$E_i^k$ : Nombre de places disponibles dans le véhicule K après avoir desservi le nœud  $V_i$

## Les contraintes

---

La seule contrainte forte est le respect des capacités des véhicules. Ce que l'on peut traduire par :

$$E_j^k + q_j * X_{ij}^k \leq Q^k, \forall k \in \{1, \dots, M\}, \forall i, j \in \{0, \dots, 2n+1\}$$

D'autre part, la solution fournie est sensée respecter les fenêtres de temps spécifiées par l'utilisateur, mais il ne s'agit pas (du moins dans un cadre général) d'une contrainte. Le non-respect de ces fenêtres de temps est toléré, mais entraîne des pénalités dont l'importance est spécifiée par l'utilisateur grâce aux variables  $\alpha, \beta$ .

## Les fonctions objectif

---

Nous avons décidé de prendre en compte trois objectifs dans notre problème :

- Minimiser la quantité de CO2 émise par les véhicules pour répondre à une demande écologique :

$$\text{Minimiser } CO2 = \sum_{k=1}^M h_k \sum_{i=0}^{2n} \sum_{\substack{j=0 \\ j \neq i}}^{2n} d_{ij} x_{ij}^k$$

- Minimiser le coût total de transport, pour l'exploitant :

$$TC = \sum_{k=1}^M b_k \sum_{i=0}^{2n} \sum_{\substack{j=0 \\ j \neq i}}^{2n} d_{ij} x_{ij}^k$$

- Minimiser l'insatisfaction des clients (maximiser la qualité de service) :

$$\text{Minimiser } QoS = \sum_{i=1}^P \max(0, \alpha_i (e_i - A_i^k)) + \max(0, \beta_i (A_i^k - l_i)) + \sum_{i=2}^P \omega_i (A_i^k - (A_{i-1}^k + t_{i-1,i} + s_{i-1}))$$

## Les données

---

Les données du problème sont fournies par quatre fichiers qui contiennent respectivement : les données sur les véhicules, sur les requêtes «statiques», sur les requêtes «dynamiques», et un ensemble de solutions initiales.

### Le fichier des véhicules

---

Ce fichier contient les informations relatives à notre flotte de véhicules hétérogène. Chaque ligne du fichier représente un véhicule qui possède les caractéristiques suivantes :

- Un identifiant
- Une capacité
- Un coût kilométrique
- Une valeur d'émission de CO2

### Le fichier des requêtes statiques

---

Il s'agit des requêtes qui ont été enregistrées plus tôt, et qui sont déjà validées. Ce sont ces requêtes que nous allons retrouver dans les solutions initiales calculées de manière statique par un algorithme que nous ne possédons pas tel que NSGAI, un algorithme génétique. Chaque ligne représente un arrêt (montée ou descente) qui présente les caractéristiques suivantes :

- Un identifiant
- Une position X et Y
- Un temps de service ( $s_i$ )
- Un nombre de personne qui montent ou descendent ( $q_i$ )
- Un temps d'arrivée au plus tôt et au plus tard ( $[e_i, l_i]$ )
- Des pénalités pour dépassement de la fenêtre ( $\alpha_i$  et  $\beta_i$ )

**N.B :** Il faut noter que les arrêts sont numérotés selon le système utilisé par le problème statique. C'est-à-dire que le point de descente correspondant au point de montée  $i$  aura pour identifiant  $N+i$ ,  $N$  étant le nombre de requête statiques. Il convient donc de modifier cette numérotation après lecture du fichier pour l'adapter à notre problème.

### Le fichier des requêtes dynamiques

---

Il s'agit des requêtes entrantes, que nous allons essayer d'ajouter dans nos tournées déjà existantes afin de servir de nouveaux clients. Ce fichier se présente exactement comme celui des requêtes statiques et les arrêts possèdent les mêmes attributs.

**N.B :** Attention, cette fois les identifiants sont générés selon la norme définie pour le problème dynamique, on a donc bien le point de descente correspondant au point de monté  $i$  avec l'identifiant  $i+1$ . Cependant, la numérotation des arrêts dans ce fichier commence à 1. Pour éviter les doublons avec les arrêts statiques, qui commencent aussi à 1, il convient de refaire la numérotation en incrémentant chaque arrêt dynamique d'une valeur  $2N$ ,  $N$  étant le nombre de requêtes statiques.

### Le fichier des solutions initiales

---

Il s'agit d'un fichier qui contient un ensemble de solutions, calculées par l'algorithme génétique NSGAI, qui sont toutes de qualité équivalente au sens multicritères. Une solution est composée d'un ensemble de tournées. Ces tournées sont décrites par :

- Un identifiant, qui correspond à l'identifiant du véhicule auquel est affecté cette tournée.
- Une valeur pour chacune des trois fonctions objectif
- Un ensemble d'arrêts « statiques » ordonnancés qui constituent la tournée. Cet ensemble commence et se termine toujours par l'arrêt d'identifiant 0, qui représente le dépôt.

A chaque arrêt constituant une tournée est associé une heure d'arrivée, et un temps d'attente.

**N.B :** Il est possible que les heures d'arrivées spécifiées dans ce fichier se trouvent en dehors des fenêtres de temps spécifiées dans le fichier des requêtes statiques. Dans ce cas, cela signifie que le client à accepter d'être servi en dehors de ses préférences de départ. Dans notre problème statique, nous considérons que si le client à accepter cette modification, alors la nouvelle fenêtre de temps devient la fenêtre de référence. On effectue donc une opération de mise à jour des fenêtres de temps tel que :

$$\begin{aligned} \text{Si } A_i^k < e_i \text{ Alors } e_i &= A_i^k \\ \text{Si } A_i^k > l_i \text{ Alors } l_i &= A_i^k \end{aligned}$$

Nous verrons, notamment dans la phase de test, que cela a une influence sur le calcul de la qualité de service, qui prend en compte les dépassements de fenêtre de temps.

### L'heure de soumission

---

Ce dernier paramètre n'est pas présent dans un fichier, il doit être spécifié manuellement par l'utilisateur (ou calculé automatiquement en fonction de l'heure réelle s'il s'agit d'un cas concret). L'heure de soumission correspond à l'heure courante, au moment où l'on tente de modifier les solutions initiales pour y ajouter les requêtes dynamiques. Comme nous l'avons vu dans la présentation du problème dynamique, cette heure va influencer sur les modifications que l'on peut apporter aux tournées, car nous ne pourrions pas changer l'affectation de requêtes en cours de traitement, ni l'ordonnancement de requêtes déjà traitées.

# Heuristique par insertion

---

La première partie du PFE a été consacrée à la spécification, puis le développement d'une heuristique simple. Le but était de fournir une solution rapide, et bien sûr de la meilleure qualité possible.

## Description générale

---

Le but est d'insérer rapidement (moins d'une seconde), un ensemble de requêtes dynamiques à une solution statique initiale. Pour simplifier la génération des solutions et diminuer ainsi le temps de calcul, nous allons nous interdire de toucher à l'affectation et à l'ordonnement des requêtes « statiques ». Nous allons donc simplement tester toutes les insertions possibles pour les deux arrêts d'une requête dynamique et sélectionner les meilleures solutions selon nos trois fonctions objectif.

## Ordonnement des requêtes

---

Nous avons en entrée de notre problème un ensemble de requêtes dynamiques à insérer. La première question que l'on se pose concerne l'ordre d'insertion de ces requêtes. En effet, il paraît logique que plus une tournée est chargée (plus elle sert de requêtes), plus il est difficile d'en insérer une nouvelle. Partant de ce constat, il est alors clair que l'ordre d'insertion des nouvelles requêtes peut avoir une importance cruciale dans la qualité de la solution finale.

En outre, il faut remarquer que toutes les requêtes entrantes n'ont pas les mêmes caractéristiques. Certaines concernent beaucoup de passagers, d'autres ont des fenêtres de temps très petites ou des arrêts de montée et descente très éloignés. En utilisant cet ensemble de critères, nous avons décidé de créer un indicateur, appelé « coefficient de difficulté », pour pouvoir comparer les requêtes et les trier.

Ce coefficient représente les caractéristiques intrinsèques de la requête, et est complètement indépendant des tournées existantes.

Nous partons du principe que le but est de réussir à insérer toutes les requêtes. Les requêtes seront donc insérées de la plus difficile à la plus facile, c'est-à-dire du coefficient le plus grand au plus petit.

### 1<sup>er</sup> composant : La largeur des fenêtres de temps.

---

On peut dire que plus les fenêtres de temps données par le client sont serrées, plus la requête sera difficile à satisfaire. En effet, si on veut répondre favorablement à sa demande en respectant ces fenêtres, une fenêtre serrée réduit les possibilités. On peut donc poser l'équation suivante :

$$C_{win} = \frac{1}{(l_i - e_i)} + \frac{1}{(l_{i+1} - e_{i+1})}$$

Plus les fenêtres de temps des deux arrêts de la requête sont petites, plus ce composant sera grand.

## 2<sup>ème</sup> composant : Le temps maximum de parcours

---

Il s'agit de la différence entre le temps de parcours idéal pour aller du sommet  $V_i$  au sommet  $V_{i+1}$  (donnée par  $D_{i(i+1)}$ ) et le temps maximum alloué par le client :

$$C_{ride} = \frac{1}{(l_{i+1} - e_i) - D_{i(i+1)}}$$

$l_{i+1}$  étant la borne supérieure pour le sommet  $V_{i+1}$  (arrivée),  $e_i$  la borne inférieure pour le sommet  $V_i$  (départ), et  $D_{i(i+1)}$  le temps entre les deux sommets d'après la matrice des distances. Plus la marge fournie par le client est petite, plus ce composant est grand.

## 3<sup>ème</sup> composant : Le nombre de clients

---

Plus il y a de clients à prendre en charge pour une requête, plus elle est difficile à satisfaire car elle risque de violer les contraintes de capacité des véhicules.

$$C_{load} = q_i$$

$q_i$  représente le nombre de client à servir au sommet  $V_i$  (départ).

## Coefficient global

---

On effectue une somme pondérée des trois composants précédemment définis :

$$C_{req} = \alpha \cdot C_{fen} + \beta \cdot C_{ride} + \gamma \cdot C_{load}$$

Si  $C_{req}$  est grand, cela signifie que la requête est difficile à insérer.

## Ordonnancement des tournées

---

Une fois la requête à insérer sélectionnée, il faut tenter de l'insérer dans l'une des tournées de la solution. Afin de diminuer encore les temps de calcul, il peut être utile de tenter de deviner dans quelle tournée la requête va le mieux s'insérer. C'est-à-dire, dans quelle tournée faut-il insérer la requête pour que la solution obtenue minimise les objectifs (coût, co2, qualité de service). Pour cela, nous allons introduire un nouveau coefficient que l'on nommera « degré d'attraction ».

Ce coefficient va nous permettre de ranger les tournées selon la probabilité que l'insertion de la requête dans cette tournée nous offre la meilleure solution possible. Si ce coefficient fonctionne correctement (voir la partie test), nous pourrions alors nous contenter de tester les premières tournées dans la liste pour insérer notre requête et non l'ensemble des tournées disponibles. Cela nous permettra alors de diminuer le temps de calcul.

Le nouveau coefficient se nomme « degré d'attraction », car la formule que nous avons retenue pour le calculer est issue de la théorie de la gravitation, qui nous apprend que l'attractivité d'un objet pour un autre est dépendante de son attractivité propre et de la distance qui sépare les deux objets.

$$C_{route}^{(r_i)k} = \frac{(m^{(r_i)k})^b}{(d^{(r_i)k})^c}$$

Ce modèle est connu dans les problèmes de géomarketing sous le nom de « Modèle de Huff ». Il est très utilisé dans les problèmes de sectorisation, pour connaître la probabilité qu'un client vienne dans un magasin, connaissant l'attractivité intrinsèque du magasin et la distance entre le client et le magasin (<http://www.geoconcept.com/Modele-de-Huff.html>).

Nous avons juste simplifié la formule en éliminant le dénominateur, qui n'avait pour but que de produire une probabilité plutôt qu'une valeur non bornée, ce qui n'est pas important pour nous.

Si on reprend la formule,  $m^{(r_i)k}$  est donc le paramètre d'attractivité de la tournée. Dans notre cas, nous allons définir l'attractivité comme le potentiel d'une tournée à satisfaire (prendre en charge) une ou plusieurs requêtes supplémentaires. Plus ce paramètre est élevé, plus il est facile d'insérer une nouvelle requête dans cette tournée.

Le deuxième paramètre  $d^{(r_i)k}$  représente la distance entre une tournée K et une requête  $r_i$  entrante. Plus cette distance est faible, plus il est facile d'insérer la requête.

Au final, on voit bien que plus le potentiel d'insertion  $m^{(r_i)k}$  est grand et plus la distance  $d^{(r_i)k}$  est petite, plus le coefficient  $C_{route}^{(r_i)k}$  sera grand.

### Calcul de $m^{(r_i)k}$

---

Le paramètre  $m^{(r_i)k}$  va représenter le potentiel d'acceptation de nouvelles requêtes pour une tournée. Il est composé de deux critères : le temps d'attente et la capacité résiduelle des véhicules.

#### 1<sup>er</sup> composant : le temps d'attente

---

Chaque arrêt associé à une tournée possède un attribut «temps d'attente», noté  $W_i^k$ . Cet attribut est utile quand la planification de la tournée contraint le véhicule à arriver en avance à un arrêt. Dans ce cas, le bus attend le temps nécessaire avant de servir cet arrêt. Ces temps d'attente sont donc des temps morts au sein de la tournée, temps mort durant lequel le véhicule reste immobilisé. Dans l'optique d'insérer une nouvelle requête, on peut imaginer que ces temps d'attente peuvent être utilisés pour servir un nouvel arrêt. Plus une tournée possède de temps d'attente, plus elle a donc de potentiel pour accueillir une nouvelle requête.

De plus, nous avons décidé de favoriser les temps d'attente qui sont plus susceptibles de servir à insérer la requête. C'est-à-dire, ceux qui sont situés dans les fenêtres de temps spécifiées par le nouvel utilisateur pour le servir. On obtient donc la formule suivante :

$$C_{wait}^{(r_i)k} = \sum_{i=0}^n \alpha_i W_i^k, n \text{ étant le nombre d'arrêts dans la tournée}$$

$$\text{avec } \begin{cases} \alpha = 1, \text{ si } A_i^k \in [e_j, l_{j+1}] \\ \alpha = 1/(e_j - A_i^k), \text{ si } A_i^k < e_j \\ \alpha = 1/(A_i^k - l_{j+1}), \text{ si } A_i^k > l_{j+1} \end{cases}$$

j étant l'identifiant de l'arrêt de montée

Plus  $C_{wait}^{(r_i)k}$  est grand, plus le potentiel de la tournée est grand.

#### 2<sup>ème</sup> composant : la capacité résiduelle

---

La capacité résiduelle est le nombre de places disponibles dans le véhicule après la montée (ou la descente) des clients à chaque arrêt de la tournée. Plus il reste de places dans le véhicule, plus le potentiel de la tournée doit être grand. Comme pour les temps d'attente, il faut favoriser les places libres sur la partie de la tournée qui coïncide avec les horaires demandés par le nouvel utilisateur. Dans cette optique, nous ne considérons ici que cette partie de la tournée dans notre calcul

$$C_{cap}^{(r_i)k} = \sum_{i=0}^n Q_k - E_i^k, \text{ pour les arrêts de la tournée tel que } A_i^k \in [e_j, l_{j+1}]$$

### Coefficient $m^{(r_i)k}$ global

---

Le coefficient m est donc la somme pondérée de ces deux critères

$$m^{(r_i)k} = \alpha \cdot C_{cap}^{(r_i)k} + \beta \cdot C_{wait}^{(r_i)k}$$

### Calcul de $d^{(r_i)k}$

---

$d^{(r_i)k}$  représente la distance entre une tournée et une requête. Dans notre cas, la distance est bien sur spatiale, mais aussi temporelle. En effet, on peut imaginer qu'il est plus facile et plus intéressant d'insérer une requête dans une tournée qui se trouve à proximité et qui tourne sur les mêmes plages horaires.

Nous allons pour cela utiliser un calcul de distance en trois dimensions. Les deux premières dimensions représentent la position des sommets (X,Y). La 3<sup>ème</sup> dimension représente le temps. De plus, la distance entre la requête et la tournée revient à calculer une distance entre un segment et un polygone. Nous allons considérer quatre points particuliers pour calculer cette distance :

- L'arrêt de départ de la requête entrante, noté  $V_d$
- L'arrêt d'arrivée de la requête entrante, noté  $V_a$
- L'arrêt appartenant à la tournée, noté  $V_i$ , tel que :

$$d_i = \min_{i=[0..N]} \left( \sum_{k=1}^3 (V_d[k] - V_i[k])^2 \right)$$

C'est donc le sommet le plus proche de  $V_d$ .

- L'arrêt appartenant à la tournée, noté  $V_j$ , tel que :
- 

$$d_j = \min_{i=[0..N]} \left( \sum_{k=1}^3 (V_a[k] - V_i[k])^2 \right)$$

C'est donc le sommet le plus proche de  $V_a$ .

**N.B :** La 3<sup>ème</sup> dimension est celle du temps. Elle est obtenue de la manière suivante :

$$\begin{cases} V_x[3] = A_x^k, & \text{si } x \text{ est un arrêt de la tournée} \\ V_x[3] = (l_x + e_x)/2, & \text{si } x \text{ est un arrêt de la requête entrante} \end{cases}$$

**N.B :** Pour obtenir de bons résultats, il faut normaliser les valeurs.

On obtient donc :

$$d^{(r_i)k} = \sum_{k=1}^3 (V_a[k] - V_i[k])^2 + \sum_{k=1}^3 (V_a[k] - V_j[k])^2$$

Le premier terme renvoie la distance minimale selon les 3 dimensions entre le sommet de départ de la requête et les sommets de la tournée. Le second terme fait de même mais entre le sommet d'arrivée de la requête et les sommets de la tournée.

## Faisabilité d'une insertion

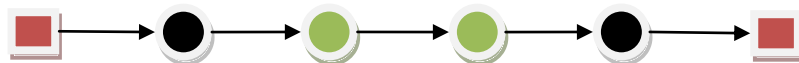
Nous avons maintenant une requête sélectionnée pour être insérée et une tournée pour l'accueillir. Il nous faut maintenant vérifier si pour une position donnée, la requête peut être insérée dans la tournée. Pour cela, il faut vérifier que les contraintes de capacité et les contraintes de fenêtre de temps sur les requêtes statiques qui composent la tournée sont toujours respectées après l'ajout de la nouvelle requête.

### Vérification de la capacité

Tout d'abord nous devons vérifier que la contrainte de capacité définie précédemment est toujours respectée.

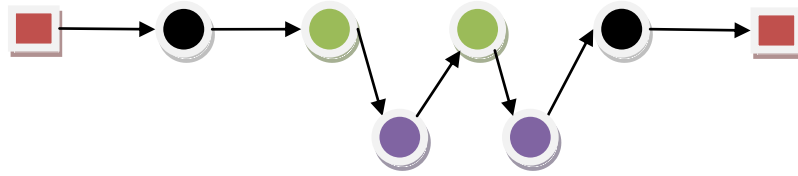
$$E_j^k + q_j * X_{ij}^k \leq Q^k$$

Voyons sur ces deux schémas la modification qui est opérée quand on insère une requête dans un ordonnancement déjà existant. On note q le nombre de personnes montant (ou descendant) à chaque arrêt et E le nombre de personnes dans le bus après le service de cet arrêt (la charge).



| Arrêt |   |   |   |    |    |   |
|-------|---|---|---|----|----|---|
| q     | 0 | 2 | 3 | -3 | -2 | 0 |
| E     | 0 | 2 | 5 | 2  | 0  | 0 |

Figure 8 : Charge de la tournée avant insertion



| Arrêt |   |   |   |   |    |    |    |   |
|-------|---|---|---|---|----|----|----|---|
| q     | 0 | 2 | 3 | 1 | -3 | -1 | -2 | 0 |
| E     | 0 | 2 | 5 | 6 | 3  | 2  | 0  | 0 |

Figure 9 : Charge de la tournée après insertion

Sur cet exemple, on remarque que la valeur de la charge  $E$  n'est modifiée que pour la partie de la tournée située entre les deux nouveaux sommets. Nous n'avons donc qu'à mettre à jour et à vérifier la contrainte pour cette sous-partie de la tournée.

Notons  $i$  la position d'insertion de l'arrêt de départ de la requête dynamique, et notons  $j$  la position de l'arrêt d'arrivée de cette requête. Pour tous les arrêts  $x$  de la tournée  $k$  compris entre  $i$  et  $j$  (inclus), on met à jour  $E_x^k = E_{x-1}^k + q_x$ . Si  $E_x^k \geq Q^k$ , alors l'insertion n'est pas possible.

### Vérification des fenêtres de temps

Nous sommes dans le cadre d'un TAD dynamique. Comme nous l'avons expliqué, la tournée de base sur laquelle nous essayons d'insérer de nouvelles requêtes a été calculée à partir de réservations enregistrées au préalable et validées. On peut imaginer par exemple qu'une personne a réservé la veille, et que l'on s'est engagé sur des horaires de temps précis avec ce client. Nous avons à notre disposition en entrée de notre problème ces fenêtres de temps. Dans le cadre de nos engagements, nous ne pouvons pas nous permettre de sortir de ces limites, et donc nous devons vérifier que l'insertion de notre requête dynamique ne rentre pas en contradiction avec ces règles.

Notons  $i$  la position d'insertion de l'arrêt de départ de la requête dynamique, et notons  $j$  la position de l'arrêt d'arrivée de cette requête.

### Les arrêts situés avant $i$

Pour ces arrêts, nous n'avons pas besoin de faire de mise à jour ou de vérifier la contrainte, puisque rien n'est modifié.

### Les arrêts situés entre $i$ (inclus) et $j$ (exclus)

Pour ces arrêts, il faut mettre à jour l'heure d'arrivée, éventuellement le temps d'attente, et vérifier les contraintes de temps. On commence par calculer l'heure d'arrivée au sommet  $i$ .  $A_i^k = A_{i-1}^k + s_{i-1} + D_{(i-1)i}$ . L'heure d'arrivée est égale à l'heure d'arrivée au sommet précédent, auquel on ajoute le temps de service et la distance entre les deux arrêts.

Si  $A_i^k < e_i$ , on est en avance. On peut alors décaler l'heure d'arrivée à l'arrêt en ajoutant un temps d'attente. On obtient alors  $W_i^k = e_i - A_i^k$  et  $A_i^k = e_i$ .

Si  $A_i^k > l_i$ , ce n'est pas important car il s'agit d'un arrêt « dynamique », on a donc le droit de dépasser les fenêtres de temps. Cette violation se répercutera sur la qualité du service.

Pour la vérification des arrêts suivants, on utilise le même processus. On met à jour l'heure d'arrivée, éventuellement le temps d'attente si on arrive en avance. Par contre, comme il s'agit maintenant d'arrêts statiques, si on obtient  $A_i^k > l_i$ , alors l'insertion est impossible.

### Introduction de la valeur *gap*

En observant la structure de notre tournée, on peut remarquer que le fait d'insérer notre arrêt à la position  $i$  aura toujours pour effet de déplacer le reste de l'ordonnancement « vers la droite ». En effet, comme on peut le voir sur le schéma suivant, le véhicule va devoir faire un détour pour servir le nouvel arrêt. Sachant que le plus court chemin pour aller d'un point à un autre est la ligne droite, le véhicule ne pourra pas arriver plus tôt qu'avant à l'arrêt  $i+1$ .

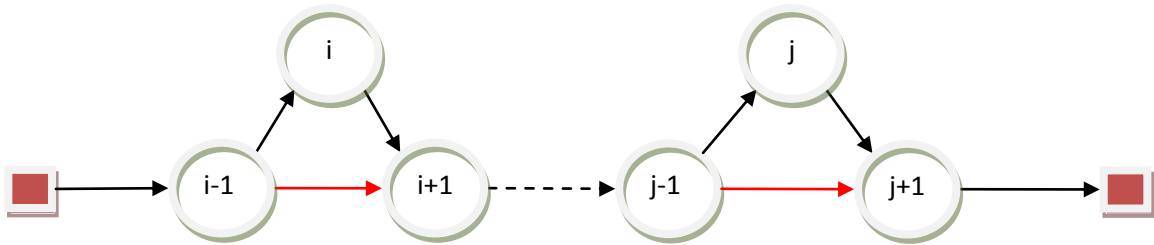


Figure 10 : Modification de la route lors d'une insertion

Dans le meilleur des cas, l'arrêt  $i$  peut être situé sur l'ancienne route pour aller de  $i-1$  à  $i+1$ . Dans ce cas, et si le temps de service  $s_i$  est nul, alors le véhicule arrivera à la même heure à l'arrêt  $i+1$  qu'avant l'insertion. Nous allons nous servir de cette propriété pour introduire une valeur *gap* qui représente la quantité de décalage vers la droite effectuée pour insérer le sommet  $i$ .

Ensuite, on peut noter que si la valeur du *gap* venait à devenir nulle, alors la suite de la tournée (jusqu'à la position  $j$ ) sera inchangée par rapport à la tournée initiale. Nous allons donc utiliser cette valeur dans notre itération pour tenter de gagner un peu de temps dans la vérification.

Le but est donc « d'absorber », ce décalage, pour se retrouver dans la situation initiale. Pour ce faire, nous allons utiliser les temps d'attente. En effet, pour arriver plus tôt à un sommet qui possède du temps d'attente, il suffit d'attendre moins longtemps avant de le servir. Donc, pour chaque arrêt  $x$  après  $i$  et avant  $j$ , si  $W_x^k > 0$ , alors on met à jour

$$\begin{cases} gap = \max(0, gap^* - W_x^k) \\ W_x^k = \max(0, W_x^k - gap^*) \end{cases}$$

Si on arrive à une itération pour laquelle  $gap = 0$ , on sait que la suite de la tournée ne sera pas modifiée et donc on peut passer directement à la vérification de l'étape suivante.

### Les arrêts situés entre $j$ (inclus) et le retour au dépôt

Pour cette partie de la tournée, on peut reprendre exactement la même procédure que précédemment, en remplaçant  $i$  par  $j$  et  $j$  par le dépôt. On peut toujours utiliser la notion de *gap*. Quand on atteint le dépôt, si tout le décalage à droite n'a pas été absorbé, alors on décale simplement l'heure de retour au dépôt de la valeur nécessaire.

## Positions d'insertion

Normalement, nous devons tester toutes les positions possibles d'insertion pour l'arrêt de départ et l'arrêt d'arrivée de la requête à insérer. Ce qui représente à tester  $N = (n+2)(n+1)/2$  configurations. On peut faire ceci en plaçant au départ de la tournée les deux arrêts de la requête. Puis en effectuant simplement un décalage à droite de la position d'insertion de l'arrêt d'arrivée, puis un décalage à droite de la position de l'arrêt de départ quand nous sommes arrivés à la fin de la tournée.

```
Booleen iterationSuivante(positionDepart, positionArrivee)
    Si (positionArrivee != FIN_TOURNEE) Alors
        positionArrivee ++
        Retourner VRAI
    Sinon // l'arrêt d'arrivée est complètement à droite
        Si (positionDepart != FIN_TOURNEE) Alors
            positionDepart ++
            positionArrivee = positionDepart
            Retourner VRAI
        Sinon // On a testé toutes les itérations
            Retourner FAUX
    Fin Si
Fin Si
```

Nous pouvons sensiblement diminuer ce nombre en appliquant quelques règles d'éliminations simples.

### Initialisation avec l'heure de soumission

Comme nous l'avons montré dans la partie « Définition » du problème de transport à la demande, l'heure de soumission des requêtes dynamiques est un élément important du problème. En effet, la tournée où l'on essaye d'insérer la requête ne peut pas être modifiée depuis le début, mais seulement à partir des sommets qui n'ont pas été servis. La première position d'insertion possible n'est donc pas  $i=0$ , mais juste après le premier sommet encore non servi.

### Dépassement de la capacité entre $i$ (inclus) et $j$

Si on observe un dépassement de capacité après insertion de la requête sur la portion de route entre  $i$  (inclus) et  $j$ , alors il ne sert à rien de tester les autres configurations qui placent  $j$  plus tard dans la tournée ou  $i$  avant le sommet qui pose problème. Il faut directement placer  $i$  et  $j$  après ce sommet problématique comme le montre le schéma suivant.

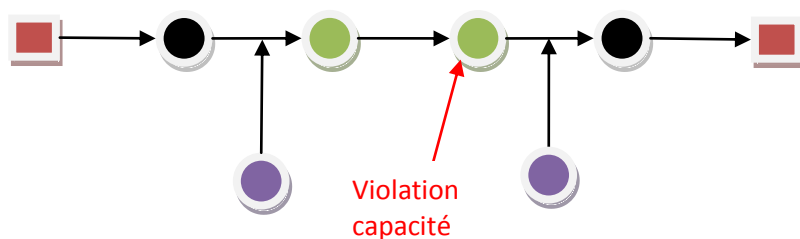


Figure 11 : Violation de la contrainte de capacité lors d'une insertion

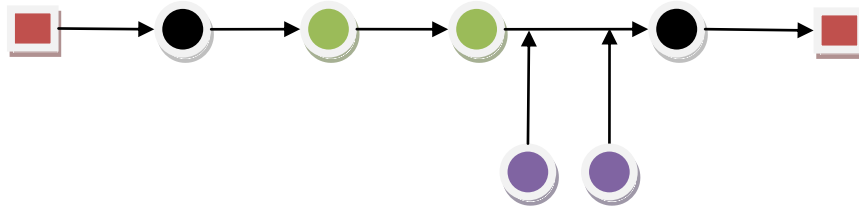


Figure 12 : Position suivante d'insertion après détection d'une violation de capacité

## Front de Pareto

Si une insertion est évaluée comme faisable (pas de violation des contraintes), alors on évalue ces trois fonctions objectif. Puis nous devons la comparer aux autres solutions déjà faisables. Comme il s'agit d'un problème multicritères, nous n'avons pas forcément une solution qui domine les autres mais on peut avoir un ensemble de solutions qui sont dites «non dominées» entre elles. Deux solutions sont dites «non dominées» entre elles si chacune d'entre elle a au moins un critère strictement inférieur (domination au sens fort) que l'autre.

Exemple :

- $A(100,100,100)$  domine  $B(200,200,200)$  si on cherche à minimiser
- $A(100,100,100)$  domine  $B(100,100,101)$
- $A(100,100,100)$  et  $B(99,200,445555555)$  ne se dominent pas

On obtient donc un ensemble de solutions non dominées, nommé Front de Pareto, que l'on peut représenter ainsi pour un problème à deux critères :

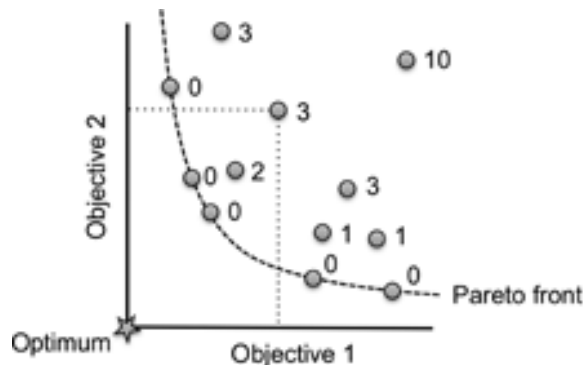


Figure 13 : Front de Pareto en deux dimensions

## Distance de Chebyshev

Après avoir testé toutes les insertions possibles pour une requête, nous obtenons donc un ensemble de solutions non dominées. Or, nous avons besoin d'une unique solution pour pouvoir poursuivre notre algorithme en insérant une nouvelle requête. Pour cela, il nous faut évaluer les solutions afin d'obtenir une valeur unique (et non trois critères) qui nous permettra de décider laquelle est la « meilleure ». Nous avons décidé d'utiliser la distance de Chebyshev plutôt qu'une simple distance euclidienne par exemple, car cette distance permet d'effectuer une normalisation sur les trois dimensions de nos solutions. C'est important car la valeur de la qualité de service a tendance à être très importante par rapport aux valeurs des deux autres critères.

La formule pour l'évaluation d'une solution  $z$  à  $k$  critères est la suivante :

$$||z - zut|| = \max_{1 \leq i \leq k} (\lambda_i |z_i - zut_i|) + \varepsilon \sum_{i=1}^k |z_i - zut_i|$$

$$\text{Avec } \begin{cases} zut, \text{ une solution inatteignable de référence} \\ \varepsilon = 10^{-3} \\ \lambda_i = 1/\text{Max}(z_i) \end{cases}$$

$\lambda_i$  permet de normaliser les trois critères selon le plus grand des trois. Nous avons choisi comme solution de référence  $zut=[0,0,0]$ .

## Optimal Timing

---

L'algorithme d'optimal Timing permet d'ajuster au mieux les horaires de passage aux arrêts dans une tournée. Il ne touche à aucun moment à l'ordonnancement qu'on lui fournit. Le but est d'améliorer la qualité de service globale de la tournée. Nous n'avons pas développé cet algorithme durant ce PFE. Il prend simplement en entrée la solution que nous avons retenue grâce à l'évaluation par la distance de Chebyshev, et nous renvoie la version optimisée de cette solution. C'est un algorithme assez complexe qu'il serait trop long d'expliquer ici et dont l'étude ne faisait pas partie des objectifs de ce PFE.

L'idée principale de l'algorithme est en fait de diviser ce problème d'optimisation en plusieurs sous problèmes plus simple à résoudre. En effet, si on regarde la façon dont est calculée la qualité de service, on voit qu'elle tient compte des temps d'attente effectués par le véhicule, mais seulement si des passagers sont à bord. En d'autres termes, faire attendre un véhicule vide ne modifiera pas la qualité de service. Cette observation permet donc de diviser notre problème à chaque fois que l'on se retrouve dans la situation où le bus est vide. Une fois chaque sous problème optimisé, il suffit de réassembler la tournée complète. S'il existe des décalages au niveau des connexions des sous-problèmes, il suffit d'augmenter la valeur des temps d'attente à cet endroit. Comme de toute façon le bus sera vide à cet endroit, cela n'aura pas de conséquences sur la qualité de service.

## Algorithme

---

Avec tous les éléments précédemment développés, nous pouvons à présent mettre en place notre méthode heuristique par insertion.

```
S1 : sommet de départ de la requete dynamique
S2 : sommet d'arrivée de la requete dynamique
Requetes[] : liste des requetes à inserer
Tournées[] : liste des tournées disponibles

Trier les requêtes par difficulté d'insertion
Pour chaque Requete R de Requetes[] Faire
    S1 = le sommet de départ de R
    S2 = le sommet d'arrivée de R
    Trier les tournées par degré d'attraction
    Pour i de 0 à MAX_TOURNEE Faire // On ne regarde que les premières
        Initialiser position S1 et S2
        Tant qu'il reste des insertions possibles Faire
            Vérifier faisabilité de l'insertion
            Si l'insertion est faisable Alors
                Calculer les fonctions objectif
                Insérer dans le front de Pareto
                Placer S1 et S2 à la position suivante

            Sinon // On a violé une contrainte
                Si violation de capacité sur sommet X Alors
                    Placer S1 et S2 juste après X
            Fin Si
        Fin Si
    Fin Tant Que
Fin Pour
Choisir la meilleure solution
Fin Pour
Appliquer l'Optimal Timing
```

# Méthode arborescente

---

## Description générale

---

Dans cette deuxième partie de PFE, nous allons mettre en place une méthode arborescente basée sur l'heuristique RBS (Recovering Beam Search). Le but de l'heuristique Beam Search est d'explorer une arborescence en visitant les nœuds les plus prometteurs dans un ensemble de nœuds choisi de taille limitée. La Recovering Beam Search (RBS) est une variante de la BS, incluant une recherche dans le voisinage des nœuds sélectionnés.

Contrairement à la première méthode développée, l'idée ici est de remettre en cause la structure des tournées (seulement la partie modifiable en fonction de l'heure), pour tenter de trouver des solutions de meilleure qualité. L'utilisation d'un faisceau limité va permettre de limiter la taille des solutions explorées et donc rester dans des temps de calcul raisonnables. Attention, du fait de la limitation du faisceau notamment, tous les états finaux de l'arbre ne sont pas explorés. Il ne s'agit donc pas d'une méthode exacte.

Une RBS est composée des éléments suivants :

- Un état initial
- Un schéma de branchement
- Des filtres
- Une évaluation
- Une sélection
- Une récupération

Nous allons maintenant décrire ces éléments un par un, de manière générale, puis appliqués à notre problème.

## L'état initial

---

Nous avons décidé de remettre en cause toutes les tournées ou parties de tournées qui peuvent l'être, c'est-à-dire les arrêts qui n'ont pas été desservis. Si l'on reprend l'exemple de la solution avec deux tournées donnée au début de ce rapport :

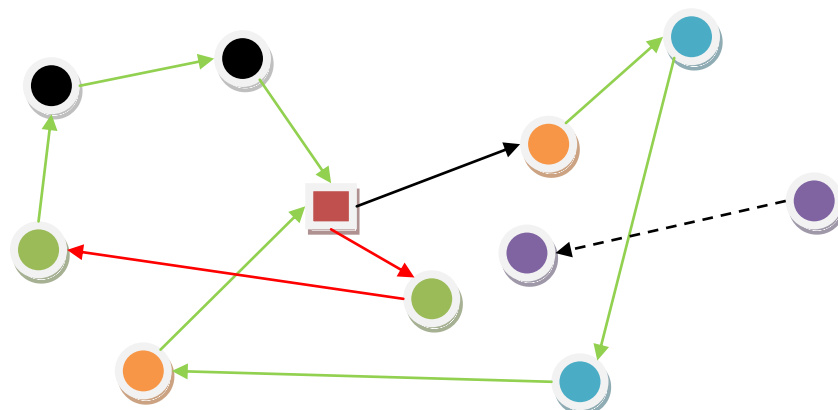


Figure 14 : Etat des tournées au temps T avant modification (les tronçons modifiables en vert)

Les trajets non encore effectués sont en vert. Si on les supprime, nous avons alors notre solution initiale qui devient :

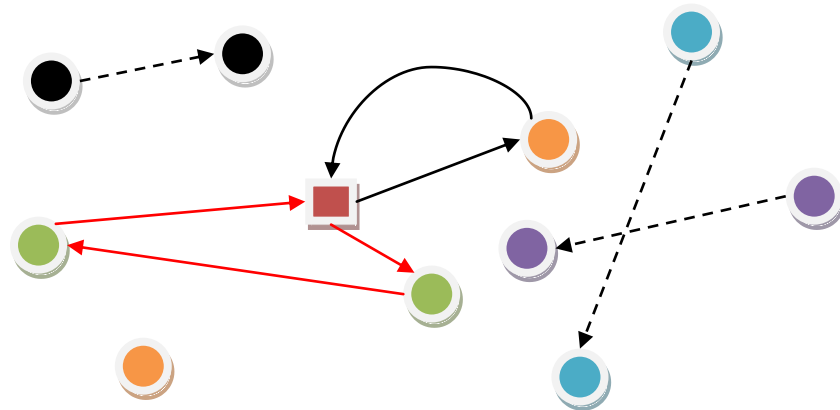


Figure 15 : Etat des tournées après modification

Bien sûr ce sont des solutions incomplètes, par exemple la route noire ne peut pas exister dans cet état, car le point de descente attaché au point de montée orange n'est pas desservi. Dans notre méthode, une solution ne pourra être garantie complète que si elle appartient au tout dernier niveau de l'arborescence que nous allons créer.

En plus de ces tournées tronquées, l'état initial est aussi constitué d'un ensemble d'arrêts non desservis qu'il va falloir insérer grâce au schéma de branchement. Il faut bien garder à l'esprit pour la suite que même si tous ces sommets sont mélangés dans la même liste, ils ont des caractéristiques différentes qui vont influencer sur la méthode quand nous allons vouloir les insérer. La liste des sommets non insérés est composée de :

- Sommets issus des requêtes dynamiques entrantes. Ces sommets sont libres d'être affectés à n'importe quel véhicule, et leurs fenêtres de temps peuvent être dépassées si besoin en dégradant la qualité de service.
- Sommets issus des requêtes statiques non commencées. Ce sont les requêtes qui étaient planifiées mais dont les clients concernés ne sont pas encore montés dans le véhicule. Ces sommets sont libres d'être affectés à n'importe quel véhicule, mais leurs fenêtres de temps sont inviolables.
- Sommets issus des requêtes en cours de traitement. Ce sont des sommets de descente, liés à des sommets de montée déjà servis, ce qui signifie que les clients sont en ce moment dans un véhicule précis. Ces sommets seront donc obligatoirement affectés au même véhicule, et leurs fenêtres de temps sont inviolables.

Cette liste de sommets est alors triée selon un ordre de priorité et les sommets seront insérés par la RBS au fur et à mesure, et dans cet ordre. Nous avons choisi de trier les sommets par la valeur de leur  $l_i$  (heure de passage au plus tard).

**N.B :** Attention, cet ordonnancement doit garder la contrainte de précédence entre un sommet de montée et son sommet de descente associé. Ainsi, un sommet de descente ne peut pas être placé avant son sommet de montée, même si son  $l_i$  est inférieur. Cela reviendrait à faire descendre les clients du véhicule avant de les avoir fait monter.

## Le schéma de branchement

Le schéma de branchement permet, à partir d'un niveau de l'arbre donné, de créer le niveau suivant. C'est-à-dire l'ensemble des fils créé à partir de cette règle de branchement.

### La règle d'affectation

A chaque niveau de l'arborescence, on insère le prochain sommet issu de la liste triée des sommets à insérer. A partir d'un nœud père, on crée alors un ensemble de nœud fils en affectant ce sommet à l'ensemble des véhicules possibles. Si le sommet est lié à un véhicule, nous n'aurons qu'un seul nœud fils. Si le sommet est libre, nous aurons autant de nœuds fils que de véhicules.

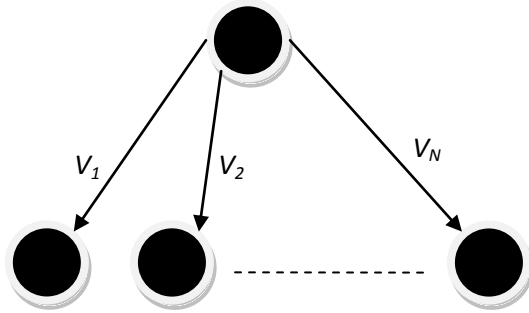


Figure 16 : Branchement pour un sommet libre

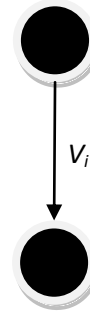


Figure 17 : Branchement pour un sommet lié

### La règle d'ordonnancement

Pour chaque nœud fils créé, nous avons choisi un véhicule pour y affecter le nouveau sommet. Cependant, nous devons aussi déterminer à quel endroit le placer dans l'ordonnancement déjà existant. Pour cela, nous allons utiliser une règle que nous nommerons « insertion au plus tard possible ». C'est-à-dire que nous allons chercher à insérer le nouveau sommet à la fin de l'ordonnancement, mais en respectant les contraintes des fenêtres de temps qui existent (si le sommet à insérer est de type « statique »). Nous avons donc deux possibilités :

- Le sommet est de type « dynamique ». Dans ce cas, nous n'avons pas de contraintes sur ces fenêtres de temps, le sommet (en violet) est donc inséré à la fin de l'ordonnancement, juste avant le dépôt (en rouge).

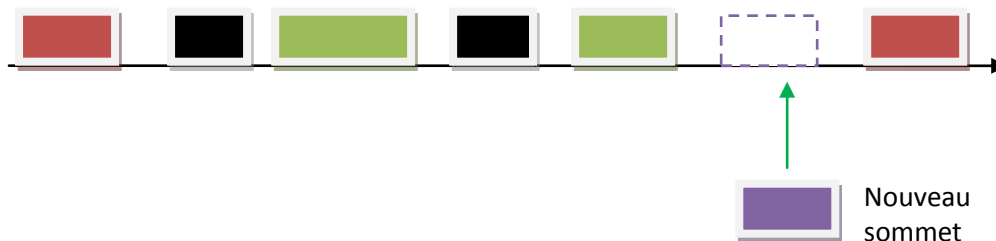


Figure 18 : Affectation d'un sommet dynamique

- Le sommet est de type « statique ». Dans ce cas, nous allons remonter l'ordonnancement jusqu'à respecter la contrainte suivante :

$$l_i \geq A_{i-1}^k + s_{i-1} + D_{(i-1)i}$$

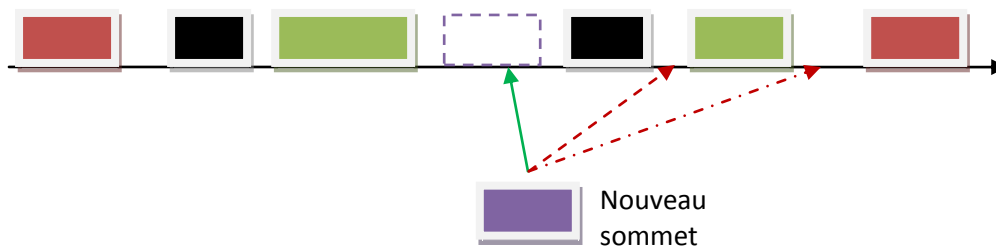


Figure 19 : Affectation d'un sommet statique

Cette règle simple, qui ne demande qu'un parcours de l'ordonnancement ( $O(N)$ ), nous permet de créer des solutions «valides». En effet, en insérant les sommets «statiques» à la fin de l'ordonnancement comme nous le faisons pour les sommets dynamiques, nous générerions la plupart du temps des solutions non valides. Cela entraînerait directement la coupe de cette branche de l'arbre affectant le sommet  $S$  au véhicule  $V$ , et donc la non-exploration des solutions pour lequel le sommet est affecté à ce véhicule.

## Les filtres

Dans une recherche par faisceau, les filtres doivent permettre d'éliminer rapidement ( $O(N)$ ) et de manière (presque) optimale, de mauvais nœuds. Cela peut être fait en installant des règles de dominance entre les nœuds par exemple, le nœud dominé sera alors coupé.

Dans notre cas, nous n'allons pas mettre pour le moment de telles règles en place. Elles pourront être ajoutées par la suite si la phase de test de la méthode ne donne pas des résultats concluants. Cependant, nous allons quand même mettre en place deux filtres, qui vont nous permettre de vérifier la « faisabilité » d'une solution. Nous allons utiliser les deux règles que nous avons déjà vu dans la partie « Heuristique par insertion » de ce rapport :

- Vérification de la capacité :  $E_j^k + q_j * X_{ij}^k \leq Q^k$ , pour tous les sommets  $j$  de la tournée  $k$
- Vérification des fenêtres de temps :  $A_i^k \leq l_i$  et  $A_i^k \geq e_i$ , pour tous les sommets  $j$  « statiques » de la tournée  $k$

**N.B.** Pour plus d'informations sur la manière de vérifier efficacement ces deux contraintes, voir la partie « Faisabilité d'une insertion » dans le chapitre sur l'heuristique par insertion. Les méthodes appliquées dans cette partie sont aussi applicables ici en ajoutant juste la notion de sommets « statiques » et « dynamiques » pour la vérification des fenêtres de temps à la position  $i$ . De plus, il faut supprimer la vérification du deuxième sommet inséré, car ici nous n'insérons qu'un sommet à la fois.

## L'évaluation

L'évaluation des nœuds est l'élément qui va permettre de guider la méthode vers des solutions ou vers d'autres. En effet, cette évaluation, si elle est bien faite, doit permettre de sélectionner à un niveau donné les nœuds les plus prometteurs et d'écarter les autres. Il existe plusieurs méthodes pour évaluer ces nœuds :

- comme dans une Procédure par Séparation et Evaluation (PSE) : on met en place une borne inférieure pour chaque nœud  $LB_S$ , et une borne supérieure globale à l'arbre  $UB_{GLOBAL}$ . Si  $LB_S \geq UB_{GLOBAL}$ , alors le nœud  $S$  est coupé.
- Pour chaque nœud  $S$ , on a une borne inférieure et une borne supérieure, on ne coupe pas de nœuds directement, ils seront sélectionnés dans la prochaine étape.

Dans notre problème, il est très difficile de calculer une borne inférieure qui serait fiable. Nous avons décidé de nous passer d'une borne inférieure et de ne travailler qu'avec une borne supérieure. Cette borne supérieure doit être la plus précise possible, mais très rapide en temps de calcul, car elle sera appelée à chaque nouveau nœud généré.

Nous avons décidé d'utiliser pour calculer notre borne supérieure, notre heuristique d'insertion développée dans la première partie du PFE. Nous la modifions juste pour permettre l'insertion des sommets un par un, et non par couple comme précédemment. L'heuristique prend alors en entrée :

- La solution partielle du nœud courant comme solution initiale
- La liste des sommets restants à insérer
- L'heure de soumission

**N.B :** De même que pour la vérification des contraintes tout à l'heure, il faut modifier légèrement l'heuristique pour prendre en compte les caractéristiques des différents sommets à insérer : libre/lié à un véhicule et statique/dynamique.

Dans la version originelle de l'heuristique, nous n'insérons que des requêtes dites « dynamiques ». Elles ont la particularité d'avoir des fenêtres de temps « souples », c'est-à-dire que si besoin on pouvait servir le client en dehors des horaires qu'il nous avait communiqués. Cette caractéristique permet d'affirmer que l'heuristique donnera toujours une solution, car dans le pire des cas on peut insérer la requête à la toute fin de la tournée.

Ici, nous utilisons l'heuristique en essayant d'insérer certains sommets « statiques ». Ces sommets ont des fenêtres « dures », on ne peut pas servir le client en dehors de ces horaires. Dans ce contexte, nous ne pouvons plus garantir que l'heuristique donnera une solution complète, mais peut être seulement une solution partielle, avec une liste de sommets restants à insérer.

Une fois la solution courante planifiée complètement ou partiellement, on lui donne une évaluation par la méthode suivante:

- Si la solution est complète, alors on calcule ces trois fonctions objectif
- Si la solution est partielle, on calcule aussi les objectifs mais on rajoute une pénalité à la valeur de chacun d'eux. Cette pénalité est de l'ordre de  $\alpha$  (une constante) multiplié par le nombre de sommets restants.

On part en effet du principe que même si l'heuristique n'a pas trouvé une solution complète, il ne s'agit que d'une borne supérieure. On peut donc espérer qu'il existe une meilleure solution que celle calculée et il est donc possible qu'on obtienne finalement une solution complète.

## La sélection

---

Nous avons maintenant un ensemble de nœuds, contenant tous une solution partielle, qui ont été évalués grâce à l'heuristique par insertion. Dans une recherche par faisceau classique, le but de la phase de sélection est de ne garder qu'un nombre  $W$  de nœuds ( $W$  étant la taille du faisceau), afin de garantir la complexité en temps polynomial de notre algorithme. Classiquement, on sélectionne les nœuds en calculant une valeur  $f(s)$  définie comme :

$$f(s) = \alpha.LB_s + (1 - \alpha).UB_s$$

Les paramètres  $\alpha$  et  $(1 - \alpha)$  permettent de favoriser l'une ou l'autre des bornes en fonction de sa qualité. Mais nous ne disposons pas de borne inférieure. De plus notre borne supérieure n'est pas une valeur scalaire, mais une combinaison de trois valeurs. Nous pourrions utiliser une combinaison linéaire de ces critères, ou alors mettre en place un algorithme de sélection totalement différent, mais nous décidons de simplement sélectionner tous les nœuds qui appartiennent au front de Pareto.

**N.B :** Idéalement, la valeur de la pénalité choisie dans la phase d'évaluation doit permettre de supprimer ici les solutions incomplètes s'il existe des solutions complètes, leurs valeurs de fonction objectif devant être supérieures aux autres (à cause de la pénalité).

## La récupération

---

Il s'agit de la dernière phase de la méthode. Ici nous allons chercher à améliorer nos solutions, en effectuant une recherche locale sur les nœuds qui ont été sélectionnés précédemment. Le but est notamment de modifier l'ordonnancement car, comme nous l'avons décrit précédemment, nous n'avons testé que des solutions en « insérant au plus tard possible » le nouveau sommet. Nous allons donc utiliser un opérateur de voisinage pour modifier cet ordonnancement. Dans l'ensemble des voisins générés « faisables », nous devons choisir le meilleur, puis le comparer à la solution initiale. Si le voisin sélectionné s'avère de meilleure qualité que la solution initiale, il prend sa place et on réitère l'opération jusqu'à ne plus trouver de voisins améliorants la solution.

### Opérateur de voisinage

---

L'opérateur de voisinage utilisé se nomme « Insertion sur le dernier sommet ». Il consiste simplement à prendre le sommet qui se trouve en fin d'ordonnancement, et de l'insérer à toutes les positions possibles de l'ordonnancement :

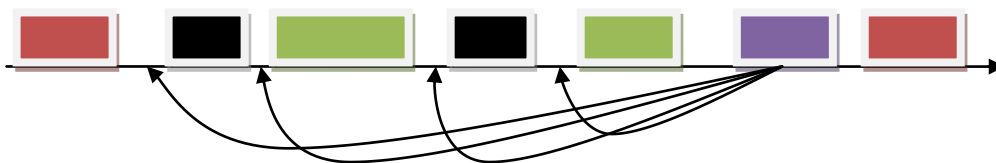


Figure 20 : Opérateur de voisinage "Insertion sur le dernier sommet"

**N.B :** Attention, nous avons modifié légèrement notre schéma de branchement en insérant le sommet « au plus tard possible » et non à la fin. Lors de la première itération dans la phase de récupération, il faut bien déplacer le sommet que nous venons d'insérer, et donc pas forcément le dernier de l'ordonnancement. Par contre, si on effectue des itérations supplémentaires, on prendra alors toujours le dernier élément de l'ordonnancement.

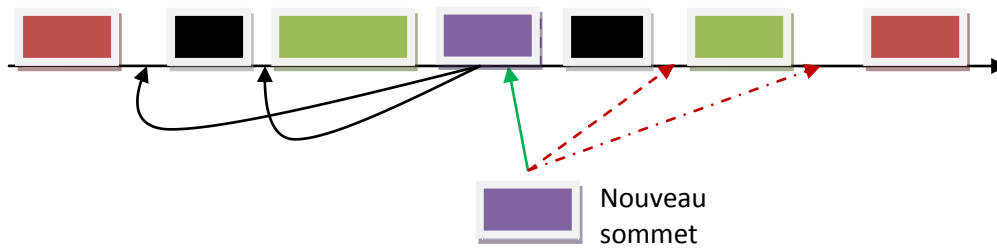


Figure 21 : Modification de l'opérateur de voisinage pour les sommets statiques

## Comparaison voisins

A partir de cette liste de voisins, on ressort la liste des voisins « faisables » en vérifiant les contraintes suivantes :

- Si le sommet déplacé est un sommet de descente, il doit être après son point de montée correspondant
- On ne peut pas modifier l'ordonnancement des sommets déjà placés dans l'état initial (au début de la RBS), car ils ont déjà été servis et l'heure est déjà passée.
- Le nouvel ordonnancement doit respecter les contraintes de capacité
- Le nouvel ordonnancement doit respecter les contraintes sur les fenêtres de temps « dures »

Nous obtenons ainsi une liste de voisins « faisables », qui vont être comparés. Pour cela, on effectue une simple combinaison linéaire sur les critères de coût et de qualité de service (on ne prend pas en compte les émissions de CO<sub>2</sub>, car pour un même véhicule, le coût et l'émission de CO<sub>2</sub> sont liés).

$$f(s') = TC + \varepsilon.QOS, \varepsilon \text{ constante}$$

**N.B :** Les valeurs des deux critères sont calculées sur les solutions partielles directement, on n'utilise pas l'heuristique comme dans la phase d'évaluation.

# Tests

---

Dans la dernière partie de ce PFE, nous avons mis en place une campagne de tests afin d'évaluer la qualité des choix que nous avons mis en place. Dans un premier temps, nous avons sorti des statistiques sur notre heuristique par insertion, puis sur l'heuristique RBS. Dans un deuxième temps, nous avons essayé de comparer ces deux méthodes entre elles, ainsi qu'avec des résultats calculés sur les mêmes jeux de données à l'aide de la méthode NSGAI.

## Méthodologie

---

Nous disposons d'un jeu de test, composé de la manière suivante :

- 3 fichiers «véhicules», qui contiennent respectivement 2,4 et 6 véhicules
- 100 fichiers «requêtes statiques», regroupés en 10 groupes. Chaque instance d'un groupe possède un nombre de requêtes statiques identique. Le nombre de requêtes statiques pour un groupe donné est défini entre 10 et 100, par pas de 10. On a donc 10 instances avec 10 requêtes statiques, 10 instances avec 20 ...
- 100 fichiers «requêtes dynamiques», possédant toutes 10 requêtes dynamiques. Un fichier «requêtes dynamiques» est lié à un fichier «requêtes statiques».
- 300 fichiers «solutions initiales», qui correspondent aux solutions calculées par NSGAI de manière statique, en utilisant un des trois fichiers «véhicules» et un des cent fichiers «requêtes statiques».
- 300 fichiers «solutions finales», qui correspondent aux solutions calculées par NSGAI de manière statique, en utilisant un des trois fichiers «véhicules», un des cent fichiers «requêtes statiques» et le fichier «requêtes dynamiques» associé (en traitant les requêtes dynamiques comme des requêtes statiques).

Nous générons alors nos propres solutions calculées avec nos deux algorithmes. Une instance est donc définie dans notre cas par :

- 1 fichier «véhicules»
- 1 fichier «requêtes statiques»
- 1 fichier «requêtes dynamiques» qui est associé au fichier «requêtes statiques»
- 1 fichier «solutions initiales» associé au véhicule et au fichier «requêtes statiques»

Au total, nous avons donc  $3 \times 100 = 300$  instances par algorithme à tester.

De plus, afin de tester la dégradation de la qualité des algorithmes en fonction de l'avancement de l'heure de soumission, nous avons décidé de tester ce jeu de données sur 5 heures différentes. Ces 5 heures de soumission sont choisies de manière à être espacées régulièrement dans la journée, entre le temps 0 et le temps où toutes les tournées programmées sont finies. Pour une heure donnée, seules les requêtes dynamiques dont le  $l_i$  sera supérieur à cette heure seront insérées. Les autres seront ignorées car il est impossible qu'un client appelle pour demander à être pris en charge dans le passé. Avec cette variable supplémentaire, nous avons donc  $300 \times 5 = 1500$  instances par algorithme à tester.

Ne sont présentées ici que quelques statistiques, très simplifiées afin de faire ressortir les grandes lignes de nos conclusions. Les résultats de tests complets sont disponibles en annexes de ce document.

## Statistiques sur l'heuristique par insertion

Dans cette partie, nous nous intéressons exclusivement à l'heuristique par insertion en elle-même. Nous allons pouvoir étudier notamment l'évolution du temps de calcul en fonction de la taille du problème et la qualité de notre méthode d'ordonnance des tournées.

### Evolution du temps de calcul

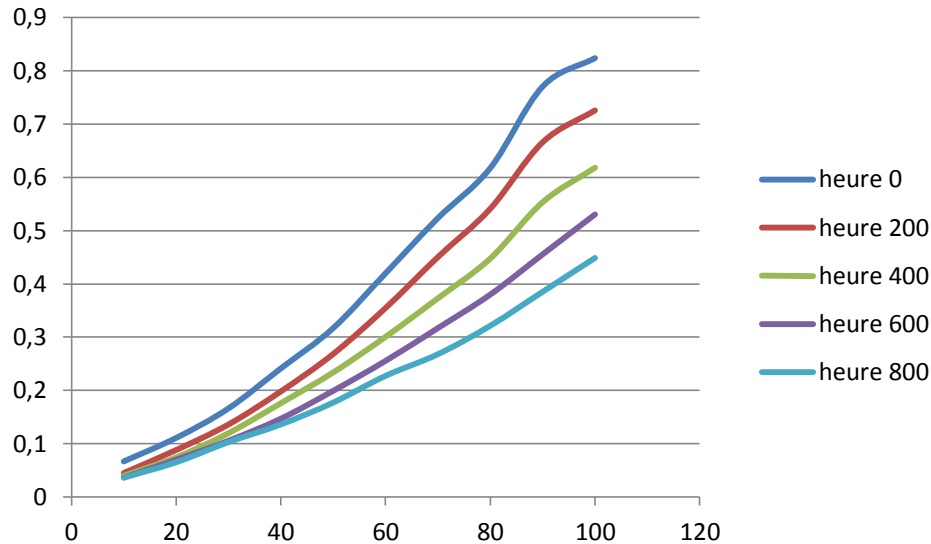


Figure 22 : Evolution du temps CPU en fonction de la taille du problème

Sur ce graphique, on peut voir l'évolution des temps de calcul pour une taille de problème donnée (nombre de requêtes statiques). On voit que cette évolution est quasiment linéaire. Même pour le problème considéré comme le plus gros dans notre jeu de données, nous restons sous la barre de la seconde qui était demandée dans le cahier des charges. De plus, ces résultats ont été produits en explorant toutes les tournées pour insérer une nouvelle requête. L'exploration des tournées étant l'un des calculs les plus coûteux dans notre algorithme, on peut estimer qu'un gain important pourrait être pris si on décide de n'explorer qu'un ensemble réduit de ces tournées.

### Position de la meilleure tournée

Ce test a justement pour but de déterminer si oui ou non, le «degré d'attraction» que nous avons défini nous permet d'ordonner les tournées pouvant accueillir une requête de la meilleure manière possible. Si c'est le cas, on pourra alors accélérer encore les calculs comme expliqué dans le test précédent.

| Nombre de vehicules | Véhicule choisi (classement) |
|---------------------|------------------------------|
| 2                   | 1,0648572                    |
| 4                   | 2,048                        |
| 6                   | 2,674                        |

Figure 23 : Efficacité du degré d'attraction

Ce tableau présente la position moyenne observée de la tournée ayant accueilli la requête à insérer. Cette moyenne est faite sur l'ensemble des instances ayant le même fichier de véhicule (sans distinction sur la taille du problème). On voit que sur l'ensemble des instances possédant 2 véhicules, notre «degré d'attraction» a bien fonctionné car il a placé quasiment à chaque fois la bonne route en première position. Sur les instances à 4 véhicules, cela nous indique qu'il aurait suffi de parcourir les deux premières routes pour obtenir la meilleure solution possible avec notre algorithme. Enfin pour les instances à 6 véhicules, le constat est le même.

Au final, on se rend compte que notre classement fonctionne bien et qu'on peut se permettre de ne parcourir que 50% des routes environ pour obtenir quasiment le même résultat qu'en parcourant 100% des routes. On peut donc obtenir un gain de temps intéressant sur l'exécution de notre heuristique.

## Statistiques sur la méthode arborescente

### Taux de réussite et d'insertion

Les tests sur la méthode arborescente ont rapidement révélé que pour bon nombre d'instances, la RBS ne nous renvoie pas de résultats. C'est tout à fait possible, il suffit que pour un niveau de l'arbre donné, tous les fils possibles soient coupés lors de la phase de filtrage. N'ayant plus de nœuds père pour générer le niveau suivant, l'algorithme s'arrête donc avec des solutions incomplètes. Le tableau suivant présente le taux de réussite de l'algorithme pour chaque type d'instance (trouve-t-on une solution complète ?) et le taux de réussite d'insertion (quel pourcentage de requêtes avons-nous réussi à insérer avant que l'algorithme ne s'arrête ?).

| Requêtes | % Réussite |       |       |       |       | % Insertion |       |       |       |       |
|----------|------------|-------|-------|-------|-------|-------------|-------|-------|-------|-------|
|          | T=0        | T=200 | T=400 | T=600 | T=800 | T=0         | T=200 | T=400 | T=600 | T=800 |
| 10,00    | 60,00      | 50,00 | 40,00 | 30,00 | 60,00 | 90,25       | 74,45 | 72,12 | 64,35 | 72,22 |
| 20,00    | 10,00      | 0,00  | 10,00 | 20,00 | 40,00 | 45,73       | 30,21 | 48,10 | 40,00 | 61,59 |
| 30,00    | 0,00       | 0,00  | 0,00  | 0,00  | 20,00 | 19,51       | 19,51 | 18,33 | 23,30 | 50,76 |
| 40,00    | 0,00       | 0,00  | 0,00  | 0,00  | 0,00  | 17,00       | 14,20 | 9,98  | 20,38 | 18,76 |
| 50,00    | 0,00       | 0,00  | 0,00  | 0,00  | 0,00  | 10,27       | 6,63  | 13,76 | 10,44 | 11,01 |
| 60,00    | 0,00       | 0,00  | 0,00  | 0,00  | 0,00  | 10,23       | 4,47  | 7,96  | 3,76  | 8,47  |
| 70,00    | 0,00       | 0,00  | 0,00  | 0,00  | 0,00  | 10,25       | 4,95  | 4,50  | 6,95  | 8,18  |
| 80,00    | 0,00       | 0,00  | 0,00  | 0,00  | 0,00  | 8,37        | 6,17  | 5,39  | 4,41  | 7,88  |
| 90,00    | 0,00       | 0,00  | 0,00  | 0,00  | 0,00  | 8,66        | 2,56  | 4,86  | 7,92  | 7,96  |
| 100,00   | 0,00       | 0,00  | 0,00  | 0,00  | 0,00  | 5,29        | 4,96  | 2,84  | 3,91  | 9,55  |

Figure 24 : Taux de réussite de la méthode RBS pour 2 véhicules

Pour les instances avec deux véhicules, notre méthode ne renvoie quasiment jamais de solutions, mise à part pour les problèmes de taille 10. Au niveau du pourcentage d'insertion, on remarque qu'il diminue, que ce soit quand la taille du problème augmente ou quand l'heure d'insertion avance. Cela signifie que plus le problème est grand, plus on est loin de réussir à obtenir une solution.

| Requêtes | % Réussite |        |        |        |       | % Insertion |        |        |        |        |
|----------|------------|--------|--------|--------|-------|-------------|--------|--------|--------|--------|
|          | T=0        | T=200  | T=400  | T=600  | T=800 | T=0         | T=200  | T=400  | T=600  | T=800  |
| 10,00    | 90,00      | 100,00 | 100,00 | 100,00 | 90,00 | 96,00       | 100,00 | 100,00 | 100,00 | 100,00 |
| 20,00    | 100,00     | 90,00  | 90,00  | 70,00  | 80,00 | 100,00      | 90,20  | 91,11  | 72,79  | 92,82  |
| 30,00    | 50,00      | 60,00  | 80,00  | 90,00  | 90,00 | 77,07       | 72,96  | 92,63  | 98,57  | 98,00  |
| 40,00    | 60,00      | 50,00  | 40,00  | 60,00  | 70,00 | 79,50       | 69,81  | 50,98  | 87,46  | 83,62  |
| 50,00    | 30,00      | 40,00  | 20,00  | 30,00  | 30,00 | 68,68       | 63,04  | 37,77  | 32,61  | 35,82  |
| 60,00    | 10,00      | 20,00  | 0,00   | 0,00   | 30,00 | 50,33       | 43,31  | 19,80  | 7,28   | 46,25  |
| 70,00    | 10,00      | 0,00   | 0,00   | 0,00   | 10,00 | 50,63       | 22,97  | 9,30   | 20,55  | 24,34  |
| 80,00    | 0,00       | 0,00   | 0,00   | 0,00   | 0,00  | 22,03       | 21,00  | 19,49  | 10,70  | 20,57  |
| 90,00    | 0,00       | 0,00   | 0,00   | 0,00   | 0,00  | 17,73       | 16,93  | 13,93  | 14,71  | 4,31   |
| 100,00   | 0,00       | 0,00   | 0,00   | 0,00   | 0,00  | 17,24       | 9,97   | 9,35   | 4,58   | 11,58  |

Figure 25: Taux de réussite de la méthode RBS pour 4 véhicules

On peut faire le même constat pour les problèmes ayant 4 véhicules. Cependant, la RBS arrive à résoudre plus d'un problème sur deux jusqu'à une taille de 40 requêtes.

| Requêtes | % Réussite |        |        |        |       | % Insertion |        |        |        |       |
|----------|------------|--------|--------|--------|-------|-------------|--------|--------|--------|-------|
|          | T=0        | T=200  | T=400  | T=600  | T=800 | T=0         | T=200  | T=400  | T=600  | T=800 |
| 10,00    | 100,00     | 100,00 | 100,00 | 100,00 | 90,00 | 100,00      | 100,00 | 100,00 | 100,00 | 92,00 |
| 20,00    | 100,00     | 100,00 | 90,00  | 60,00  | 70,00 | 100,00      | 100,00 | 91,11  | 68,15  | 82,02 |
| 30,00    | 100,00     | 100,00 | 100,00 | 100,00 | 90,00 | 100,00      | 100,00 | 100,00 | 100,00 | 90,59 |
| 40,00    | 100,00     | 100,00 | 90,00  | 100,00 | 90,00 | 100,00      | 100,00 | 94,06  | 100,00 | 90,48 |
| 50,00    | 90,00      | 70,00  | 50,00  | 50,00  | 70,00 | 96,47       | 77,88  | 55,10  | 50,91  | 78,73 |
| 60,00    | 90,00      | 90,00  | 60,00  | 40,00  | 70,00 | 95,50       | 94,47  | 63,50  | 44,47  | 75,77 |
| 70,00    | 20,00      | 20,00  | 30,00  | 30,00  | 30,00 | 61,00       | 35,82  | 44,42  | 31,51  | 41,52 |
| 80,00    | 30,00      | 30,00  | 20,00  | 30,00  | 40,00 | 61,12       | 37,61  | 38,25  | 43,40  | 52,17 |
| 90,00    | 10,00      | 20,00  | 20,00  | 20,00  | 0,00  | 51,19       | 48,97  | 47,55  | 38,17  | 8,51  |
| 100,00   | 0,00       | 10,00  | 0,00   | 10,00  | 20,00 | 39,04       | 23,34  | 12,68  | 22,65  | 24,65 |

Figure 26 : Taux de réussite de la méthode RBS pour 6 véhicules

Le constat est toujours le même, la taille de problème maximale pour des instances avec six véhicules passe à 60.

### Taille du front de Pareto

Tout ceci donne à penser que le problème vient notamment du nombre de nœuds non coupés à un niveau donné de l'arbre. On peut le vérifier en affichant la taille moyenne du front de Pareto au dernier niveau (les solutions complètes), pour les instances qui ont réussi à aboutir. Attention, il faut prendre tous ces résultats avec précautions, car certaines moyennes ne sont calculées qu'à partir d'une ou deux instances qui ont réussi sur les dix prévues initialement.

| Nombre de véhicules | Nombre de solutions |
|---------------------|---------------------|
| 2                   | 2,02                |
| 4                   | 4,160105667         |
| 6                   | 5,31871             |

Figure 27 : Nombre de solutions moyen

On voit que même en ne prenant en compte que les instances qui aboutissent, le nombre moyen de solutions proposées en sortie est très faible. Nous avons ici au mieux quelques solutions, quand on sait que l'algorithme NSGAI par exemple sort facilement une centaine de solutions pour la plupart des instances.

## Temps d'exécution

Au niveau des temps d'exécution, en ne considérant que les instances ayant réussies, on voit que les temps d'exécution sont assez raisonnables. Mais le faible nombre de nœuds développés y contribue beaucoup. Nous n'avons considéré que les instances jusqu'à 60 requêtes car, comme nous l'avons vu précédemment, les calculs sont faussés par la suite du fait du faible nombre d'instances.

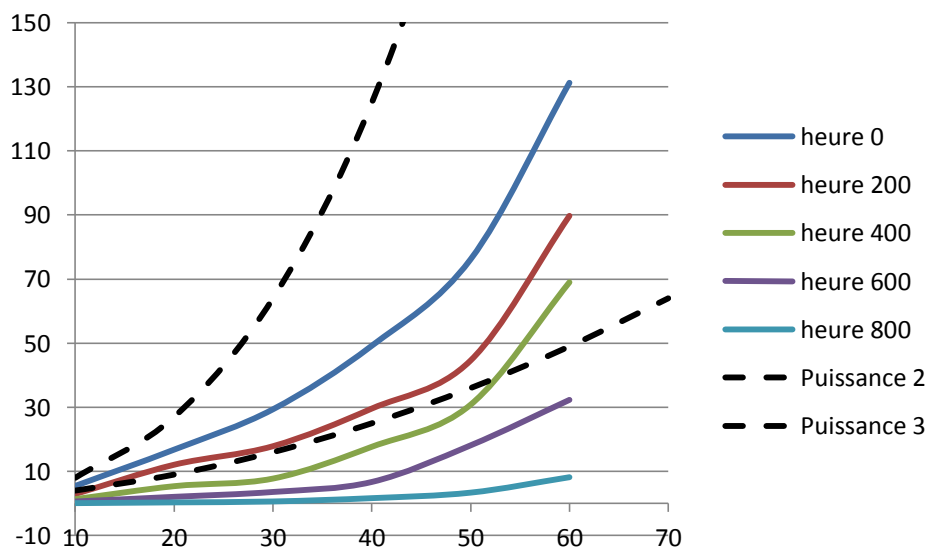


Figure 28 : Complexité en temps de la méthode RBS

D'après cette courbe, notre algorithme doit avoir une complexité en  $O(N^3)$ .

## Heuristique Insertion vs RBS

Nous allons maintenant comparer la qualité des solutions obtenues par les deux méthodes. Pour cela, pour chaque instance ayant sorti une solution complète pour les deux méthodes, nous allons extraire une solution unique de l'ensemble des solutions proposées. Cette solution sera choisie grâce à la distance de Chebyshev, déjà évoquée dans ce document. Les résultats ci-dessous montrent les valeurs des trois critères (coût, qualité, co2) pour la moyenne des solutions choisies, regroupées par taille de problème.

Comme les deux méthodes utilisent les mêmes fenêtres de temps pour les requêtes statiques, utilisent les mêmes règles d'acceptation d'une insertion et calculent les fonctions objectif de la même façon, nous pouvons les comparer.

| Taille | Flotte | Coût   |         |        | QOS      |         |        | CO2    |         |        |
|--------|--------|--------|---------|--------|----------|---------|--------|--------|---------|--------|
|        |        | Insert | RBS     | %Diff  | Insert   | RBS     | %Diff  | Insert | RBS     | %Diff  |
| 10     | 2      | 305056 | 283797  | -6,97  | 3724710  | 243821  | -93,45 | 252166 | 202505  | -19,69 |
| 10     | 4      | 292432 | 260439  | -10,94 | 4055450  | 603400  | -85,12 | 208488 | 189551  | -9,08  |
| 10     | 6      | 308123 | 274527  | -10,90 | 4233310  | 103953  | -97,54 | 171803 | 171164  | -0,37  |
| 20     | 2      | 388316 | 376082  | -3,15  | 22674200 | 1951000 | -91,40 | 338678 | 287550  | -15,10 |
| 20     | 4      | 472238 | 381016  | -19,32 | 2347660  | 1007050 | -57,10 | 376349 | 289903  | -22,97 |
| 20     | 6      | 442629 | 378874  | -14,40 | 1473780  | 772210  | -47,60 | 297081 | 261398  | -12,01 |
| 30     | 2      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 30     | 4      | 561700 | 589166  | 4,89   | 1308550  | 1650300 | 26,12  | 447760 | 448777  | 0,23   |
| 30     | 6      | 551154 | 550031  | -0,20  | 847276   | 621722  | -26,62 | 411793 | 423468  | 2,84   |
| 40     | 2      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 40     | 4      | 687511 | 714398  | 3,91   | 2333930  | 3594390 | 54,01  | 542872 | 568428  | 4,71   |
| 40     | 6      | 681344 | 721387  | 5,88   | 369615   | 647585  | 75,21  | 545012 | 564985  | 3,66   |
| 50     | 2      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 50     | 4      | 763700 | 835687  | 9,43   | 4222080  | 1621220 | -61,60 | 606738 | 670140  | 10,45  |
| 50     | 6      | 767945 | 837789  | 9,09   | 960654   | 956477  | -0,43  | 645265 | 665682  | 3,16   |
| 60     | 2      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 60     | 4      | 920223 | 957612  | 4,06   | 7108570  | 3045260 | -57,16 | 710223 | 761847  | 7,27   |
| 60     | 6      | 943673 | 1022550 | 8,36   | 1379330  | 873325  | -36,68 | 757307 | 812836  | 7,33   |
| 70     | 2      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 70     | 4      | 884820 | 1003750 | 13,44  | 2510190  | 2034050 | -18,97 | 713990 | 796715  | 11,59  |
| 70     | 6      | 933206 | 1062390 | 13,84  | 1420770  | 2806040 | 97,50  | 730094 | 844376  | 15,65  |
| 80     | 2      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 80     | 4      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 80     | 6      | 886388 | 1143160 | 28,97  | 561438   | 3545410 | 531,49 | 715216 | 906269  | 26,71  |
| 90     | 2      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 90     | 4      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 90     | 6      | 895741 | 1260390 | 40,71  | 1094870  | 9933980 | 807,32 | 716417 | 1002070 | 39,87  |
| 100    | 2      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 100    | 4      | -      | -       | -      | -        | -       | -      | -      | -       | -      |
| 100    | 6      | -      | -       | -      | -        | -       | -      | -      | -       | -      |

Figure 29 : Comparaison RBS - Heuristique par insertion

Si on analyse la première partie des résultats (jusqu'à une taille de 60), on se rend compte que la méthode RBS améliore énormément la qualité de service des solutions proposées (à l'exception des instances de taille 40). Cela est principalement dû au fait que l'on permet l'ouverture de nouvelles tournées (utiliser un véhicule qui est au garage) dans la RBS, ce qui n'est pas le cas dans l'heuristique par insertion. Concernant les deux autres critères, ils sont au début à l'avantage de la RBS puis de l'heuristique simple, mais la différence reste toujours aux alentours de 10%, ce qui est négligeable par rapport aux gains sur la qualité de service.

Sur la deuxième partie du tableau, on note alors une dégradation très rapide des performances de la RBS, notamment la qualité de service qui explose (+531% sur les instances de taille 80). Bien sûr, il faut prendre ces chiffres avec précaution quand on sait que pour les grandes instances, il y a peu de réussite et donc peu d'éléments de comparaison.

Si on regarde les mêmes chiffres sur ces mêmes tableaux pour les autres heures de soumission, on retrouve exactement la même tendance. La RBS améliore grandement les solutions sur les petites instances (surtout la qualité de service), mais se dégrade très rapidement pour obtenir très peu de solutions et très mauvaises sur les plus grosses instances.

## Heuristique vs NSGAI vs RBS

---

L'algorithme NSGAI est un algorithme génétique qui nous a permis de générer des solutions initiales pour notre problème. Nous l'avons utilisé pour qu'il calcule aussi des solutions finales comparables aux nôtres. Cependant, l'algorithme NSGAI présente des caractéristiques différentes :

- Les requêtes statiques et dynamiques sont toutes traitées de la même manière.
- Les violations de fenêtres de temps sont autorisées car aucun client n'a encore eu de confirmation. Ces violations entraînent une répercussion sur la qualité de service.
- Lors du calcul de la qualité de service, NSGAI utilise comme référence les fenêtres de temps fournies par le client. Nos méthodes utilisent les fenêtres corrigées par NSGAI, et donc plus larges si il y avait eu un dépassement.
- Il n'y a pas de date de soumission ni de tournées en cours, la solution initiale est vide, avec tous les bus au dépôt et toutes les requêtes à insérer.

On voit bien que NSGAI et nos algorithmes n'ont pas les mêmes contraintes ni le même mode opératoire, toute comparaison est donc à prendre avec précaution. Concernant la qualité de service, cette comparaison est même impossible, étant donné que les fenêtres de temps conduisant aux pénalités ne sont même pas identiques. Nous nous contentons donc de comparer les méthodes sur les deux critères restants.

| Taille | Vehicule | Temps          |        |        | Cout   |        |        | Co2            |        |       |
|--------|----------|----------------|--------|--------|--------|--------|--------|----------------|--------|-------|
|        |          | NSGAll         | Insert | %Diff  | NSGAll | Insert | %Diff  | NSGAll         | Insert | %Diff |
| 10     | 2        | 335,34         | 0,08   | -99,98 | 193204 | 278645 | 44,22  | 177034         | 230628 | 30,27 |
| 10     | 4        | 382,33         | 0,06   | -99,98 | 251813 | 285793 | 13,49  | 218511         | 207014 | -5,26 |
| 10     | 6        | 403,81         | 0,06   | -99,99 | 258010 | 308123 | 19,42  | 179796         | 171803 | -4,45 |
| 20     | 2        | 455,06         | 0,12   | -99,97 | 387175 | 356460 | -7,93  | 290539         | 309844 | 6,64  |
| 20     | 4        | 469,43         | 0,11   | -99,98 | 440679 | 472238 | 7,16   | 340965         | 376349 | 10,38 |
| 20     | 6        | 487,67         | 0,11   | -99,98 | 432128 | 442629 | 2,43   | 306820         | 297081 | -3,17 |
| 30     | 2        | 574,08         | 0,20   | -99,97 | 452282 | 549818 | 21,57  | 343055         | 426788 | 24,41 |
| 30     | 4        | 566,56         | 0,16   | -99,97 | 518266 | 558354 | 7,74   | 396003         | 444633 | 12,28 |
| 30     | 6        | 579,46         | 0,15   | -99,97 | 561216 | 551154 | -1,79  | 420860         | 411793 | -2,15 |
| 40     | 2        | 707,88         | 0,30   | -99,96 | 538719 | 609963 | 13,22  | 403224         | 483097 | 19,81 |
| 40     | 4        | 669,61         | 0,22   | -99,97 | 614304 | 692832 | 12,78  | 475384         | 546260 | 14,91 |
| 40     | 6        | 670,28         | 0,21   | -99,97 | 681169 | 681344 | 0,03   | 537127         | 545012 | 1,47  |
| 50     | 2        | 832,24         | 0,42   | -99,95 | 624257 | 696846 | 11,63  | 454110         | 529893 | 16,69 |
| 50     | 4        | 765,43         | 0,29   | -99,96 | 713604 | 759148 | 6,38   | 555191         | 610142 | 9,90  |
| 50     | 6        | 758,03         | 0,25   | -99,97 | 784244 | 779844 | -0,56  | 611053         | 647081 | 5,90  |
| 60     | 2        | 1020,25        | 0,59   | -99,94 | 687698 | 791048 | 15,03  | 507807         | 597894 | 17,74 |
| 60     | 4        | 898,30         | 0,35   | -99,96 | 779577 | 862694 | 10,66  | 606195         | 697122 | 15,00 |
| 60     | 6        | 864,77         | 0,32   | -99,96 | 963231 | 934019 | -3,03  | 763191         | 758973 | -0,55 |
| 70     | 2        | 1140,01        | 0,73   | -99,94 | 674563 | 807255 | 19,67  | 481717         | 614093 | 27,48 |
| 70     | 4        | 998,29         | 0,48   | -99,95 | 790981 | 936318 | 18,37  | 615378         | 749481 | 21,79 |
| 70     | 6        | 933,56         | 0,36   | -99,96 | 939480 | 913782 | -2,74  | 738567         | 758156 | 2,65  |
| 80     | 2        | 1335,19        | 0,81   | -99,94 | 675374 | 764500 | 13,20  | 484876         | 571681 | 17,90 |
| 80     | 4        | 1128,58        | 0,58   | -99,95 | 820516 | 866474 | 5,60   | 633781         | 690406 | 8,93  |
| 80     | 6        | 1033,51        | 0,46   | -99,96 | 917152 | 881285 | -3,91  | 711985         | 726243 | 2,00  |
| 90     | 2        | 1545,66        | 1,19   | -99,92 | 720586 | 770589 | 6,94   | 516457         | 596998 | 15,59 |
| 90     | 4        | 1198,43        | 0,61   | -99,95 | 800706 | 875014 | 9,28   | 626990         | 693239 | 10,57 |
| 90     | 6        | 1114,56        | 0,50   | -99,95 | 955722 | 854531 | -10,59 | 765976         | 732872 | -4,32 |
| 100    | 2        | 1749,61        | 1,08   | -99,94 | 736390 | 845929 | 14,88  | 535605         | 624765 | 16,65 |
| 100    | 4        | 1323,30        | 0,76   | -99,94 | 791795 | 890620 | 12,48  | 616885         | 713350 | 15,64 |
| 100    | 6        | 1193,41        | 0,62   | -99,95 | 870960 | 868757 | -0,25  | 695934         | 726090 | 4,33  |
|        |          | <u>Moyenne</u> |        |        | 8,51   |        |        | <u>Moyenne</u> |        |       |
|        |          |                |        |        |        |        |        | 10,30          |        |       |

Figure 30 : Comparaison NSGAll - Heuristique par insertion

|        |          | Temps |        |        | Cout    |         |        | Co2     |         |        |
|--------|----------|-------|--------|--------|---------|---------|--------|---------|---------|--------|
| Taille | Vehicule | NSGAI | RBS    | %Diff  | NSGAI   | RBS     | %Diff  | NSGAI   | RBS     | %Diff  |
| 10     | 2        | 322   | 2,41   | -99,25 | 177745  | 283797  | 59,67  | 1183020 | 1417530 | 19,82  |
| 10     | 4        | 382   | 6,12   | -98,40 | 254086  | 260439  | 2,50   | 2206340 | 1895510 | -14,09 |
| 10     | 6        | 404   | 7,85   | -98,06 | 258010  | 274527  | 6,40   | 1977750 | 1882800 | -4,80  |
| 20     | 2        | 451   | 6,19   | -98,63 | 366148  | 376082  | 2,71   | 554198  | 575099  | 3,77   |
| 20     | 4        | 469   | 17,60  | -96,25 | 440679  | 381016  | -13,54 | 3750610 | 3188940 | -14,98 |
| 20     | 6        | 488   | 26,57  | -94,55 | 432128  | 378874  | -12,32 | 3375020 | 2875380 | -14,80 |
| 30     | 2        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 30     | 4        | 562   | 18,14  | -96,77 | 521908  | 589166  | 12,89  | 2399920 | 2692660 | 12,20  |
| 30     | 6        | 579   | 40,59  | -92,99 | 561216  | 550031  | -1,99  | 4629460 | 4658150 | 0,62   |
| 40     | 2        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 40     | 4        | 675   | 39,08  | -94,21 | 603018  | 714398  | 18,47  | 3242440 | 3978990 | 22,72  |
| 40     | 6        | 670   | 59,32  | -91,15 | 681169  | 721387  | 5,90   | 5908390 | 6214830 | 5,19   |
| 50     | 2        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 50     | 4        | 770   | 56,30  | -92,69 | 736972  | 835687  | 13,39  | 2315360 | 2680560 | 15,77  |
| 50     | 6        | 759   | 96,06  | -87,34 | 773463  | 837789  | 8,32   | 6028450 | 6656820 | 10,42  |
| 60     | 2        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 60     | 4        | 879   | 101,86 | -88,41 | 777451  | 957612  | 23,17  | 1152620 | 1523690 | 32,19  |
| 60     | 6        | 866   | 160,68 | -81,45 | 969619  | 1022550 | 5,46   | 7729260 | 8128360 | 5,16   |
| 70     | 2        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 70     | 4        | 958   | 72,13  | -92,47 | 792327  | 1003750 | 26,68  | 1192740 | 1593430 | 33,59  |
| 70     | 6        | 935   | 164,24 | -82,43 | 999030  | 1062390 | 6,34   | 2403190 | 2533130 | 5,41   |
| 80     | 2        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 80     | 4        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 80     | 6        | 1024  | 168,10 | -83,59 | 887654  | 1143160 | 28,78  | 2698890 | 3625070 | 34,32  |
| 90     | 2        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 90     | 4        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 90     | 6        | 1133  | 709,42 | -37,39 | 906374  | 1260390 | 39,06  | 1522160 | 2004130 | 31,66  |
| 100    | 2        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 100    | 4        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
| 100    | 6        | -     | -      | -      | -       | -       | -      | -       | -       | -      |
|        |          |       |        |        | Moyenne |         | 12,88  | Moyenne |         | 10,23  |

Figure 31 : Comparaison NSGAII - RBS

Par rapport à NSGAII, les deux méthodes affichent à peu près le même bilan. Une perte de 10% environ de qualité des solutions sur les deux critères que sont le coût et le rejet de CO2.

# Conclusion

---

Les objectifs de ce PFE étaient :

- Définir clairement le problème (données, contraintes, critères)
- Définir et programmer des heuristiques simples (algorithme de liste, gloutons...)
- Définir et programmer une ou plusieurs méthodes arborescentes tronquées
- Phase de tests.

Nous avons donc complété l'ensemble des objectifs de ce PFE.

Malheureusement, les résultats des tests nous ont montré que nous avons surement fait des erreurs dans la définition de nos algorithmes, notamment pour la méthode arborescente. Cependant, nous avons passé beaucoup de temps sur la programmation de ces algorithmes. Maintenant qu'ils fonctionnent parfaitement et que nous avons des résultats, il serait dommage de ne pas poursuivre dans ce sens pour tenter de les améliorer. En effet, je ne pense pas que la méthode RBS soit mauvaise en elle-même pour résoudre ce problème. On voit notamment que la complexité et les temps de calcul sont bons. Nous avons fait beaucoup de choix arbitraires pour mettre en place cette méthode. Avec l'aide des résultats expérimentaux, nous pouvons envisager de remettre en cause ces choix et de les remplacer par d'autres. Il existe de nombreux points sur lesquels nous pouvons jouer :

- Ordonnancement des sommets restants à insérer
- Schéma de branchement
- Calcul de la borne supérieure
- Sélection des meilleures solutions
- Opérateur de voisinage
- Sélection du meilleur voisin

Concernant l'heuristique simple, on peut aussi envisager de l'améliorer en autorisant l'ouverture de nouvelles tournées par exemple si on juge que la qualité de service est trop dégradée. Nous n'avons pas pris cette initiative car pour cela il faut répondre à la question : A partir de quelle valeur une tournée a-t-elle une qualité de service trop mauvaise ?

# Bibliographie

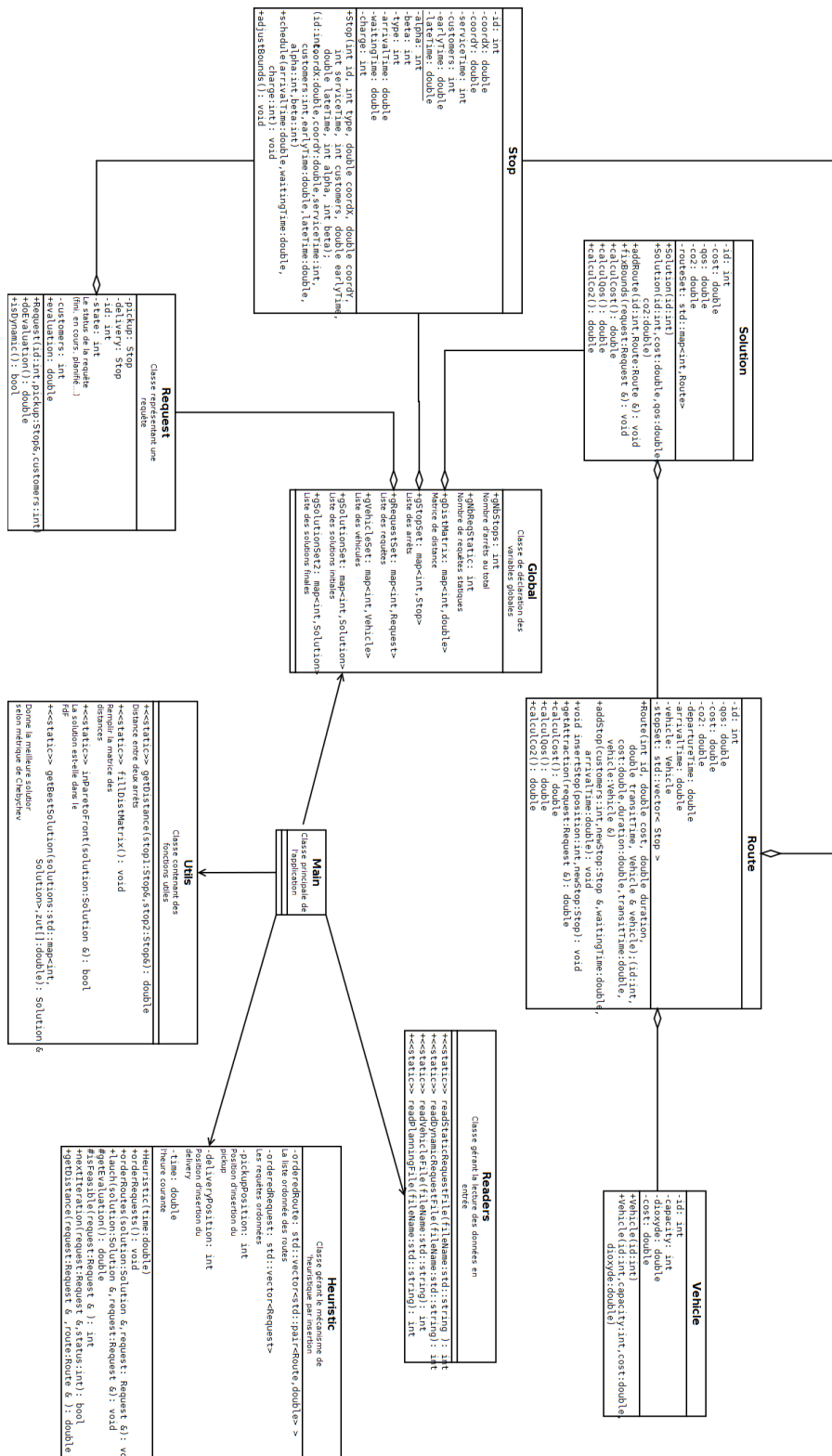
---

- [1] B.Hunsaker et M.Savelsbergh, Efficient feasibility testing for dial-a-ride problems, Operations Research Letters 30 (2002) 169 -173
- [2] Oli B.G Madsen, Hans F. Ravn et Jens Moberg Rygaard, A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities and multiple objectives, Annals of Operations Research 60(1995) 193-208
- [3] Murat Firat, Gerhard J. Woeginger, Analysis of the dial-a-ride problem of Hunsaker and Savelsberg, Operations Research Letters 39 (2011) 32-35
- [4] Jiufa Tang, Yuan Kong, Henry Lau, Andrew W.H. Ip, A note on "Efficient feasibility testing for dial-a-ride problems", Operations Research Letters 38 (2010) 405-407
- [5] Certu, Les outils à la mobilité fiche n°2 « Le transport à la demande en zone urbaine » - Avril 2009
- [6] M. Schilde, K.F. Doerner, R.F. Hart, Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transport, Computers & Operations Research 38 (2011) 1719-1730
- [7] Sophie N. Parragh, Introducing heterogeneous users and vehicles into models and algorithms for a dial-a-ride problem, Transportation Research Part C 19 (2011) 912-930
- [8] Julie Paquette, J-F Cordeau, Gilbert Laporte et Marta M. B. Pascoal, Combining Multicriteria Analysis and Tabu Search for Dial-a-Ride Problems, 3 août 2011
- [9] Thierry Garaix, Christian Artigues, Dominique Feillet, Didier Josselin, Vehicle routing problems with alternative paths : An application to on-demand transportation, European Journal of Operational Research 204 (2010) 62-75
- [10] Ravindra K. Ahuja, Robert Dial, Fast Infeasibility Detection Algorithms for Dial-A-Ride Transit Problems, SWP# 4005 février 1998
- [11] Thierry Garaix, Etude et résolution du problème de transport à la demande avec qualité de service, thèse à l'université d'avignon (2007)
- [12] Jesper Bjerring Hansen, The dial-a-ride problem and Exploiting knowledge about future customer requests, master's thesis in mathematics-economics, department of mathematical sciences, Aarhus university (2010)
- [13] Bureau d'études en transports et déplacements, Services à la demande et transports innovants en milieu rural : de l'inventaire à la valorisation des expériences, Rapport final, Novembre 2004

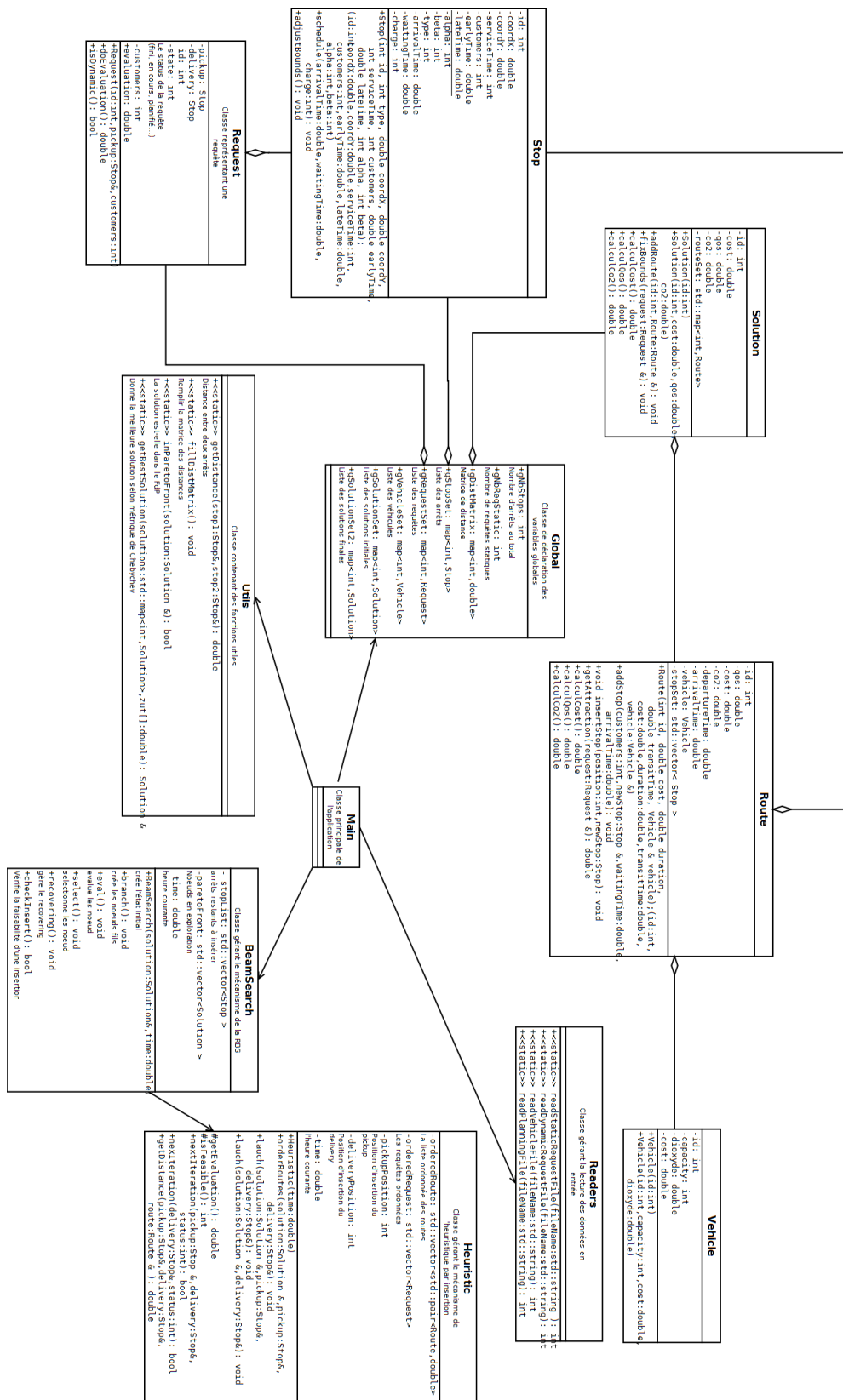
# Annexes

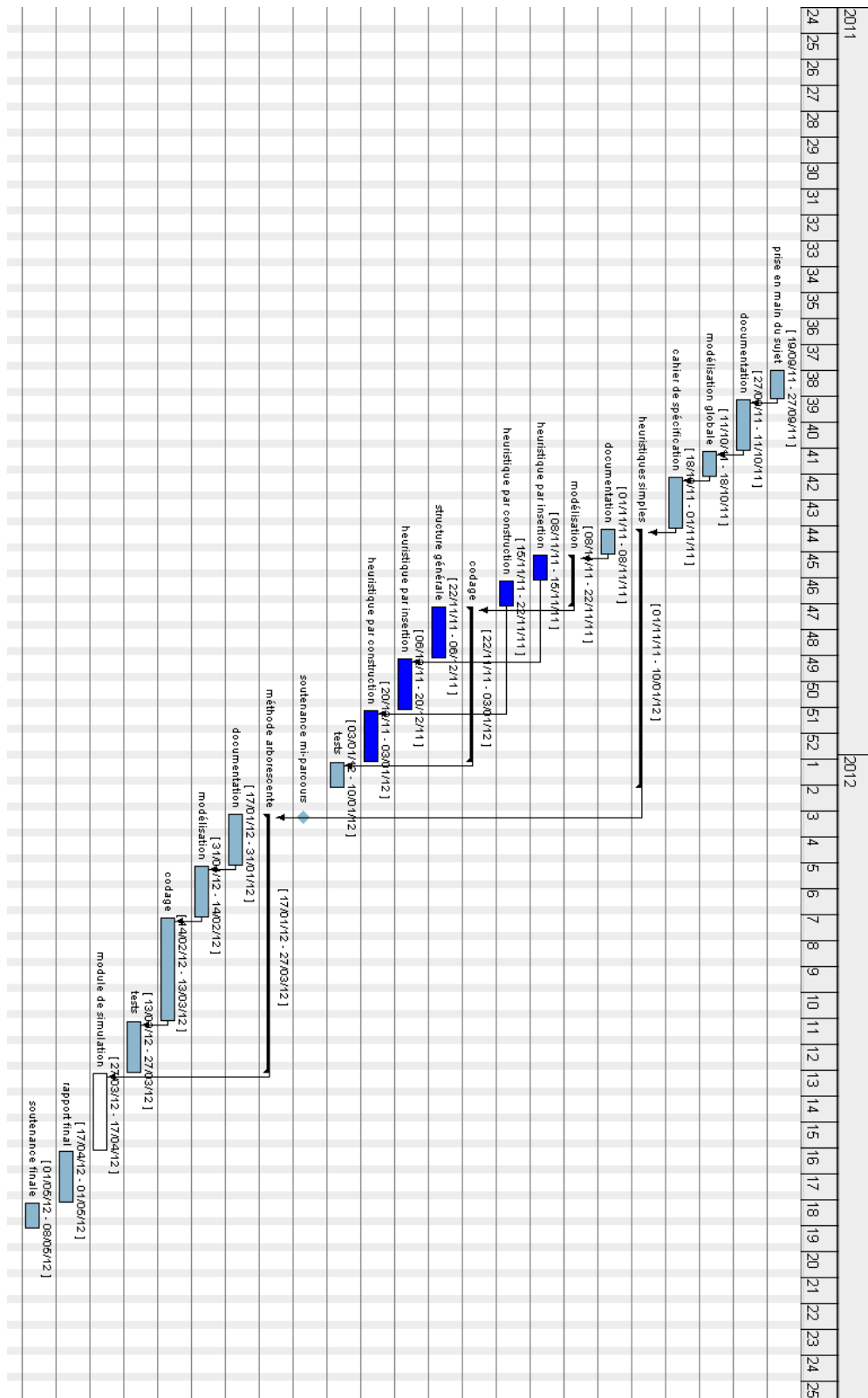
---

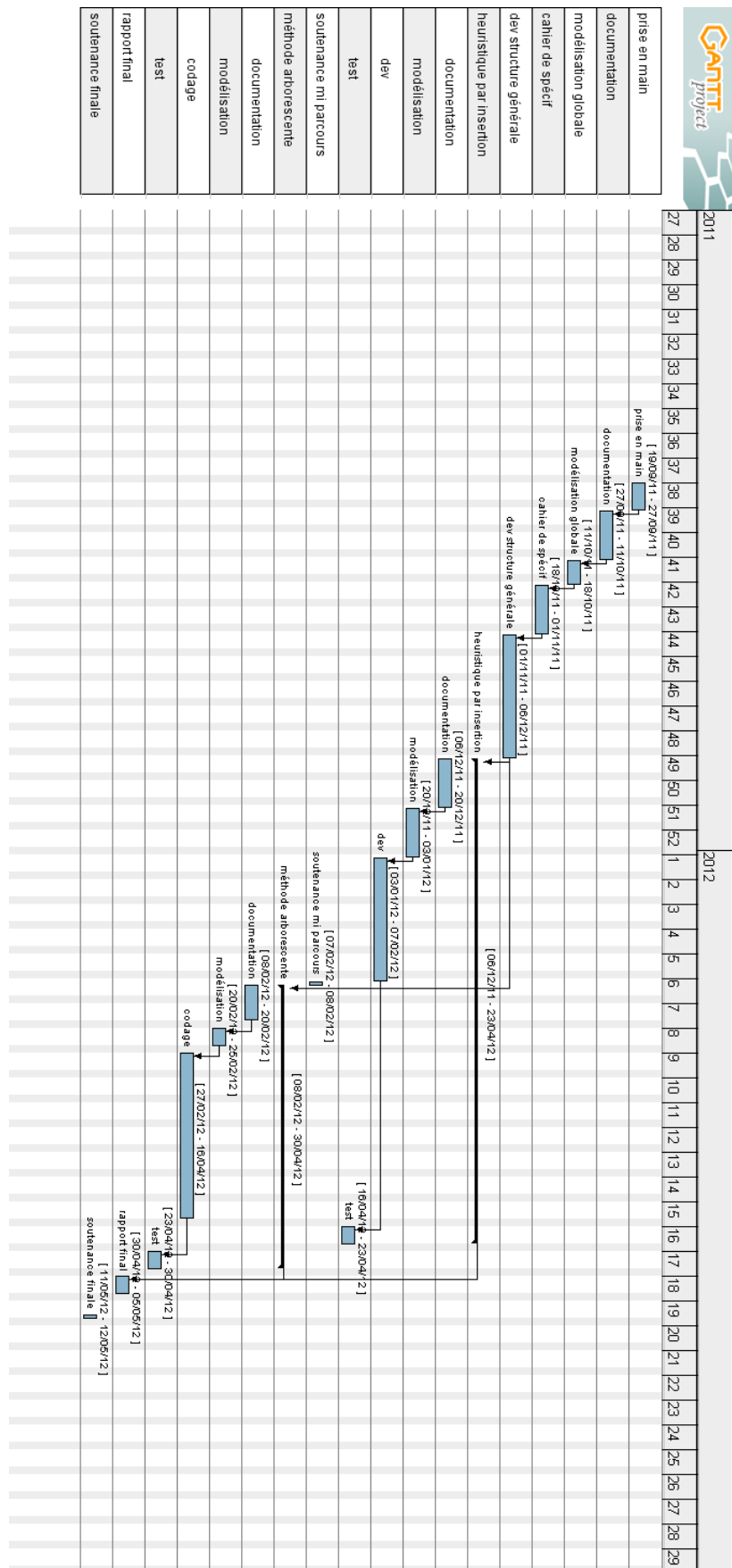
## Diagramme de classe de l'heuristique par insertion



## Diagramme de classe de la RBS







| vehicle_4.txt |            |    |          |     |              |                 |
|---------------|------------|----|----------|-----|--------------|-----------------|
| 1             | ID_Vehicle |    | capacity |     | Cost_Vehicle | Dioxyde_emitted |
| 2             |            |    |          |     |              |                 |
| 3             | 1          | 27 | 400      | 420 | 0            |                 |
| 4             | 2          | 15 | 500      | 400 | 0            |                 |
| 5             | 3          | 9  | 350      | 300 | 0            |                 |
| 6             | 4          | 4  | 400      | 200 | 0            |                 |

| vehicle_4.txt |                      | Instance_1_Stat_R_10.txt |                   |                       |                   |                  |                  |                       |       |                   |  |          |        |        |       |        |
|---------------|----------------------|--------------------------|-------------------|-----------------------|-------------------|------------------|------------------|-----------------------|-------|-------------------|--|----------|--------|--------|-------|--------|
| 1             | Nombre de requetes : |                          |                   | 10                    | coordX            |                  | coordY           | temps de service(sec) |       | Nbre de personnes |  | earliest | latest | alphax | betax | gammax |
| 2             | requete N°           | coordX                   | coordY            | temps de service(sec) | Nbre de personnes | earliest         | latest           | alphax                | betax | gammax            |  |          |        |        |       |        |
| 3             | 0                    | 0                        | 0                 | 0                     | 1440              | 0                | 0                | 0                     |       |                   |  |          |        |        |       |        |
| 4             | 1                    | 4,12932529306473         | 3,64274039102846  | 10                    | 3                 | 265              | 278              | 1000                  | 10000 | 0                 |  |          |        |        |       |        |
| 5             | 2                    | 2,8554524028932          | -5,17673156465252 | 10                    | 4                 | 26               | 44               | 100                   | 10000 | 0                 |  |          |        |        |       |        |
| 6             | 3                    | 16,4010978845885         | -10,3734862387057 | 10                    | 3                 | 898              | 908              | 100                   | 100   | 0                 |  |          |        |        |       |        |
| 7             | 4                    | 15,822553539566          | 10,0946370466122  | 10                    | 3                 | 495              | 505              | 1                     | 10000 | 0                 |  |          |        |        |       |        |
| 8             | 5                    | 12,4361421691422         | -8,60617428487454 | 10                    | 1                 | 27               | 49               | 10000                 | 10000 | 0                 |  |          |        |        |       |        |
| 9             | 6                    | 18,7020439648545         | -19,052274226701  | 10                    | 3                 | 644              | 667              | 10000                 | 100   | 0                 |  |          |        |        |       |        |
| 10            | 7                    | 14,672080499433          | 14,4340310406098  | 10                    | 1                 | 122              | 135              | 100                   | 10000 | 0                 |  |          |        |        |       |        |
| 11            | 8                    | -2,41592520029094        | 9,4276231105568   | 10                    | 3                 | 97               | 124              | 100                   | 1000  | 0                 |  |          |        |        |       |        |
| 12            | 9                    | 13,7654090829964         | 16,928691098899   | 10                    | 3                 | 434              | 459              | 100                   | 1000  | 0                 |  |          |        |        |       |        |
| 13            | 10                   | -12,4110957758553        | -17,688079270389  | 10                    | 3                 | 833              | 850              | 30000                 | 10000 | 0                 |  |          |        |        |       |        |
| 14            | 11                   | 18,3193175114316         | 17,5024675566249  | 10                    | -3                | 294,753282366651 | 307,753282366651 | 10                    | 100   | 0                 |  |          |        |        |       |        |
| 15            | 12                   | 3,23601787129232         | 9,22936681833062  | 10                    | -4                | 47,6166889984392 | 65,6166889984392 | 1                     | 1000  | 0                 |  |          |        |        |       |        |
| 16            | 13                   | -7,14640017000325        | -13,8384180301048 | 10                    | -3                | 933,701588734083 | 943,701588734083 | 10000                 | 1000  | 0                 |  |          |        |        |       |        |
| 17            | 14                   | -1,95519114935547        | 17,6954752941129  | 10                    | -3                | 524,00167466976  | 534,00167466976  | 30000                 | 30000 | 0                 |  |          |        |        |       |        |
| 18            | 15                   | -14,1022893758967        | -12,2030990161947 | 10                    | -1                | 67,1716191718155 | 89,1716191718155 | 10                    | 100   | 0                 |  |          |        |        |       |        |
| 19            | 16                   | -2,52627928858915        | 0,244957683722006 | 10                    | -3                | 687,03261497985  | 710,03261497985  | 100                   | 1000  | 0                 |  |          |        |        |       |        |
| 20            | 17                   | 15,7347484099375         | 10,6537874371995  | 10                    | -1                | 127,8901515919   | 140,8901515919   | 1000                  | 1000  | 0                 |  |          |        |        |       |        |
| 21            | 18                   | 17,0527408444568         | -8,68225508773804 | 10                    | -3                | 136,884112793413 | 163,884112793413 | 30000                 | 1000  | 0                 |  |          |        |        |       |        |
| 22            | 19                   | 13,0512358215876         | -17,9946203520496 | 10                    | -3                | 486,395919534171 | 511,395919534171 | 30000                 | 10000 | 0                 |  |          |        |        |       |        |
| 23            | 20                   | -18,5010104060643        | -16,7824306324043 | 10                    | -3                | 842,235330759216 | 859,235330759216 | 1                     | 100   | 0                 |  |          |        |        |       |        |
| 24            |                      |                          |                   |                       |                   |                  |                  |                       |       |                   |  |          |        |        |       |        |

# Fichier de données « requêtes dynamiques »

| Instance_1_R_30_Dyn_10tx |                   |                    |                       |                   |                  |                  |       |       |          |    |    |    |    |    |    |
|--------------------------|-------------------|--------------------|-----------------------|-------------------|------------------|------------------|-------|-------|----------|----|----|----|----|----|----|
| 1                        | 2                 | 3                  | 4                     | 5                 | 6                | 7                | 8     | 9     | 10       | 11 | 12 | 13 | 14 | 15 | 16 |
| requete N°               | cooridx           | coorxy             | temps de service(sec) | Nbre de personnes | earliest         | latest           | alpha | beta  | callDate |    |    |    |    |    |    |
| 1                        | 10,4398404389805  | -18,3994291342792  | 10                    | 1                 | 124,268179595164 | 149,268179595164 | 1000  | 30000 | 94       | 0  |    |    |    |    |    |
| 2                        | 15,5652570517572  | 17,3267251985738   | 10                    | -1                | 178,40608680865  | 203,40608680865  | 30000 | 1000  | 94       | 0  |    |    |    |    |    |
| 3                        | 18,80318226237    | 14,1076084943989   | 10                    | 2                 | 218,536359190328 | 228,536359190328 | 10000 | 30000 | 188      | 0  |    |    |    |    |    |
| 4                        | 5,54012938660576  | -2,08900214434555  | 10                    | -2                | 249,949195737792 | 259,949195737792 | 1000  | 1000  | 188      | 0  |    |    |    |    |    |
| 5                        | -16,5220497532385 | 7,53193908721764   | 10                    | 2                 | 312,804538785492 | 325,804538785492 | 10    | 100   | 282      | 0  |    |    |    |    |    |
| 6                        | 1,92947206177259  | 7,76036833774316   | 10                    | -2                | 340,483942386866 | 353,483942386866 | 10000 | 10000 | 282      | 0  |    |    |    |    |    |
| 7                        | 2,7326364226214   | -12,9897032645483  | 10                    | 3                 | 407,072718380656 | 426,072718380656 | 1     | 100   | 377      | 0  |    |    |    |    |    |
| 8                        | 1,31915583336687  | 12,7178643423681   | 10                    | -3                | 445,692313685449 | 464,692313685449 | 10    | 1000  | 377      | 0  |    |    |    |    |    |
| 9                        | 16,3396885042729  | -14,2799720886536  | 10                    | 3                 | 501,34089797582  | 523,34089797582  | 100   | 1000  | 471      | 0  |    |    |    |    |    |
| 10                       | 17,7412531886908  | -4,76382230630323  | 10                    | -3                | 515,769111761686 | 537,769111761686 | 10    | 100   | 471      | 0  |    |    |    |    |    |
| 11                       | 12,6947096514957  | -0,793478852807416 | 10                    | 3                 | 595,609077570984 | 606,609077570984 | 10000 | 1000  | 565      | 0  |    |    |    |    |    |
| 12                       | -3,00883208541611 | 16,4616463130627   | 10                    | -3                | 630,605738511929 | 641,605738511929 | 1     | 10000 | 565      | 0  |    |    |    |    |    |
| 13                       | -2,32425234388758 | 5,30463353531303   | 10                    | 3                 | 689,877257166149 | 717,877257166149 | 1     | 100   | 659      | 0  |    |    |    |    |    |
| 14                       | -5,17282604480759 | -11,876050397696   | 10                    | -3                | 709,882258985783 | 737,882258985783 | 100   | 30000 | 659      | 0  |    |    |    |    |    |
| 15                       | -4,3662650624133  | 7,43127986203473   | 10                    | 3                 | 784,145436761313 | 805,145436761313 | 10    | 100   | 754      | 0  |    |    |    |    |    |
| 16                       | 4,87030488665695  | -19,5915180070286  | 10                    | -3                | 816,249895313208 | 837,249895313208 | 30000 | 30000 | 754      | 0  |    |    |    |    |    |
| 17                       | -17,6582671318475 | -17,4408026027683  | 10                    | -3                | 899,413616356477 | 899,413616356477 | 10000 | 30000 | 848      | 0  |    |    |    |    |    |
| 18                       | 10,555693763567   | 0,572025282574828  | 10                    | 3                 | 920,857339377151 | 941,857339377151 | 30000 | 10000 | 848      | 0  |    |    |    |    |    |
| 19                       | 3,30074233156664  | -14,6815501873761  | 10                    | -3                | 972,681795951641 | 997,681795951641 | 100   | 30000 | 942      | 0  |    |    |    |    |    |
| 20                       | 17,0060449824697  |                    | 10                    | -3                | 1003,44119276112 | 1028,44119276112 | 1     | 1000  | 942      | 0  |    |    |    |    |    |
| 21                       |                   |                    |                       |                   |                  |                  |       |       |          |    |    |    |    |    |    |
| 22                       |                   |                    |                       |                   |                  |                  |       |       |          |    |    |    |    |    |    |
| 23                       |                   |                    |                       |                   |                  |                  |       |       |          |    |    |    |    |    |    |

## Fichier de données « solutions »

| Routes Instance_1 Stat_R_10M |                                                                                                                                               |
|------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 1                            | solution:0[ cost: 128878; QoS: 0; Co2: 105647                                                                                                 |
| 2                            | route ID:1 ( C: 71.7287129 QoS: 0.000000 d: 554.8301505 T: 42.5011479 ) : [ 0(A: 287.5230952 W: 0.000000 q: 0)>1(A: 299.000000 W: 0.000000    |
| 3                            | route ID:3 ( C: 171.6381002 QoS: 0.000000 d: 771.2973117 T: 84.9002355 ) : [ 0(A: 298.5753141 W: 0.000000 q: 0)>8(A: 303.6587679 W: 0.000000  |
| 4                            | route ID:2 ( C: 121.7929542 QoS: 0.000000 d: 908.8816504 T: 67.5659700 ) : [ 0(A: 40.4957921 W: 0.000000 q: 0)>7(A: 46.0000000 W: 0.000000    |
| 5                            | solution:1[ cost: 106236.8774367; QoS: 18080111.9792000; Co2: 53118.4387183                                                                   |
| 6                            | route ID:1 ( C: 265.5921936 QoS: 18080111.9792000 d: 465.5921936 T: 320.5700554 ) : [ 0(A: 983.8157414 W: 0.000000 q: 0)>3(A: 993.0000000 W:  |
| 7                            | solution:2[ cost: 42770.1338240; QoS: 17902348.9124800; Co2: 74847.7341919                                                                    |
| 8                            | route ID:1 ( C: 213.8506691 QoS: 17902348.9124800 d: 1069.7298615 T: 1410.4251375 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000   |
| 9                            | solution:3[ cost: 143026.6466582; QoS: 344712.9789400; Co2: 71513.3233291                                                                     |
| 10                           | route ID:1 ( C: 357.5666166 QoS: 344712.9789400 d: 1088.2390611 T: 678.2593091 ) : [ 0(A: 69.4957921 W: 0.000000 q: 0)>7(A: 75.0000000        |
| 11                           | solution:4[ cost: 42813.8560542; QoS: 16456500.2171800; Co2: 74924.2480949                                                                    |
| 12                           | route ID:1 ( C: 214.0692803 QoS: 16456500.2171800 d: 1069.9484726 T: 1531.3499955 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000   |
| 13                           | solution:5[ cost: 45054.1916250; QoS: 9722439.0768300; Co2: 78844.8353438                                                                     |
| 14                           | route ID:1 ( C: 225.2709581 QoS: 9722439.0768300 d: 861.6473652 T: 1147.2619861 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000 W:  |
| 15                           | solution:6[ cost: 54685.3911733; QoS: 11954127.9099800; Co2: 78626.2325623                                                                    |
| 16                           | route ID:1 ( C: 34.1464040 QoS: 0.000000 d: 54.1464040 T: 11.9540214 ) : [ 0(A: 895.2310386 W: 0.000000 q: 0)>2(A: 901.0000000 W: 0.000000    |
| 17                           | route ID:3 ( C: 205.1341479 QoS: 11954127.9099800 d: 1041.0133403 T: 1438.7604362 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000   |
| 18                           | solution:7[ cost: 42865.9145426; QoS: 15304212.0851800; Co2: 75015.3504496                                                                    |
| 19                           | route ID:1 ( C: 214.3295727 QoS: 15304212.0851800 d: 1070.2087651 T: 1509.4863532 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000   |
| 20                           | solution:8[ cost: 44578.9033348; QoS: 13360998.6920800; Co2: 78013.0808358                                                                    |
| 21                           | route ID:1 ( C: 222.8945167 QoS: 13360998.6920800 d: 959.2338106 T: 1319.6485953 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000 W: |
| 22                           | solution:9[ cost: 44405.8717879; QoS: 14463813.7773800; Co2: 77710.2756289                                                                    |
| 23                           | route ID:1 ( C: 222.0293589 QoS: 14463813.7773800 d: 1077.9085513 T: 1393.2008884 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000   |
| 24                           | solution:10[ cost: 89830.5122011; QoS: 13003438.3818800; Co2: 60428.8512350                                                                   |
| 25                           | route ID:1 ( C: 193.5490902 QoS: 12663371.2342000 d: 353.5490902 T: 246.7779795 ) : [ 0(A: 983.8157414 W: 0.000000 q: 0)>3(A: 993.0000000 W:  |
| 26                           | route ID:3 ( C: 62.0543805 QoS: 340067.1476800 d: 424.0882233 T: 406.0363141 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000 W:     |
| 27                           | solution:11[ cost: 54618.0714789; QoS: 11976222.2648800; Co2: 78508.4230973                                                                   |
| 28                           | route ID:1 ( C: 34.1464040 QoS: 0.000000 d: 54.1464040 T: 11.9540214 ) : [ 0(A: 895.2310386 W: 0.000000 q: 0)>2(A: 901.0000000 W: 0.000000    |
| 29                           | route ID:3 ( C: 204.7975494 QoS: 11976222.2648800 d: 1040.6767418 T: 1447.5192736 ) : [ 0(A: 317.9165462 W: 0.000000 q: 0)>8(A: 323.0000000   |

## Résultats heuristique par insertion pour T=0

| Req. Stat. | Req. Dyn. | Vehicule | Temps  | Front | Choix | Cout Min. | Cout Moy. | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max. |
|------------|-----------|----------|--------|-------|-------|-----------|-----------|--------------|--------------|--------------|--------------|----------|----------|----------|
| 10         | 10        | 2        | 0.0797 | 2.8   | 1     | 184419    | 288272    | 349747       | 208261       | 1.15671e+007 | 1.30662e+008 | 92209.4  | 237406   | 283119   |
| 10         | 10        | 4        | 0.0637 | 3.9   | 1.1   | 218083    | 285086    | 382330       | 6183.18      | 4.45081e+006 | 1.89902e+007 | 152563   | 205470   | 270801   |
| 10         | 10        | 6        | 0.0574 | 2.9   | 1     | 268652    | 306967    | 331165       | 6183.18      | 4.12106e+006 | 2.78431e+007 | 134326   | 177894   | 224311   |
| 20         | 10        | 2        | 0.1161 | 1.5   | 1     | 177993    | 346650    | 496803       | 1994.39      | 3.58911e+007 | 1.10339e+008 | 88996.6  | 289228   | 399344   |
| 20         | 10        | 4        | 0.1108 | 4.7   | 1.9   | 405769    | 472317    | 532855       | 42510.4      | 5.9276e+006  | 1.83727e+007 | 327991   | 374803   | 454048   |
| 20         | 10        | 6        | 0.1062 | 4.6   | 2     | 323477    | 435654    | 494015       | 82016.1      | 4.74093e+006 | 3.01409e+007 | 241358   | 297378   | 370936   |
| 30         | 10        | 2        | 0.1963 | 1.9   | 1     | 468581    | 550428    | 602494       | 1.5778e+007  | 7.16016e+007 | 1.10059e+008 | 355512   | 425261   | 467343   |
| 30         | 10        | 4        | 0.1554 | 4.8   | 2.1   | 419025    | 557192    | 629496       | 236765       | 5.68398e+006 | 2.97002e+007 | 366708   | 441439   | 482287   |
| 30         | 10        | 6        | 0.1474 | 4.7   | 2.5   | 453581    | 544866    | 585711       | 95922.4      | 3.70103e+006 | 2.22208e+007 | 333326   | 411164   | 487186   |
| 40         | 10        | 2        | 0.298  | 1.7   | 1     | 424539    | 621245    | 686919       | 2.77145e+007 | 9.83368e+007 | 2.30073e+008 | 445766   | 486038   | 537328   |
| 40         | 10        | 4        | 0.2182 | 5.4   | 2.2   | 573637    | 689681    | 758626       | 1.15463e+006 | 6.79276e+006 | 2.54394e+007 | 450048   | 543324   | 602910   |
| 40         | 10        | 6        | 0.2081 | 5.9   | 3     | 583488    | 682756    | 776736       | 225171       | 5.88802e+006 | 3.24629e+007 | 426911   | 549420   | 658731   |
| 50         | 10        | 2        | 0.4153 | 2.3   | 1     | 653762    | 695432    | 751585       | 8.5403e+007  | 1.31221e+008 | 2.20292e+008 | 501121   | 528778   | 563195   |
| 50         | 10        | 4        | 0.2855 | 4.6   | 2.1   | 517962    | 761184    | 893789       | 482020       | 1.04851e+007 | 3.43561e+007 | 440482   | 609374   | 718251   |
| 50         | 10        | 6        | 0.2503 | 4.8   | 3.1   | 512571    | 764866    | 934144       | 343037       | 6.21335e+006 | 4.21448e+007 | 496370   | 637375   | 729318   |
| 60         | 10        | 2        | 0.5868 | 2.6   | 1     | 738952    | 782534    | 859767       | 7.63982e+007 | 1.623e+008   | 2.28054e+008 | 558700   | 593450   | 650391   |
| 60         | 10        | 4        | 0.3513 | 2.9   | 2.1   | 686419    | 842189    | 1.0545e+006  | 1.84996e+006 | 1.77238e+007 | 5.53359e+007 | 543186   | 677674   | 839569   |
| 60         | 10        | 6        | 0.3219 | 3.4   | 2.8   | 840333    | 924354    | 1.00399e+006 | 249431       | 5.46137e+006 | 3.00516e+007 | 655170   | 748694   | 898835   |
| 70         | 10        | 2        | 0.7286 | 2.4   | 1     | 759781    | 808298    | 878112       | 8.83383e+007 | 2.02099e+008 | 3.50542e+008 | 553970   | 617932   | 753214   |
| 70         | 10        | 4        | 0.4778 | 3.8   | 2.2   | 866660    | 925298    | 1.0567e+006  | 2.49352e+006 | 1.81802e+007 | 6.06177e+007 | 680549   | 741833   | 870685   |
| 70         | 10        | 6        | 0.3643 | 5.4   | 3.3   | 717809    | 922335    | 1.02304e+006 | 20694.9      | 7.10741e+006 | 2.8669e+007  | 668321   | 744204   | 829522   |
| 80         | 10        | 2        | 0.8121 | 1.6   | 1     | 709330    | 761472    | 860112       | 1.25753e+008 | 2.13547e+008 | 2.86492e+008 | 524047   | 570879   | 651670   |
| 80         | 10        | 4        | 0.578  | 2.5   | 2.2   | 760158    | 863382    | 1.12554e+006 | 1.90391e+006 | 1.79316e+007 | 4.74759e+007 | 611421   | 691520   | 896368   |
| 80         | 10        | 6        | 0.4622 | 3.5   | 3     | 717811    | 878367    | 998273       | 453116       | 3.93168e+006 | 1.48667e+007 | 620355   | 725018   | 806173   |
| 90         | 10        | 2        | 1.1942 | 1.4   | 1     | 589748    | 778136    | 859127       | 1.44263e+008 | 2.57617e+008 | 3.4265e+008  | 552688   | 597395   | 639326   |
| 90         | 10        | 4        | 0.612  | 3.3   | 1.9   | 692460    | 910409    | 1.06875e+006 | 1.47045e+007 | 3.85523e+007 | 7.87517e+007 | 571829   | 717696   | 819837   |
| 90         | 10        | 6        | 0.5048 | 4.1   | 2.5   | 645984    | 857184    | 1.1254e+006  | 724559       | 7.99083e+006 | 2.04144e+007 | 617222   | 729840   | 892803   |
| 100        | 10        | 2        | 1.0826 | 2.2   | 1     | 725065    | 840900    | 995059       | 1.62333e+008 | 2.85325e+008 | 4.14357e+008 | 530013   | 620221   | 760979   |
| 100        | 10        | 4        | 0.7647 | 2.2   | 2.1   | 672337    | 888536    | 1.06322e+006 | 1.67672e+006 | 4.15467e+007 | 1.6452e+008  | 603210   | 707267   | 833917   |
| 100        | 10        | 6        | 0.6241 | 3.5   | 2.8   | 553284    | 900935    | 1.0454e+006  | 330054       | 5.73698e+006 | 2.84752e+007 | 460076   | 749615   | 851078   |

## Résultats heuristique par insertion pour T=200

| Req. Stat. | Req. Dyn. | Vehicule | Temps     | Front   | Choix | Cout Min. | Cout Moy. | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max. |
|------------|-----------|----------|-----------|---------|-------|-----------|-----------|--------------|--------------|--------------|--------------|----------|----------|----------|
| 10         | 9         | 2        | 0.0427    | 2.2     | 1     | 155657    | 289616    | 362552       | 201080       | 1.23872e+007 | 8.83048e+007 | 87503    | 232513   | 316591   |
| 10         | 9         | 4        | 0.0482    | 2.7     | 1.1   | 209264    | 282583    | 392798       | 13600        | 5.89977e+006 | 3.18858e+007 | 140835   | 201683   | 281466   |
| 10         | 9         | 6        | 0.0446    | 2.4     | 1.2   | 266277    | 294675    | 330317       | 6183.18      | 8.0099e+006  | 2.57119e+007 | 133138   | 162405   | 232065   |
| 20         | 9         | 2        | 0.0884444 | 1.33333 | 1     | 195572    | 373995    | 457163       | 6.53832e+006 | 4.47798e+007 | 8.84622e+007 | 97786.1  | 321730   | 387148   |
| 20         | 9         | 4        | 0.0901    | 4.1     | 1.8   | 411830    | 459907    | 508285       | 18419.5      | 7.7586e+006  | 2.13483e+007 | 316140   | 360933   | 432171   |
| 20         | 9         | 6        | 0.0864    | 4.3     | 2.2   | 322215    | 441638    | 531016       | 580568       | 7.2498e+006  | 3.0693e+007  | 241886   | 291528   | 351352   |
| 30         | 9         | 2        | 0.1538    | 1.7     | 1     | 447210    | 532612    | 581962       | 1.57377e+007 | 6.92421e+007 | 1.36605e+008 | 338439   | 411684   | 457077   |
| 30         | 9         | 4        | 0.1361    | 4.5     | 2.3   | 429408    | 545522    | 619965       | 407499       | 6.80874e+006 | 3.07052e+007 | 378320   | 430406   | 474596   |
| 30         | 9         | 6        | 0.1189    | 5.1     | 2.6   | 442986    | 524934    | 572111       | 282009       | 5.26292e+006 | 2.48123e+007 | 324245   | 397996   | 460294   |
| 40         | 9         | 2        | 0.2386    | 1.9     | 1     | 394021    | 606644    | 664783       | 2.24971e+007 | 9.00173e+007 | 1.74115e+008 | 413722   | 475022   | 522901   |
| 40         | 9         | 4        | 0.1864    | 4.8     | 2.4   | 552940    | 665909    | 727132       | 1.17216e+006 | 8.45964e+006 | 2.56619e+007 | 421951   | 521922   | 568166   |
| 40         | 9         | 6        | 0.1698    | 5       | 3.3   | 580474    | 661681    | 748679       | 214638       | 5.66929e+006 | 3.24568e+007 | 421705   | 525766   | 622510   |
| 50         | 9         | 2        | 0.3443    | 2.5     | 1     | 640352    | 683416    | 750792       | 6.81891e+007 | 1.0564e+008  | 1.64176e+008 | 490342   | 515497   | 553906   |
| 50         | 9         | 4        | 0.2341    | 3.6     | 2.4   | 514417    | 728986    | 870690       | 748318       | 8.82411e+006 | 2.5708e+007  | 436759   | 584432   | 698647   |
| 50         | 9         | 6        | 0.2272    | 4       | 2.9   | 495821    | 744202    | 918427       | 252505       | 4.03845e+006 | 1.8485e+007  | 495901   | 625905   | 713880   |
| 60         | 9         | 2        | 0.4976    | 2.5     | 1     | 715854    | 765751    | 831337       | 7.49344e+007 | 1.34793e+008 | 2.28054e+008 | 534778   | 579214   | 629462   |
| 60         | 9         | 4        | 0.2933    | 2.8     | 2.2   | 661831    | 832933    | 1.05371e+006 | 1.99996e+006 | 1.74307e+007 | 5.53288e+007 | 540345   | 676051   | 839172   |
| 60         | 9         | 6        | 0.2737    | 3.5     | 3     | 826636    | 916062    | 983403       | 249429       | 4.94337e+006 | 2.07213e+007 | 654179   | 737447   | 922402   |
| 70         | 9         | 2        | 0.627     | 2.6     | 1     | 744447    | 791524    | 876273       | 8.62896e+007 | 1.75509e+008 | 2.99516e+008 | 536950   | 602151   | 751284   |
| 70         | 9         | 4        | 0.4064    | 3       | 2.1   | 841419    | 917499    | 1.05777e+006 | 1.58836e+006 | 2.15358e+007 | 6.05402e+007 | 658642   | 732916   | 844546   |
| 70         | 9         | 6        | 0.318     | 4.4     | 3.3   | 701964    | 893736    | 999338       | 20694.9      | 7.0086e+006  | 2.50464e+007 | 663112   | 723712   | 813957   |
| 80         | 9         | 2        | 0.7257    | 1.8     | 1     | 687460    | 749207    | 842106       | 1.09459e+008 | 1.78984e+008 | 2.56193e+008 | 509063   | 561436   | 642958   |
| 80         | 9         | 4        | 0.4896    | 3       | 2.3   | 721646    | 854803    | 1.10275e+006 | 1.90391e+006 | 1.80346e+007 | 4.74759e+007 | 585149   | 681768   | 872433   |
| 80         | 9         | 6        | 0.408     | 3.9     | 2.8   | 699698    | 853171    | 991818       | 157172       | 4.2568e+006  | 1.48656e+007 | 596925   | 701165   | 784443   |
| 90         | 9         | 2        | 1.0275    | 1.6     | 1     | 584470    | 748668    | 847422       | 1.41279e+008 | 2.30142e+008 | 2.8592e+008  | 530833   | 585107   | 630547   |
| 90         | 9         | 4        | 0.525     | 3.3     | 2     | 685268    | 894837    | 1.06018e+006 | 1.07982e+007 | 3.61826e+007 | 7.32927e+007 | 558254   | 704874   | 815554   |
| 90         | 9         | 6        | 0.4448    | 4.1     | 2.6   | 614453    | 835974    | 1.10033e+006 | 681125       | 8.20311e+006 | 2.04144e+007 | 578036   | 709815   | 875322   |
| 100        | 9         | 2        | 0.9667    | 2.1     | 1     | 706141    | 831748    | 974385       | 1.58131e+008 | 2.40083e+008 | 4.12461e+008 | 506797   | 610469   | 743855   |
| 100        | 9         | 4        | 0.6871    | 2.4     | 1.9   | 657022    | 870684    | 1.04462e+006 | 1.16272e+007 | 4.58633e+007 | 1.63901e+008 | 566023   | 689515   | 807003   |
| 100        | 9         | 6        | 0.523     | 3.5     | 2.9   | 550054    | 875941    | 1.02052e+006 | 333789       | 6.59602e+006 | 2.84753e+007 | 460662   | 729329   | 838638   |

## Résultats heuristique par insertion pour T=400

| Req. Stat. | Req.Dyn. | Vehicule | Temps     | Front   | Choix | Cout Min. | Cout Moy. | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max. |
|------------|----------|----------|-----------|---------|-------|-----------|-----------|--------------|--------------|--------------|--------------|----------|----------|----------|
| 10         | 7        | 2        | 0.0354    | 2.2     | 1     | 108133    | 251725    | 313555       | 103160       | 1.24208e+007 | 4.07012e+007 | 69409.1  | 205643   | 245681   |
| 10         | 7        | 4        | 0.0414    | 2.8     | 1.1   | 185231    | 264552    | 333432       | 170961       | 7.7676e+006  | 1.84021e+007 | 150132   | 184650   | 222755   |
| 10         | 7        | 6        | 0.0436    | 2       | 1.2   | 230722    | 264814    | 289319       | 6246.78      | 7.16878e+006 | 1.84021e+007 | 115361   | 143684   | 195393   |
| 20         | 6.9      | 2        | 0.0775556 | 1.44444 | 1     | 165443    | 339082    | 438877       | 3.50807e+006 | 2.78378e+007 | 7.40452e+007 | 82721.7  | 297001   | 363907   |
| 20         | 6.9      | 4        | 0.0733    | 3.7     | 1.7   | 369955    | 441452    | 498399       | 252849       | 6.32991e+006 | 1.62404e+007 | 303102   | 347783   | 397250   |
| 20         | 6.9      | 6        | 0.0729    | 4.1     | 2.2   | 297076    | 422413    | 482420       | 130613       | 6.97514e+006 | 4.60036e+007 | 221095   | 280203   | 336127   |
| 30         | 7        | 2        | 0.1349    | 1.9     | 1     | 413428    | 491844    | 566406       | 7.08888e+006 | 4.23481e+007 | 6.934e+007   | 311857   | 377749   | 452393   |
| 30         | 7        | 4        | 0.1137    | 3.8     | 2     | 353720    | 498082    | 564567       | 209582       | 3.60276e+006 | 2.82494e+007 | 311941   | 385765   | 426609   |
| 30         | 7        | 6        | 0.1112    | 4.4     | 2.9   | 433077    | 488355    | 545133       | 53270.9      | 5.73412e+006 | 3.13439e+007 | 312180   | 363424   | 448900   |
| 40         | 7        | 2        | 0.2085    | 1.5     | 1.1   | 379729    | 568706    | 646732       | 2.26033e+007 | 6.9152e+007  | 1.53277e+008 | 398715   | 443562   | 503947   |
| 40         | 7        | 4        | 0.1679    | 3.7     | 2.3   | 530880    | 613218    | 678583       | 1.50723e+006 | 8.3196e+006  | 2.23632e+007 | 418178   | 482455   | 529321   |
| 40         | 7        | 6        | 0.1521    | 4.4     | 3.3   | 568790    | 634862    | 715094       | 159497       | 5.37675e+006 | 1.87978e+007 | 409264   | 503794   | 594289   |
| 50         | 7        | 2        | 0.2869    | 2.6     | 1     | 602475    | 649122    | 696384       | 3.59739e+007 | 7.01585e+007 | 9.1615e+007  | 460228   | 489739   | 529194   |
| 50         | 7        | 4        | 0.2182    | 2.9     | 2.1   | 462451    | 701497    | 824486       | 371976       | 8.51815e+006 | 3.21623e+007 | 387715   | 557544   | 652242   |
| 50         | 7        | 6        | 0.1966    | 3.8     | 2.9   | 491966    | 729930    | 862066       | 300695       | 4.14466e+006 | 1.83224e+007 | 487350   | 603511   | 680910   |
| 60         | 7        | 2        | 0.3909    | 2.4     | 1.1   | 674027    | 733897    | 834629       | 4.15198e+007 | 8.98138e+007 | 1.66642e+008 | 502841   | 550923   | 626510   |
| 60         | 7        | 4        | 0.2624    | 2.9     | 2.2   | 652049    | 809341    | 1.03155e+006 | 1.59677e+006 | 1.73753e+007 | 5.53918e+007 | 512556   | 645016   | 823272   |
| 60         | 7        | 6        | 0.2488    | 2.8     | 3.3   | 803540    | 874315    | 954696       | 222404       | 2.81925e+006 | 9.75226e+006 | 620562   | 703435   | 889053   |
| 70         | 7        | 2        | 0.4972    | 2.4     | 1     | 713458    | 764661    | 847080       | 5.0863e+007  | 1.19138e+008 | 2.33233e+008 | 525887   | 579443   | 720631   |
| 70         | 7        | 4        | 0.3423    | 2.8     | 2     | 782377    | 869814    | 1.03444e+006 | 576129       | 1.79991e+007 | 3.66769e+007 | 613339   | 692241   | 811794   |
| 70         | 7        | 6        | 0.2798    | 3.2     | 3.3   | 656260    | 874951    | 975386       | 412120       | 7.82412e+006 | 2.46887e+007 | 625183   | 713474   | 773103   |
| 80         | 7        | 2        | 0.591     | 1.8     | 1     | 675155    | 725335    | 825393       | 8.49507e+007 | 1.29252e+008 | 1.92156e+008 | 487496   | 539870   | 625619   |
| 80         | 7        | 4        | 0.4019    | 2.3     | 2.3   | 690536    | 817541    | 1.07993e+006 | 2.54727e+006 | 1.95292e+007 | 4.74676e+007 | 562576   | 650697   | 854177   |
| 80         | 7        | 6        | 0.3501    | 3.8     | 3.3   | 664534    | 831517    | 950694       | 167571       | 5.43683e+006 | 3.79873e+007 | 573205   | 681276   | 759429   |
| 90         | 7        | 2        | 0.8282    | 1.8     | 1     | 541930    | 741271    | 814636       | 1.03411e+008 | 1.7334e+008  | 2.77104e+008 | 512686   | 563739   | 612090   |
| 90         | 7        | 4        | 0.4457    | 3.2     | 2     | 650003    | 843121    | 1.01503e+006 | 9.26313e+006 | 3.52992e+007 | 9.89693e+007 | 528113   | 665575   | 781140   |
| 90         | 7        | 6        | 0.386     | 3.1     | 2.6   | 583088    | 743910    | 1.0566e+006  | 562250       | 9.40876e+006 | 3.77258e+007 | 548507   | 644830   | 818070   |
| 100        | 7        | 2        | 0.8288    | 1.9     | 1     | 688339    | 796896    | 930434       | 8.25165e+007 | 1.60465e+008 | 2.73651e+008 | 494194   | 585488   | 707013   |
| 100        | 7        | 4        | 0.5698    | 2.3     | 2     | 624397    | 838330    | 993520       | 1.14799e+007 | 3.21515e+007 | 8.52798e+007 | 540939   | 665908   | 775360   |
| 100        | 7        | 6        | 0.4554    | 3.6     | 2.8   | 522058    | 830415    | 963842       | 392202       | 6.8777e+006  | 2.08459e+007 | 430555   | 692727   | 820598   |

## Résultats heuristique par insertion pour T=600

| Req. Stat. | Req. Dyn. | Vehicule | Temps  | Front | Choix | Cout Min. | Cout Moy. | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max. |
|------------|-----------|----------|--------|-------|-------|-----------|-----------|--------------|--------------|--------------|--------------|----------|----------|----------|
| 10         | 4.8       | 2        | 0.0346 | 1.8   | 1     | 81566.9   | 222572    | 273207       | 16621.4      | 8.43274e+006 | 2.35461e+007 | 49355.3  | 18187.1  | 210051   |
| 10         | 4.8       | 4        | 0.0378 | 2.6   | 1.2   | 149863    | 239804    | 334495       | 79315.1      | 5.19605e+006 | 1.41616e+007 | 112813   | 165184   | 227250   |
| 10         | 4.8       | 6        | 0.0375 | 2.1   | 1.2   | 193372    | 238507    | 270720       | 117378       | 5.5529e+006  | 1.37874e+007 | 96686    | 138441   | 175061   |
| 20         | 4.8       | 2        | 0.0735 | 1.75  | 1     | 219762    | 337412    | 412410       | 1.18001e+006 | 2.53261e+007 | 6.23651e+007 | 230751   | 297944   | 336666   |
| 20         | 4.8       | 4        | 0.0659 | 2.7   | 1.7   | 331605    | 412071    | 451236       | 35980.4      | 8.46832e+006 | 3.39382e+007 | 272315   | 324308   | 362013   |
| 20         | 4.8       | 6        | 0.071  | 2.4   | 1.7   | 257393    | 390285    | 456441       | 502.43       | 6.49385e+006 | 3.00625e+007 | 189692   | 249523   | 308631   |
| 30         | 5         | 2        | 0.1126 | 1.8   | 1     | 396863    | 470873    | 519002       | 1.77126e+006 | 2.14456e+007 | 5.904e+007   | 299342   | 358132   | 398467   |
| 30         | 5         | 4        | 0.1006 | 3.7   | 2.3   | 338525    | 464520    | 536270       | 20464        | 1.53389e+006 | 1.11728e+007 | 298917   | 361320   | 420840   |
| 30         | 5         | 6        | 0.1039 | 3.7   | 2.1   | 403078    | 466377    | 517863       | 40542.3      | 948923       | 6.36249e+006 | 302929   | 335972   | 426420   |
| 40         | 5         | 2        | 0.1633 | 1.5   | 1     | 356889    | 540203    | 598298       | 9.87668e+006 | 3.48139e+007 | 7.87691e+007 | 374039   | 421705   | 472120   |
| 40         | 5         | 4        | 0.1436 | 3.4   | 2.5   | 469453    | 589480    | 645114       | 36077.8      | 4.34188e+006 | 1.93367e+007 | 359405   | 460224   | 496153   |
| 40         | 5         | 6        | 0.1344 | 3.8   | 3.2   | 537165    | 620025    | 699959       | 22.8759      | 2.27078e+006 | 1.48424e+007 | 394671   | 493348   | 588762   |
| 50         | 5         | 2        | 0.2387 | 2.4   | 1     | 582390    | 625184    | 666346       | 2.31205e+007 | 4.42655e+007 | 7.90781e+007 | 439030   | 466808   | 497929   |
| 50         | 5         | 4        | 0.1859 | 2.6   | 2.4   | 439257    | 680395    | 784559       | 369990       | 7.59255e+006 | 2.55819e+007 | 365324   | 542236   | 622095   |
| 50         | 5         | 6        | 0.1736 | 3.4   | 2.8   | 442448    | 704175    | 851423       | 200447       | 2.38268e+006 | 7.13321e+006 | 428017   | 588862   | 667891   |
| 60         | 5         | 2        | 0.3275 | 2.2   | 1.2   | 656686    | 713470    | 792628       | 2.6792e+007  | 6.65359e+007 | 1.10526e+008 | 489332   | 531947   | 594307   |
| 60         | 5         | 4        | 0.2271 | 2.6   | 2.3   | 620376    | 779709    | 998641       | 742617       | 1.09222e+007 | 3.29634e+007 | 497375   | 623224   | 795886   |
| 60         | 5         | 6        | 0.2124 | 2.1   | 3     | 769098    | 862425    | 939402       | 14511.4      | 1.80485e+006 | 5.99153e+006 | 630412   | 710766   | 873855   |
| 70         | 5         | 2        | 0.3944 | 2.3   | 1.2   | 678028    | 724944    | 806159       | 2.5218e+007  | 6.962e+007   | 1.50012e+008 | 497195   | 539487   | 677664   |
| 70         | 5         | 4        | 0.3016 | 2.8   | 2.3   | 743812    | 824399    | 990919       | 346182       | 1.21124e+007 | 3.29837e+007 | 577302   | 649077   | 769859   |
| 70         | 5         | 6        | 0.2555 | 3.5   | 3.4   | 636467    | 836984    | 943466       | 6099.69      | 4.51258e+006 | 2.12311e+007 | 608912   | 683269   | 739678   |
| 80         | 5         | 2        | 0.4887 | 1.9   | 1.2   | 640188    | 694157    | 788004       | 4.33097e+007 | 7.91331e+007 | 1.55674e+008 | 463920   | 514831   | 597371   |
| 80         | 5         | 4        | 0.3444 | 2.2   | 2.2   | 642190    | 780217    | 1.05023e+006 | 1.88914e+006 | 1.19973e+007 | 4.5894e+007  | 517022   | 619468   | 832242   |
| 80         | 5         | 6        | 0.3069 | 3.4   | 3.4   | 647369    | 817950    | 930740       | 128733       | 3.57313e+006 | 1.4736e+007  | 549780   | 665505   | 744778   |
| 90         | 5         | 2        | 0.6429 | 1.8   | 1.1   | 517073    | 703059    | 761038       | 7.89226e+007 | 1.07933e+008 | 2.1235e+008  | 488079   | 532424   | 567068   |
| 90         | 5         | 4        | 0.3861 | 3     | 2     | 619560    | 798709    | 984146       | 1.90722e+006 | 3.11855e+007 | 6.74651e+007 | 500762   | 627959   | 763837   |
| 90         | 5         | 6        | 0.3373 | 3.3   | 2.6   | 560392    | 796309    | 1.03167e+006 | 121114       | 1.08802e+007 | 3.51719e+007 | 521272   | 659986   | 805606   |
| 100        | 5         | 2        | 0.7078 | 2.3   | 1     | 656627    | 758076    | 897456       | 3.35756e+007 | 9.7388e+007  | 1.55173e+008 | 471157   | 555370   | 673415   |
| 100        | 5         | 4        | 0.4727 | 2.3   | 1.9   | 564214    | 803742    | 948278       | 1.56674e+006 | 1.88676e+007 | 5.69712e+007 | 489681   | 639172   | 752155   |
| 100        | 5         | 6        | 0.4106 | 3.1   | 3     | 498380    | 806788    | 927675       | 1381.31      | 2.87763e+006 | 1.89839e+007 | 412302   | 671229   | 779975   |

## Résultats heuristique par insertion pour T=800

| Req. Stat. | Req. Dyn. | Vehicule | Temps     | Front   | Choix   | Cout Min. | Cout Moy. | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max. |
|------------|-----------|----------|-----------|---------|---------|-----------|-----------|--------------|--------------|--------------|--------------|----------|----------|----------|
| 10         | 2.7       | 2        | 0.0295    | 1.5     | 1.1     | 55302.5   | 187753    | 264117       | 0            | 1.83934e+006 | 6.0665e+006  | 27651.2  | 155226   | 227037   |
| 10         | 2.7       | 4        | 0.0383    | 1.8     | 1.1     | 117771    | 200137    | 295570       | 803.386      | 1.60718e+006 | 5.61688e+006 | 86844.2  | 140076   | 207216   |
| 10         | 2.7       | 6        | 0.0418    | 1.7     | 1.2     | 183284    | 211861    | 246916       | 803.386      | 735396       | 2.50612e+006 | 91642    | 124291   | 160255   |
| 20         | 2.6       | 2        | 0.0672857 | 1.14286 | 1.14286 | 215080    | 315349    | 393383       | 865196       | 9.39876e+006 | 2.37984e+007 | 225834   | 273495   | 296515   |
| 20         | 2.6       | 4        | 0.0635    | 2.3     | 1.7     | 293405    | 380185    | 431820       | 28962.5      | 4.96794e+006 | 1.4679e+007  | 242411   | 295657   | 351996   |
| 20         | 2.6       | 6        | 0.0653    | 2.5     | 1.6     | 232146    | 374987    | 425892       | 0            | 2.89599e+006 | 1.25102e+007 | 179715   | 243691   | 279743   |
| 30         | 2.9       | 2        | 0.118     | 1.7     | 1       | 400328    | 435038    | 487252       | 1.25046e+006 | 6.90605e+006 | 2.09848e+007 | 299954   | 333172   | 380342   |
| 30         | 2.9       | 4        | 0.0983    | 2.8     | 2.2     | 332789    | 451395    | 512434       | 19421.9      | 2.16261e+006 | 9.68252e+006 | 290019   | 346569   | 414542   |
| 30         | 2.9       | 6        | 0.0931    | 2.7     | 2.7     | 387065    | 445505    | 486113       | 44478.3      | 1.81402e+006 | 1.28363e+007 | 270224   | 317295   | 389848   |
| 40         | 3         | 2        | 0.1427    | 1.8     | 1.2     | 319943    | 512326    | 559358       | 3.77346e+006 | 1.01222e+007 | 1.89928e+007 | 335940   | 394694   | 427141   |
| 40         | 3         | 4        | 0.1336    | 3.1     | 2.3     | 454402    | 561171    | 610279       | 159.987      | 1.50801e+006 | 7.81415e+006 | 346844   | 433594   | 479678   |
| 40         | 3         | 6        | 0.1328    | 3.1     | 2.9     | 493074    | 592448    | 694579       | 12491.5      | 960750       | 5.11952e+006 | 372036   | 468407   | 583150   |
| 50         | 3         | 2        | 0.1989    | 2.6     | 1.1     | 556390    | 602302    | 644887       | 1.22902e+007 | 2.63072e+007 | 5.24678e+007 | 406903   | 447444   | 493387   |
| 50         | 3         | 4        | 0.173     | 2.5     | 2.6     | 419872    | 641735    | 717230       | 2274.43      | 4.59163e+006 | 2.53837e+007 | 346388   | 509786   | 578805   |
| 50         | 3         | 6        | 0.1602    | 3.2     | 2.7     | 422601    | 697622    | 812345       | 507.682      | 1.75942e+006 | 7.77612e+006 | 406292   | 563982   | 635222   |
| 60         | 3         | 2        | 0.2599    | 2.4     | 1.6     | 628752    | 677697    | 756145       | 2.10381e+006 | 2.97692e+007 | 7.43757e+007 | 468366   | 500861   | 565912   |
| 60         | 3         | 4        | 0.2154    | 2.9     | 2.2     | 589425    | 753418    | 975135       | 4980.7       | 4.6051e+006  | 3.28132e+007 | 458690   | 597608   | 764615   |
| 60         | 3         | 6        | 0.2072    | 1.9     | 3.2     | 724013    | 822539    | 893306       | 35.8329      | 426722       | 3.59445e+006 | 618832   | 671832   | 824784   |
| 70         | 3         | 2        | 0.3162    | 2.6     | 1.3     | 638643    | 694040    | 762352       | 1.5097e+007  | 4.21969e+007 | 1.05568e+008 | 459416   | 508272   | 618398   |
| 70         | 3         | 4        | 0.2521    | 3.3     | 2.5     | 713112    | 780797    | 959546       | 354976       | 1.0291e+007  | 6.66256e+007 | 562046   | 612190   | 742739   |
| 70         | 3         | 6        | 0.2361    | 3.7     | 3.6     | 620248    | 805389    | 913978       | 0.00573567   | 3.88065e+006 | 2.54851e+007 | 583058   | 651977   | 720407   |
| 80         | 3         | 2        | 0.3949    | 1.6     | 1.4     | 666622    | 666015    | 751158       | 1.58822e+007 | 4.01528e+007 | 8.29002e+007 | 441482   | 486056   | 571449   |
| 80         | 3         | 4        | 0.2973    | 2.6     | 2.5     | 603121    | 754126    | 1.00435e+006 | 132090       | 4.6328e+006  | 1.37598e+007 | 475999   | 595803   | 799462   |
| 80         | 3         | 6        | 0.2724    | 2.9     | 3.3     | 623890    | 783383    | 911840       | 186.446      | 3.49961e+006 | 1.38373e+007 | 531506   | 631339   | 692911   |
| 90         | 3         | 2        | 0.5068    | 1.8     | 1.2     | 492588    | 671436    | 724543       | 1.59347e+007 | 5.16091e+007 | 7.66264e+007 | 459193   | 503264   | 542999   |
| 90         | 3         | 4        | 0.3388    | 2.9     | 2.3     | 600201    | 785471    | 954907       | 557515       | 8.69226e+006 | 2.44102e+007 | 482044   | 614464   | 744382   |
| 90         | 3         | 6        | 0.3101    | 2.7     | 2.5     | 525477    | 745769    | 1.01011e+006 | 88503.2      | 3.41012e+006 | 8.76089e+006 | 479130   | 620447   | 785131   |
| 100        | 3         | 2        | 0.5787    | 2       | 1.3     | 624503    | 727666    | 853088       | 1.87165e+007 | 5.18954e+007 | 8.4663e+007  | 449520   | 531138   | 648588   |
| 100        | 3         | 4        | 0.4055    | 2.4     | 2.1     | 533268    | 761217    | 886111       | 178262       | 8.65988e+006 | 2.37422e+007 | 457187   | 605447   | 701231   |
| 100        | 3         | 6        | 0.3621    | 3.4     | 3       | 475334    | 771087    | 873366       | 137.941      | 3.10904e+006 | 1.94127e+007 | 384646   | 642568   | 761649   |

| Req. Stat. | Req.Dyn. | Vehicule | % Reussi | % Insert | Temps   | Front   | Cout Min.    | Cout Moy.    | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy.     | Co2 Max.   |
|------------|----------|----------|----------|----------|---------|---------|--------------|--------------|--------------|--------------|--------------|--------------|----------|--------------|------------|
| 10         | 10       | 2        | 60       | 90.25    | 2.40733 | 2.33333 | 233396       | 276105       | 322689       | 17920.1      | 646697       | 3.61223e+006 | 167059   | 195047       | 220541     |
| 10         | 10       | 4        | 90       | 96       | 6.12367 | 3.44444 | 218455       | 262747       | 307522       | 90845.4      | 2.41222e+006 | 1.44987e+007 | 157706   | 187803       | 238635     |
| 10         | 10       | 6        | 100      | 100      | 7.8458  | 4.8     | 230808       | 285382       | 351655       | 0            | 368513       | 3.61935e+006 | 131174   | 180737       | 236211     |
| 20         | 10       | 2        | 10       | 45.7345  | 6.193   | 2       | 356460       | 366271       | 376082       | 1.951e+006   | 2.06862e+006 | 2.18624e+006 | 278598   | 283074       | 287550     |
| 20         | 10       | 4        | 100      | 100      | 17.5968 | 2       | 326387       | 379482       | 441102       | 6.84764      | 1.01949e+006 | 3.95212e+006 | 232973   | 286617       | 336004     |
| 20         | 10       | 6        | 100      | 100      | 26.5654 | 4.5     | 338468       | 402954       | 472918       | 129.988      | 1.87302e+006 | 7.5728e+006  | 211301   | 279286       | 331968     |
| 30         | 10       | 2        | 0        | 19.5127  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 30         | 10       | 4        | 50       | 77.0712  | 18.1394 | 5.4     | 527999       | 586302       | 671914       | 497005       | 3.38216e+006 | 9.35595e+006 | 405687   | 448839       | 514377     |
| 30         | 10       | 6        | 100      | 100      | 40.5946 | 7       | 482484       | 558444       | 633699       | 38660.2      | 852961       | 6.75684e+006 | 391022   | 432377       | 503652     |
| 40         | 10       | 2        | 0        | 17       | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 40         | 10       | 4        | 60       | 79.5     | 39.084  | 3.16667 | 649160       | 719124       | 768755       | 164594       | 4.50167e+006 | 1.57383e+007 | 516261   | 573750       | 615562     |
| 40         | 10       | 6        | 100      | 100      | 59.3238 | 6.9     | 627320       | 727212       | 819878       | 20833.2      | 1.86109e+006 | 1.32062e+007 | 475937   | 572499       | 666713     |
| 50         | 10       | 2        | 0        | 10.2654  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 50         | 10       | 4        | 30       | 68.6821  | 56.304  | 3.33333 | 771871       | 835054       | 904645       | 197512       | 1.86733e+006 | 6.1029e+006  | 614945   | 669194       | 738519     |
| 50         | 10       | 6        | 90       | 96.4706  | 96.0646 | 3.77778 | 779421       | 854424       | 923536       | 111016       | 1.32333e+006 | 5.98968e+006 | 591401   | 680307       | 736992     |
| 60         | 10       | 2        | 0        | 10.2271  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 60         | 10       | 4        | 10       | 50.3268  | 101.859 | 2       | 957612       | 960533       | 963454       | 3.04526e+006 | 3.04526e+006 | 3.04526e+006 | 758529   | 760188       | 761847     |
| 60         | 10       | 6        | 90       | 95.5     | 160.676 | 5.88889 | 904584       | 1.02589e+006 | 1.1064e+006  | 238497       | 1.24372e+006 | 2.094e+006   | 718737   | 812810       | 874113     |
| 70         | 10       | 2        | 0        | 10.25    | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 70         | 10       | 4        | 10       | 50.625   | 72.125  | 4       | 996094       | 1.00705e+006 | 1.0156e+006  | 2.03405e+006 | 2.03467e+006 | 2.03647e+006 | 796715   | 805677       | 815753     |
| 70         | 10       | 6        | 20       | 61       | 164.236 | 2.5     | 1.03425e+006 | 1.06698e+006 | 1.09158e+006 | 1.77612e+006 | 3.06492e+006 | 4.1004e+006  | 837047   | 846725       | 853784     |
| 80         | 10       | 2        | 0        | 8.37182  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 80         | 10       | 4        | 0        | 22.0299  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 80         | 10       | 6        | 30       | 61.1173  | 168.101 | 3.33333 | 1.09308e+006 | 1.15741e+006 | 1.21263e+006 | 1.71915e+006 | 2.95321e+006 | 6.73607e+006 | 863158   | 917912       | 946197     |
| 90         | 10       | 2        | 0        | 8.66382  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 90         | 10       | 4        | 0        | 17.7319  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 90         | 10       | 6        | 10       | 51.1942  | 709.415 | 10      | 1.24708e+006 | 1.25791e+006 | 1.26529e+006 | 9.93398e+006 | 1.0123e+007  | 1.05886e+007 | 993709   | 1.00122e+006 | 1.006e+006 |
| 100        | 10       | 2        | 0        | 5.29265  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 100        | 10       | 4        | 0        | 17.2418  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |
| 100        | 10       | 6        | 0        | 39.0361  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -            | -          |

| Req. Stat. | Req. Dyn. | Vehicule | % Reussi | % Insert | Temps   | Front   | Cout Min.    | Cout Moy.    | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max.     |
|------------|-----------|----------|----------|----------|---------|---------|--------------|--------------|--------------|--------------|--------------|--------------|----------|----------|--------------|
| 10         | 9         | 2        | 50       | 74.4543  | 1.3758  | 2.2     | 244145       | 278786       | 310664       | 478695       | 3.57681e+006 | 8.36792e+006 | 172145   | 206838   | 237995       |
| 10         | 9         | 4        | 100      | 100      | 3.0159  | 4.9     | 221153       | 283087       | 366100       | 1590.26      | 1.81865e+006 | 1.09865e+007 | 160559   | 219171   | 282435       |
| 10         | 9         | 6        | 100      | 100      | 4.4073  | 6.2     | 182830       | 275119       | 339331       | 159.428      | 1.85772e+006 | 1.17045e+007 | 110847   | 171033   | 249034       |
| 20         | 9         | 2        | 0        | 30.2116  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 20         | 9         | 4        | 90       | 90.2041  | 9.901   | 3.11111 | 335544       | 392423       | 457173       | 31994.3      | 847672       | 2.68023e+006 | 238991   | 306615   | 347355       |
| 20         | 9         | 6        | 100      | 100      | 14.2298 | 4.8     | 335419       | 404834       | 524196       | 864.498      | 1.42e+006    | 9.2038e+006  | 241931   | 294316   | 392344       |
| 30         | 9         | 2        | 0        | 19.5062  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 30         | 9         | 4        | 60       | 72.9617  | 15.6573 | 6.66667 | 511006       | 547853       | 642513       | 78388.9      | 793503       | 3.29522e+006 | 394854   | 430195   | 499598       |
| 30         | 9         | 6        | 100      | 100      | 20.0452 | 7       | 490288       | 538953       | 589001       | 95759.6      | 742047       | 4.98479e+006 | 338981   | 407368   | 475941       |
| 40         | 9         | 2        | 0        | 14.1983  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 40         | 9         | 4        | 50       | 69.8082  | 25.955  | 3.4     | 639482       | 670737       | 731979       | 145648       | 2.38295e+006 | 5.68415e+006 | 490473   | 518767   | 573529       |
| 40         | 9         | 6        | 100      | 100      | 33.2386 | 5.7     | 623382       | 693055       | 747846       | 15280.9      | 2.08907e+006 | 1.03894e+007 | 473697   | 537713   | 582304       |
| 50         | 9         | 2        | 0        | 6.62743  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 50         | 9         | 4        | 40       | 63.0357  | 30.3483 | 3.75    | 763982       | 816603       | 875144       | 1.06069e+006 | 2.70307e+006 | 7.89951e+006 | 619455   | 651049   | 688506       |
| 50         | 9         | 6        | 70       | 77.8846  | 58.8753 | 6.14286 | 723509       | 822996       | 885539       | 12600.8      | 598870       | 3.64503e+006 | 552359   | 648979   | 723026       |
| 60         | 9         | 2        | 0        | 4.46598  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 60         | 9         | 4        | 20       | 43.309   | 68.7575 | 5       | 927482       | 1.00704e+006 | 1.15372e+006 | 2.66269e+006 | 3.5112e+006  | 1.03375e+007 | 716304   | 806430   | 968567       |
| 60         | 9         | 6        | 90       | 94.4737  | 110.738 | 5.11111 | 912284       | 974979       | 1.04977e+006 | 231830       | 1.6811e+006  | 8.8695e+006  | 721826   | 781695   | 851545       |
| 70         | 9         | 2        | 0        | 4.94935  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 70         | 9         | 4        | 0        | 22.9744  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 70         | 9         | 6        | 20       | 35.8226  | 96.774  | 4.5     | 1.0643e+006  | 1.08502e+006 | 1.1126e+006  | 720366       | 724775       | 740199       | 824784   | 857394   | 881802       |
| 80         | 9         | 2        | 0        | 6.17276  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 80         | 9         | 4        | 0        | 21.0037  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 80         | 9         | 6        | 30       | 37.605   | 118.146 | 2       | 1.04376e+006 | 1.0684e+006  | 1.14396e+006 | 1.01465e+006 | 1.14188e+006 | 1.3572e+006  | 817424   | 848297   | 930669       |
| 90         | 9         | 2        | 0        | 2.56046  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 90         | 9         | 4        | 0        | 16.9323  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 90         | 9         | 6        | 20       | 48.9702  | 132.281 | 8.5     | 1.09892e+006 | 1.182e+006   | 1.25929e+006 | 1.51275e+006 | 2.4241e+006  | 3.7334e+006  | 923122   | 982473   | 1.04167e+006 |
| 100        | 9         | 2        | 0        | 4.96431  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 100        | 9         | 4        | 0        | 9.97431  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -            |
| 100        | 9         | 6        | 10       | 23.342   | 128.283 | 2       | 1.14606e+006 | 1.1477e+006  | 1.14934e+006 | 3.10542e+006 | 3.10561e+006 | 3.10579e+006 | 941392   | 941839   | 942286       |

## Résultats RBS pour T=400

| Req. Stat. | Req. Dyn. | Vehicule | % Reussi | % Insert | Temps   | Front   | Cout Min.    | Cout Moy.    | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max. |
|------------|-----------|----------|----------|----------|---------|---------|--------------|--------------|--------------|--------------|--------------|--------------|----------|----------|----------|
| 10         | 7         | 2        | 40       | 72.121   | 0.6575  | 2.5     | 222474       | 236196       | 254856       | 21068.2      | 5.92864e+006 | 1.64505e+007 | 169933   | 191410   | 199395   |
| 10         | 7         | 4        | 100      | 100      | 1.4167  | 3.9     | 207767       | 244766       | 301482       | 19.3497      | 445266       | 4.67673e+006 | 147648   | 184113   | 234823   |
| 10         | 7         | 6        | 100      | 100      | 1.9407  | 4.8     | 180457       | 242110       | 281921       | 54.4271      | 531616       | 4.37388e+006 | 113079   | 153083   | 208074   |
| 20         | 6.9       | 2        | 10       | 48.0995  | 1.516   | 1       | 306743       | 306743       | 306743       | 2.69114e+006 | 2.69114e+006 | 2.69114e+006 | 247342   | 247342   | 247342   |
| 20         | 6.9       | 4        | 90       | 91.1111  | 5.01722 | 6       | 339242       | 419097       | 492466       | 225576       | 2.08769e+006 | 6.62301e+006 | 253164   | 317197   | 378398   |
| 20         | 6.9       | 6        | 90       | 91.1111  | 9.60067 | 4.33333 | 331863       | 378329       | 451865       | 3.91294      | 594507       | 6.64026e+006 | 236832   | 265638   | 304371   |
| 30         | 7         | 2        | 0        | 18.3335  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 30         | 7         | 4        | 80       | 92.6282  | 6.40275 | 5.25    | 465172       | 527086       | 592380       | 0            | 714281       | 2.51607e+006 | 342530   | 406608   | 474566   |
| 30         | 7         | 6        | 100      | 100      | 9.0899  | 6.7     | 473618       | 518689       | 559763       | 98.6526      | 687891       | 7.06864e+006 | 325010   | 373584   | 410292   |
| 40         | 7         | 2        | 0        | 9.9792   | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 40         | 7         | 4        | 40       | 50.9816  | 15.375  | 5       | 607704       | 634671       | 663472       | 2828.3       | 4.70828e+006 | 1.00432e+007 | 468151   | 493344   | 510853   |
| 40         | 7         | 6        | 90       | 94.058   | 19.9693 | 6.33333 | 581867       | 661281       | 779861       | 2828.3       | 1.5885e+006  | 8.23642e+006 | 417025   | 512808   | 647946   |
| 50         | 7         | 2        | 0        | 13.7629  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 50         | 7         | 4        | 20       | 37.7654  | 17.33   | 3       | 691380       | 739486       | 782863       | 2.01441e+006 | 5.31509e+006 | 9.88886e+006 | 554738   | 593238   | 632355   |
| 50         | 7         | 6        | 50       | 55.0995  | 44.2876 | 10      | 702161       | 852109       | 932294       | 3.97904e-013 | 1.80726e+006 | 6.9023e+006  | 539764   | 666758   | 735042   |
| 60         | 7         | 2        | 0        | 7.96421  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 60         | 7         | 4        | 0        | 19.8041  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 60         | 7         | 6        | 60       | 63.4992  | 69.0087 | 4.83333 | 875701       | 924024       | 994913       | 46851.4      | 866139       | 4.52298e+006 | 693939   | 741144   | 796917   |
| 70         | 7         | 2        | 0        | 4.4975   | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 70         | 7         | 4        | 0        | 9.30038  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 70         | 7         | 6        | 30       | 44.417   | 79.479  | 7.66667 | 951494       | 1.0075e+006  | 1.07138e+006 | 1.42268e+006 | 2.43479e+006 | 2.68364e+006 | 720307   | 790700   | 863146   |
| 80         | 7         | 2        | 0        | 5.38727  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 80         | 7         | 4        | 0        | 19.4881  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 80         | 7         | 6        | 20       | 38.2516  | 71.6365 | 3.5     | 944019       | 963571       | 976885       | 1.35019e+006 | 1.38365e+006 | 1.46351e+006 | 770612   | 797256   | 817430   |
| 90         | 7         | 2        | 0        | 4.86354  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 90         | 7         | 4        | 0        | 13.9265  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 90         | 7         | 6        | 20       | 47.5508  | 150.103 | 1.5     | 1.04968e+006 | 1.06515e+006 | 1.09414e+006 | 696496       | 1.12514e+006 | 1.98243e+006 | 825314   | 841091   | 868660   |
| 100        | 7         | 2        | 0        | 2.8403   | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 100        | 7         | 4        | 0        | 9.35233  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 100        | 7         | 6        | 0        | 12.6801  | -       | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |

| Req. Stat. | Req. Dyn. | Vehicule | % Reussi | % Insert | Temps    | Front   | Cout Min.    | Cout Moy.    | Cout Max.    | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max. |
|------------|-----------|----------|----------|----------|----------|---------|--------------|--------------|--------------|--------------|--------------|--------------|----------|----------|----------|
| 10         | 4.8       | 2        | 30       | 64.3496  | 0.293667 | 5.66667 | 177515       | 204889       | 236165       | 0            | 1.82479e+006 | 1.27094e+007 | 149247   | 166506   | 184698   |
| 10         | 4.8       | 4        | 100      | 100      | 0.6655   | 5.6     | 143339       | 224058       | 296640       | 1.13687e-013 | 1.57862e+006 | 1.45791e+007 | 103733   | 164965   | 221543   |
| 10         | 4.8       | 6        | 100      | 100      | 0.8326   | 3.6     | 183777       | 230266       | 279618       | 0            | 2.62778      | 1.51222e+006 | 106097   | 140261   | 196300   |
| 20         | 4.8       | 2        | 20       | 39.9977  | 2.0765   | 1.5     | 340544       | 344492       | 348144       | 1.51105e+006 | 2.03493e+006 | 2.76768e+006 | 282545   | 289824   | 296599   |
| 20         | 4.8       | 4        | 70       | 72.791   | 1.70586  | 3.28571 | 291880       | 404661       | 467143       | 899.816      | 3.04122e+006 | 1.22019e+007 | 230332   | 315518   | 354086   |
| 20         | 4.8       | 6        | 60       | 68.149   | 2.37333  | 6       | 320892       | 386081       | 444877       | 3.91294      | 1.73352      | 1.18433e+006 | 203631   | 258297   | 297397   |
| 30         | 5         | 2        | 0        | 23.3031  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 30         | 5         | 4        | 90       | 98.5714  | 3.14056  | 5.11111 | 432034       | 486763       | 542257       | 48509.5      | 626007       | 2.36562e+006 | 325123   | 376485   | 415604   |
| 30         | 5         | 6        | 100      | 100      | 4.0057   | 6.2     | 409345       | 490643       | 548812       | 1.20462      | 510633       | 5.11229e+006 | 297592   | 359814   | 453322   |
| 40         | 5         | 2        | 0        | 20.3849  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 40         | 5         | 4        | 60       | 87.4628  | 6.04767  | 2.5     | 507994       | 589472       | 625366       | 77981.3      | 2.31695e+006 | 5.59003e+006 | 387707   | 455135   | 488788   |
| 40         | 5         | 6        | 100      | 100      | 7.3277   | 6.5     | 553937       | 617003       | 734842       | 0.00061217   | 817209       | 6.92494e+006 | 389925   | 474440   | 613611   |
| 50         | 5         | 2        | 0        | 10.4422  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 50         | 5         | 4        | 30       | 32.6106  | 16.935   | 6.33333 | 665419       | 740959       | 781869       | 444325       | 2.75419e+006 | 3.42973e+006 | 526725   | 604533   | 631644   |
| 50         | 5         | 6        | 50       | 50.9076  | 19.2988  | 4.2     | 633467       | 834220       | 893470       | 227.057      | 827591       | 1.10557e+006 | 508459   | 651405   | 712581   |
| 60         | 5         | 2        | 0        | 3.76475  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 60         | 5         | 4        | 0        | 7.27877  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 60         | 5         | 6        | 40       | 44.4748  | 32.3128  | 4.25    | 855659       | 909587       | 963548       | 16416.2      | 1.43022e+006 | 2.70242e+006 | 649091   | 723249   | 798392   |
| 70         | 5         | 2        | 0        | 6.94763  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 70         | 5         | 4        | 0        | 20.5524  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 70         | 5         | 6        | 30       | 31.513   | 53.4053  | 6       | 840162       | 892013       | 936518       | 27204.9      | 784215       | 2.61799e+006 | 666792   | 716605   | 756390   |
| 80         | 5         | 2        | 0        | 4.4146   | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 80         | 5         | 4        | 0        | 10.6961  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 80         | 5         | 6        | 30       | 43.4047  | 81.009   | 4.66667 | 862248       | 885710       | 908880       | 256339       | 698470       | 1.06218e+006 | 699634   | 715294   | 740851   |
| 90         | 5         | 2        | 0        | 7.91556  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 90         | 5         | 4        | 0        | 14.7091  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 90         | 5         | 6        | 20       | 38.1654  | 40.099   | 2       | 945682       | 1.02391e+006 | 1.05613e+006 | 811132       | 7.16456e+006 | 9.4478e+006  | 751732   | 811298   | 833162   |
| 100        | 5         | 2        | 0        | 3.91045  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 100        | 5         | 4        | 0        | 4.57958  | -        | 0       | -            | -            | -            | -            | -            | -            | -        | -        | -        |
| 100        | 5         | 6        | 10       | 22.6481  | 49.616   | 14      | 1.03024e+006 | 1.05562e+006 | 1.08026e+006 | 1.02025e+006 | 1.03359e+006 | 1.0521e+006  | 857078   | 876726   | 893808   |

| Req. Stat. | Req. Dyn. | Vehicule | % Reussi | % Insert | Temps     | Front   | Cout Min. | Cout Moy. | Cout Max. | Qos Min.     | Qos Moy.     | Qos Max.     | Co2 Min. | Co2 Moy. | Co2 Max. |
|------------|-----------|----------|----------|----------|-----------|---------|-----------|-----------|-----------|--------------|--------------|--------------|----------|----------|----------|
| 10         | 2.7       | 2        | 60       | 72.2179  | 0.0495    | 1       | 102803    | 199719    | 280077    | 0            | 1.46355e+006 | 3.79919e+006 | 69560.9  | 160309   | 237967   |
| 10         | 2.7       | 4        | 90       | 100      | 0.0858889 | 3.11111 | 122673    | 197037    | 289162    | 0            | 398904       | 2.63069e+006 | 84546    | 145535   | 214703   |
| 10         | 2.7       | 6        | 90       | 92       | 0.114     | 4       | 168061    | 229300    | 297383    | 20.5174      | 545945       | 3.97266e+006 | 84171.8  | 141377   | 197165   |
| 20         | 2.6       | 2        | 40       | 61.5895  | 0.3955    | 1       | 218675    | 300080    | 348429    | 0            | 3.46964e+006 | 9.41947e+006 | 206258   | 261980   | 289022   |
| 20         | 2.6       | 4        | 80       | 92.8205  | 0.199625  | 3       | 282776    | 394598    | 464082    | 864.498      | 4.12692e+006 | 1.52512e+007 | 229537   | 306979   | 341956   |
| 20         | 2.6       | 6        | 70       | 82.0192  | 0.353429  | 3.28571 | 326127    | 379876    | 419218    | 4.26326e-013 | 900689       | 3.82098e+006 | 187516   | 257157   | 282159   |
| 30         | 2.9       | 2        | 20       | 50.7604  | 0.678     | 1       | 407998    | 408309    | 408621    | 1.22067e+006 | 1.25801e+006 | 1.29534e+006 | 303792   | 307554   | 311316   |
| 30         | 2.9       | 4        | 90       | 98       | 0.624     | 4.77778 | 383819    | 468259    | 513830    | 474.242      | 1.10826e+006 | 6.04976e+006 | 318462   | 356129   | 414542   |
| 30         | 2.9       | 6        | 90       | 90.5882  | 0.517     | 4.77778 | 407872    | 455404    | 511797    | 1.13687e-013 | 1.08113e+006 | 4.92693e+006 | 285395   | 328067   | 404946   |
| 40         | 3         | 2        | 0        | 18.7615  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 40         | 3         | 4        | 70       | 83.6228  | 1.86714   | 2.42857 | 500310    | 569912    | 638268    | 157.119      | 453344       | 4.36943e+006 | 373846   | 443077   | 471903   |
| 40         | 3         | 6        | 90       | 90.4762  | 1.41033   | 2.66667 | 531052    | 603337    | 662140    | 6.82121e-013 | 236140       | 3.71106e+006 | 358736   | 475615   | 559351   |
| 50         | 3         | 2        | 0        | 11.0094  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 50         | 3         | 4        | 30       | 35.8247  | 2.98367   | 1.66667 | 622617    | 663933    | 711243    | 12776.4      | 2.3432e+006  | 4.99359e+006 | 491544   | 519206   | 539940   |
| 50         | 3         | 6        | 70       | 78.7254  | 3.73386   | 6.71429 | 581700    | 732242    | 841278    | 0.00128724   | 559591       | 3.15903e+006 | 466856   | 586951   | 667361   |
| 60         | 3         | 2        | 0        | 8.46509  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 60         | 3         | 4        | 30       | 46.2541  | 6.15667   | 2.66667 | 719090    | 757985    | 837809    | 382355       | 705624       | 2.49095e+006 | 577920   | 607755   | 675839   |
| 60         | 3         | 6        | 70       | 75.7709  | 10.1946   | 6.71429 | 736510    | 842570    | 919661    | 18.9271      | 309894       | 3.08451e+006 | 624899   | 679790   | 750708   |
| 70         | 3         | 2        | 0        | 8.17806  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 70         | 3         | 4        | 10       | 24.3422  | 41.787    | 11      | 745158    | 761391    | 777533    | 57058.6      | 165135       | 283846       | 603654   | 617241   | 631647   |
| 70         | 3         | 6        | 30       | 41.5204  | 22.463    | 8.33333 | 675040    | 842638    | 916083    | 18.2113      | 2.17276e+006 | 5.29516e+006 | 586685   | 671260   | 738489   |
| 80         | 3         | 2        | 0        | 7.87598  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 80         | 3         | 4        | 0        | 20.5664  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 80         | 3         | 6        | 40       | 52.1741  | 17.3955   | 1.75    | 764825    | 854760    | 912818    | 299989       | 2.0414e+006  | 4.12966e+006 | 599579   | 687251   | 728981   |
| 90         | 3         | 2        | 0        | 7.95883  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 90         | 3         | 4        | 0        | 4.3065   | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 90         | 3         | 6        | 0        | 8.51488  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 100        | 3         | 2        | 0        | 9.54901  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 100        | 3         | 4        | 0        | 11.5775  | -         | 0       | -         | -         | -         | -            | -            | -            | -        | -        | -        |
| 100        | 3         | 6        | 20       | 24.6473  | 16.245    | 4       | 871085    | 890461    | 897358    | 1.26311e+006 | 2.07128e+006 | 2.98429e+006 | 728716   | 744955   | 751844   |

| Req. Stat. | Vehicule | % Reussi | Temps H.  | Temps RBS | %Temps Diff. | Cout H. | Cout RBS     | %Cout Diff. | Qos H.       | Qos RBS      | %Qos Diff. | Co2 H. | Co2 RBS      | %Co2 Diff. |
|------------|----------|----------|-----------|-----------|--------------|---------|--------------|-------------|--------------|--------------|------------|--------|--------------|------------|
| 10         | 2        | 60       | 0.073667  | 2.4073    | 3167.87      | 305056  | 283797       | -6.96884    | 3.72471e+006 | 243821       | -93.454    | 252166 | 202505       | -19.6939   |
| 10         | 4        | 90       | 0.0633333 | 6.12367   | 9568.95      | 292432  | 260439       | -10.9403    | 4.05545e+006 | 603400       | -85.1213   | 208488 | 189551       | -9.08309   |
| 10         | 6        | 100      | 0.0574    | 7.8458    | 13568.6      | 308123  | 274527       | -10.9033    | 4.23331e+006 | 103953       | -97.5444   | 171803 | 171164       | -0.372008  |
| 20         | 2        | 10       | 0.138     | 6.193     | 4387.68      | 388316  | 376082       | -3.15059    | 2.26742e+007 | 1.951e+006   | -91.3955   | 338678 | 287550       | -15.0964   |
| 20         | 4        | 100      | 0.1108    | 17.5968   | 15781.6      | 472238  | 381016       | -19.317     | 2.34766e+006 | 1.00705e+006 | -57.104    | 376349 | 289903       | -22.9696   |
| 20         | 6        | 100      | 0.1062    | 26.5654   | 24914.5      | 442629  | 378874       | -14.4036    | 1.47378e+006 | 772210       | -47.6036   | 297081 | 261398       | -12.0111   |
| 30         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 30         | 4        | 50       | 0.1634    | 18.1394   | 11001.2      | 561700  | 589166       | 4.88992     | 1.30855e+006 | 1.6503e+006  | 26.1163    | 447760 | 448777       | 0.227121   |
| 30         | 6        | 100      | 0.1474    | 40.5946   | 27440.4      | 551154  | 550031       | -0.20365    | 847276       | 621722       | -26.6211   | 411793 | 423468       | 2.83531    |
| 40         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 40         | 4        | 60       | 0.22067   | 39.084    | 17611.8      | 687511  | 714398       | 3.91074     | 2.33393e+006 | 3.59439e+006 | 54.0057    | 542872 | 568428       | 4.70752    |
| 40         | 6        | 100      | 0.2081    | 59.3238   | 28407.4      | 681344  | 721387       | 5.87709     | 369615       | 647585       | 75.205     | 545012 | 564985       | 3.6646     |
| 50         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 50         | 4        | 30       | 0.288333  | 56.304    | 19427.4      | 763700  | 835687       | 9.42601     | 4.22208e+006 | 1.62122e+006 | -61.6014   | 606738 | 670140       | 10.4496    |
| 50         | 6        | 90       | 0.254444  | 96.0646   | 37654.6      | 767945  | 837789       | 9.0949      | 960654       | 956477       | -0.434836  | 645265 | 665682       | 3.16417    |
| 60         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 60         | 4        | 10       | 0.354     | 101.859   | 28673.7      | 920223  | 957612       | 4.063       | 7.10857e+006 | 3.04526e+006 | -57.1607   | 710223 | 761847       | 7.26873    |
| 60         | 6        | 90       | 0.320444  | 160.676   | 50041.6      | 943673  | 1.02255e+006 | 8.35826     | 1.37933e+006 | 873325       | -36.6846   | 757307 | 812836       | 7.33233    |
| 70         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 70         | 4        | 10       | 0.445     | 72.125    | 16107.9      | 884820  | 1.00375e+006 | 13.4416     | 2.51019e+006 | 2.03405e+006 | -18.9681   | 713990 | 796715       | 11.5863    |
| 70         | 6        | 20       | 0.385     | 164.236   | 42558.7      | 933206  | 1.06239e+006 | 13.8432     | 1.4207e+006  | 2.80604e+006 | 97.5014    | 730094 | 844376       | 15.653     |
| 80         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 80         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 80         | 6        | 30       | 0.43667   | 168.101   | 38396.5      | 886388  | 1.14316e+006 | 28.9686     | 561438       | 3.54541e+006 | 531.488    | 715216 | 906269       | 26.7125    |
| 90         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 90         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 90         | 6        | 10       | 0.6       | 709.415   | 118136       | 895741  | 1.26039e+006 | 40.7097     | 1.09487e+006 | 9.93398e+006 | 807.322    | 716417 | 1.00207e+006 | 39.8719    |
| 100        | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 100        | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |
| 100        | 6        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -            | -          |

| Req. Stat. | Vehicule | % Reussi | Temps H.  | Temps RBS | %Temps Diff. | Cout H. | Cout RBS     | %Cout Diff. | Qos H.       | Qos RBS      | %Qos Diff. | Co2 H. | Co2 RBS | %Co2 Diff. |
|------------|----------|----------|-----------|-----------|--------------|---------|--------------|-------------|--------------|--------------|------------|--------|---------|------------|
| 10         | 2        | 50       | 0.0508    | 1.3758    | 2608.27      | 295519  | 276155       | -6.55242    | 5.81019e+006 | 3.72474e+006 | -35.893    | 241868 | 210113  | -13.1288   |
| 10         | 4        | 100      | 0.0482    | 3.0159    | 6157.05      | 282968  | 281392       | -0.557136   | 7.96255e+006 | 891917       | -88.7986   | 204014 | 221506  | 8.57362    |
| 10         | 6        | 100      | 0.0446    | 4.4073    | 9781.84      | 301086  | 272617       | -9.45548    | 6.70708e+006 | 1.0974e+006  | -83.6382   | 168741 | 172729  | 2.36314    |
| 20         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 20         | 4        | 90       | 0.0903333 | 9.901     | 10860.5      | 466786  | 391985       | -16.0247    | 5.40196e+006 | 624437       | -88.4405   | 364329 | 299749  | -17.7257   |
| 20         | 6        | 100      | 0.0864    | 14.2298   | 16369.7      | 443978  | 416768       | -6.12851    | 3.99538e+006 | 602712       | -84.9148   | 296211 | 299759  | 1.19792    |
| 30         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 30         | 4        | 60       | 0.1325    | 15.6573   | 11716.9      | 554165  | 554897       | 0.131936    | 2.16326e+006 | 912569       | -57.8151   | 435041 | 437012  | 0.453055   |
| 30         | 6        | 100      | 0.1189    | 20.0452   | 16758.9      | 535531  | 540218       | 0.875085    | 1.53525e+006 | 1.22401e+006 | -20.2733   | 400027 | 398795  | -0.307893  |
| 40         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 40         | 4        | 50       | 0.1892    | 25.955    | 13618.3      | 665471  | 674610       | 1.37342     | 2.09551e+006 | 2.13639e+006 | 1.95066    | 517608 | 524161  | 1.26603    |
| 40         | 6        | 100      | 0.1698    | 33.2386   | 19475.1      | 662928  | 697227       | 5.17385     | 379508       | 246243       | -35.1153   | 526567 | 540385  | 2.62433    |
| 50         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 50         | 4        | 40       | 0.2315    | 30.3483   | 13009.4      | 797724  | 818939       | 2.65932     | 4.42937e+006 | 2.95411e+006 | -33.3062   | 644506 | 650214  | 0.885764   |
| 50         | 6        | 70       | 0.230714  | 58.8753   | 25418.7      | 741879  | 796987       | 7.42814     | 1.13927e+006 | 457653       | -59.8294   | 628013 | 631411  | 0.541149   |
| 60         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 60         | 4        | 20       | 0.2745    | 68.7575   | 24948.3      | 865424  | 1.0406e+006  | 20.2417     | 4.74715e+006 | 6.50007e+006 | 36.9259    | 667033 | 842435  | 26.2959    |
| 60         | 6        | 90       | 0.272444  | 110.738   | 40546        | 933840  | 969448       | 3.81311     | 1.39627e+006 | 1.52129e+006 | 8.95447    | 754142 | 769609  | 2.05093    |
| 70         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 70         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 70         | 6        | 20       | 0.314     | 96.774    | 30719.7      | 928440  | 1.06837e+006 | 15.0717     | 328048       | 730282       | 122.614    | 714917 | 840032  | 17.5006    |
| 80         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 80         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 80         | 6        | 30       | 0.401333  | 118.146   | 29338.3      | 910934  | 1.07784e+006 | 18.3221     | 528840       | 1.17559e+006 | 122.296    | 720591 | 859887  | 19.3308    |
| 90         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 90         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 90         | 6        | 20       | 0.487     | 132.281   | 27062.3      | 967608  | 1.17874e+006 | 21.8202     | 4.10174e+006 | 2.61448e+006 | -36.2591   | 823380 | 980446  | 19.0757    |
| 100        | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 100        | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 100        | 6        | 10       | 0.575     | 128.283   | 22210.1      | 951496  | 1.14934e+006 | 20.7934     | 1.89224e+006 | 3.10542e+006 | 64.1132    | 782287 | 941392  | 20.3385    |

| Req. Stat. | Vehicule | %Reussi | Temps H.  | Temps RBS | %Temps Diff. | Cout H. | Cout RBS     | %Cout Diff. | Qos H.       | Qos RBS      | %Qos Diff. | Co2 H. | Co2 RBS | %Co2 Diff. |
|------------|----------|---------|-----------|-----------|--------------|---------|--------------|-------------|--------------|--------------|------------|--------|---------|------------|
| 10         | 2        | 40      | 0.04275   | 0.6575    | 1438.01      | 266968  | 237674       | -10.9729    | 1.1435e+007  | 3.985e+006   | -65.1508   | 226559 | 187924  | -17.0529   |
| 10         | 4        | 100     | 0.0414    | 1.4167    | 3321.98      | 256382  | 254081       | -0.89727    | 6.69223e+006 | 216378       | -96.7667   | 182718 | 195081  | 6.76617    |
| 10         | 6        | 100     | 0.0436    | 1.9407    | 4351.15      | 270745  | 249796       | -7.73733    | 5.54063e+006 | 209443       | -96.2199   | 150591 | 160082  | 6.30275    |
| 20         | 2        | 10      | 0.08      | 1.516     | 1795         | 257047  | 306743       | 19.3333     | 7.20509e+007 | 2.69114e+006 | -96.2649   | 269900 | 247342  | -8.35783   |
| 20         | 4        | 90      | 0.0734444 | 5.01722   | 6731.32      | 444743  | 408378       | -8.1767     | 3.69106e+006 | 1.13943e+006 | -69.1301   | 349228 | 311297  | -10.8613   |
| 20         | 6        | 90      | 0.0735556 | 9.60067   | 12952.3      | 415373  | 385184       | -7.26796    | 2.68633e+006 | 276762       | -89.6974   | 278909 | 273331  | -1.9997    |
| 30         | 2        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 30         | 4        | 80      | 0.113125  | 6.40275   | 5559.89      | 505565  | 514497       | 1.76658     | 873319       | 288683       | -66.9441   | 392936 | 398671  | 1.4596     |
| 30         | 6        | 100     | 0.1112    | 9.0899    | 8074.37      | 499195  | 507342       | 1.63201     | 702772       | 147215       | -79.0523   | 366451 | 367537  | 0.296199   |
| 40         | 2        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 40         | 4        | 40      | 0.175     | 13.375    | 8685.71      | 634600  | 638194       | 0.566376    | 3.60613e+006 | 2.7454e+006  | -23.8683   | 498539 | 496569  | -0.395117  |
| 40         | 6        | 90      | 0.151556  | 19.9693   | 13076.2      | 639593  | 657821       | 2.84993     | 540268       | 801141       | 48.286     | 504404 | 515628  | 2.22522    |
| 50         | 2        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 50         | 4        | 20      | 0.209     | 17.33     | 8191.87      | 772727  | 755302       | -2.25504    | 3.6788e+006  | 3.15871e+006 | -14.1375   | 603630 | 606128  | 0.413902   |
| 50         | 6        | 50      | 0.187     | 44.2876   | 23583.2      | 785799  | 806693       | 2.65889     | 1.24684e+006 | 539818       | -56.7052   | 632798 | 637853  | 0.798873   |
| 60         | 2        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 60         | 4        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 60         | 6        | 60      | 0.258667  | 69.0087   | 26578.6      | 899070  | 921796       | 2.52772     | 2.09193e+006 | 1.03966e+006 | -50.3014   | 751365 | 746282  | -0.676541  |
| 70         | 2        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 70         | 4        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 70         | 6        | 30      | 0.278667  | 79.479    | 28421.2      | 926008  | 1.00319e+006 | 8.33506     | 498200       | 2.10216e+006 | 321.952    | 726184 | 784195  | 7.98855    |
| 80         | 2        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 80         | 4        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 80         | 6        | 20      | 0.373     | 71.6365   | 19105.5      | 848335  | 952844       | 12.3192     | 529149       | 1.40511e+006 | 165.542    | 695027 | 779512  | 12.1557    |
| 90         | 2        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 90         | 4        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 90         | 6        | 20      | 0.431     | 150.103   | 34726.8      | 866314  | 1.07191e+006 | 23.7324     | 656338       | 1.33946e+006 | 104.081    | 690503 | 846987  | 22.6624    |
| 100        | 2        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 100        | 4        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 100        | 6        | 0       | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |

## Comparaison heuristique – RBS pour T=600

| Req. Stat. | Vehicule | % Reussi | Temps H.  | Temps RBS | %Temps Diff. | Cout H. | Cout RBS     | %Cout Diff. | Qos H.       | Qos RBS      | %Qos Diff. | Co2 H. | Co2 RBS | %Co2 Diff. |
|------------|----------|----------|-----------|-----------|--------------|---------|--------------|-------------|--------------|--------------|------------|--------|---------|------------|
| 10         | 2        | 30       | 0.037     | 0.293667  | 693.694      | 206618  | 209028       | 1.16662     | 1.6929e+006  | 1.30295e+006 | -23.0344   | 178101 | 171191  | -3.87968   |
| 10         | 4        | 100      | 0.0378    | 0.6655    | 1660.58      | 231867  | 230092       | -0.765579   | 4.63963e+006 | 372404       | -91.9734   | 161887 | 171830  | 6.14183    |
| 10         | 6        | 100      | 0.0375    | 0.8326    | 2120.27      | 236679  | 232318       | -1.84276    | 3.53041e+006 | 147558       | -95.8204   | 132243 | 142315  | 7.61661    |
| 20         | 2        | 20       | 0.0915    | 2.0765    | 2169.4       | 347893  | 346467       | -0.409919   | 2.13346e+007 | 2.13937e+006 | -89.9723   | 329514 | 286436  | -13.0733   |
| 20         | 4        | 70       | 0.0661429 | 1.70586   | 2479.05      | 409831  | 396151       | -3.33785    | 8.28374e+006 | 3.95935e+006 | -52.2033   | 322199 | 308354  | -4.29704   |
| 20         | 6        | 60       | 0.0751667 | 2.37333   | 3057.43      | 379332  | 384535       | 1.37168     | 1.42866e+006 | 147886       | -89.6486   | 259961 | 257773  | -0.841839  |
| 30         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 30         | 4        | 90       | 0.100444  | 3.14056   | 3026.66      | 467967  | 478416       | 2.23286     | 849781       | 403477       | -52.5199   | 367684 | 373321  | 1.53332    |
| 30         | 6        | 100      | 0.1039    | 4.0057    | 3755.34      | 470199  | 478919       | 1.85444     | 418189       | 126521       | -69.7455   | 344430 | 351119  | 1.94224    |
| 40         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 40         | 4        | 60       | 0.144     | 6.04767   | 4099.77      | 580438  | 583383       | 0.507461    | 1.21312e+006 | 1.12545e+006 | -7.22719   | 451924 | 449808  | -0.46812   |
| 40         | 6        | 100      | 0.1344    | 7.3277    | 5352.16      | 613927  | 627089       | 2.14395     | 274936       | 109766       | -60.0757   | 484433 | 495848  | 2.35631    |
| 50         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 50         | 4        | 30       | 0.185     | 16.935    | 9054.05      | 700510  | 727365       | 3.83371     | 3.52954e+006 | 1.7784e+006  | -49.6138   | 564680 | 583017  | 3.24733    |
| 50         | 6        | 50       | 0.1812    | 19.2988   | 10550.6      | 730924  | 748555       | 2.41214     | 494195       | 368447       | -25.4449   | 581765 | 585805  | 0.694364   |
| 60         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 60         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 60         | 6        | 40       | 0.21075   | 32.3128   | 15232.3      | 888920  | 898465       | 1.07379     | 320816       | 890204       | 177.481    | 698247 | 717939  | 2.82028    |
| 70         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 70         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 70         | 6        | 30       | 0.250333  | 53.4053   | 21233.7      | 844126  | 881085       | 4.37836     | 160959       | 148873       | -7.50895   | 684677 | 714046  | 4.28948    |
| 80         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 80         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 80         | 6        | 30       | 0.308667  | 81.009    | 26144.8      | 828430  | 885870       | 6.93362     | 405308       | 561612       | 38.5644    | 657172 | 712677  | 8.44604    |
| 90         | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 90         | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 90         | 6        | 20       | 0.3395    | 40.099    | 11711.2      | 876090  | 999308       | 14.0645     | 162850       | 5.00539e+006 | 2973.62    | 688412 | 792447  | 15.1123    |
| 100        | 2        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 100        | 4        | 0        | -         | -         | -            | -       | -            | -           | -            | -            | -          | -      | -       | -          |
| 100        | 6        | 10       | 0.376     | 49.616    | 13095.7      | 872429  | 1.03024e+006 | 18.0882     | 296388       | 1.0521e+006  | 254.973    | 716461 | 857078  | 19.6266    |

## Comparaison heuristique – RBS pour T=800

| Req. Stat. | Vehicule | % Reussi | Temps H.  | Temps RBS | %Temps Diff. | Cout H. | Cout RBS | %Cout Diff. | Qos H.       | Qos RBS      | %Qos Diff. | Co2 H. | Co2 RBS | %Co2 Diff. |
|------------|----------|----------|-----------|-----------|--------------|---------|----------|-------------|--------------|--------------|------------|--------|---------|------------|
| 10         | 2        | 60       | 0.0308333 | 0.0495    | 60.5405      | 183459  | 199719   | 8.86328     | 1.33653e+006 | 1.46355e+006 | 9.50388    | 156654 | 160309  | 2.3331     |
| 10         | 4        | 90       | 0.0368889 | 0.085889  | 132.831      | 200824  | 201902   | 0.536933    | 991641       | 374384       | -62.2461   | 142952 | 147067  | 2.87904    |
| 10         | 6        | 90       | 0.0426667 | 0.114     | 167.187      | 214101  | 214038   | -0.0295916  | 658059       | 335451       | -49.0242   | 117747 | 127705  | 8.4568     |
| 20         | 2        | 30       | 0.0753333 | 0.504667  | 569.912      | 326601  | 327215   | 0.188147    | 7.79346e+006 | 4.62618e+006 | -40.6402   | 289982 | 280554  | -3.25138   |
| 20         | 4        | 80       | 0.06275   | 0.199625  | 218.127      | 383476  | 381853   | -0.423249   | 5.0175e+006  | 3.08755e+006 | -38.4645   | 300862 | 297264  | -1.19584   |
| 20         | 6        | 70       | 0.065     | 0.353429  | 443.736      | 376749  | 374240   | -0.66591    | 2.02406e+006 | 859125       | -57.5545   | 245242 | 247999  | 1.12441    |
| 30         | 2        | 20       | 0.185     | 0.678     | 266.486      | 409094  | 408309   | -0.19175    | 2.70521e+006 | 1.25801e+006 | -53.4969   | 314945 | 307554  | -2.34678   |
| 30         | 4        | 90       | 0.0987778 | 0.624     | 531.721      | 436972  | 453788   | 3.84809     | 2.36358e+006 | 1.27054e+006 | -46.2451   | 338542 | 351287  | 3.76471    |
| 30         | 6        | 90       | 0.0936667 | 0.517     | 451.957      | 449413  | 457870   | 1.88186     | 1.69065e+006 | 725975       | -57.0594   | 319996 | 330993  | 3.43647    |
| 40         | 2        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 40         | 4        | 70       | 0.132     | 1.86714   | 1314.5       | 563829  | 571764   | 1.40728     | 466104       | 242296       | -48.0169   | 435727 | 442003  | 1.44034    |
| 40         | 6        | 90       | 0.133889  | 1.41033   | 953.361      | 579039  | 585344   | 1.08878     | 209259       | 63548.8      | -69.6315   | 452847 | 455669  | 0.623148   |
| 50         | 2        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 50         | 4        | 30       | 0.167     | 2.98367   | 1686.63      | 662184  | 671670   | 1.43268     | 119868       | 2.23654e+006 | 1765.83    | 514839 | 522150  | 1.42008    |
| 50         | 6        | 70       | 0.154     | 3.73386   | 2324.58      | 699907  | 708463   | 1.22255     | 425335       | 45419.1      | -89.3216   | 561400 | 565730  | 0.771175   |
| 60         | 2        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 60         | 4        | 30       | 0.231667  | 6.15667   | 2557.55      | 775375  | 773055   | -0.299197   | 149188       | 1.18014e+006 | 691.043    | 600718 | 620440  | 3.28301    |
| 60         | 6        | 70       | 0.206429  | 10.1946   | 4838.55      | 836460  | 842948   | 0.775666    | 185689       | 378712       | 103.949    | 678635 | 683984  | 0.788089   |
| 70         | 2        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 70         | 4        | 10       | 0.243     | 41.787    | 17096.3      | 732011  | 745158   | 1.79601     | 427374       | 272856       | -36.1551   | 587083 | 614973  | 4.75046    |
| 70         | 6        | 30       | 0.253667  | 22.463    | 8755.32      | 768959  | 802021   | 4.29952     | 26051.7      | 1.19293e+006 | 4479.09    | 629650 | 649557  | 3.16166    |
| 80         | 2        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 80         | 4        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 80         | 6        | 40       | 0.2775    | 17.3955   | 6168.65      | 830663  | 839716   | 1.08988     | 391559       | 1.87849e+006 | 379.747    | 657237 | 670012  | 1.94384    |
| 90         | 2        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 90         | 4        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 90         | 6        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 100        | 2        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 100        | 4        | 0        | -         | -         | -            | -       | -        | -           | -            | -            | -          | -      | -       | -          |
| 100        | 6        | 20       | 0.3615    | 16.245    | 4393.78      | 858134  | 890224   | 3.73947     | 564526       | 2.00108e+006 | 254.471    | 724686 | 748764  | 3.32259    |

| Req. Stat. | Vehicule | % Reussi | Temps NSGAI. | Temps H. | %Temps Diff. | Cout NSGAI. | Cout H. | %Cout Diff. | Co2 NSGAI. | Co2 RBS | %Co2 H.   |
|------------|----------|----------|--------------|----------|--------------|-------------|---------|-------------|------------|---------|-----------|
| 10         | 2        | 100      | 335.335      | 0.0797   | -99.9762     | 193204      | 278645  | 44.2227     | 177034     | 230628  | 30.2729   |
| 10         | 4        | 100      | 382.327      | 0.0637   | -99.9833     | 251813      | 285793  | 13.4939     | 218511     | 207014  | -5.26144  |
| 10         | 6        | 100      | 403.808      | 0.0574   | -99.9858     | 258010      | 308123  | 19.4229     | 179796     | 171803  | -4.4455   |
| 20         | 2        | 100      | 455.058      | 0.1161   | -99.9745     | 387175      | 356460  | -7.9331     | 290539     | 309844  | 6.64448   |
| 20         | 4        | 100      | 469.429      | 0.1108   | -99.9764     | 440679      | 472238  | 7.16141     | 340965     | 376349  | 10.3778   |
| 20         | 6        | 100      | 487.666      | 0.1062   | -99.9782     | 432128      | 442629  | 2.42994     | 306820     | 297081  | -3.17427  |
| 30         | 2        | 100      | 574.076      | 0.1963   | -99.9658     | 452282      | 549818  | 21.5651     | 343055     | 426788  | 24.408    |
| 30         | 4        | 100      | 566.562      | 0.1554   | -99.9726     | 518266      | 558354  | 7.73503     | 396003     | 444633  | 12.2801   |
| 30         | 6        | 100      | 579.464      | 0.1474   | -99.9746     | 561216      | 551154  | -1.79289    | 420860     | 411793  | -2.15436  |
| 40         | 2        | 100      | 707.884      | 0.298    | -99.9579     | 538719      | 609963  | 13.2248     | 403224     | 483097  | 19.8085   |
| 40         | 4        | 100      | 669.606      | 0.2182   | -99.9674     | 614304      | 692832  | 12.7832     | 475384     | 546260  | 14.9092   |
| 40         | 6        | 100      | 670.284      | 0.2081   | -99.969      | 681169      | 681344  | 0.0257234   | 537127     | 545012  | 1.46812   |
| 50         | 2        | 100      | 832.235      | 0.4153   | -99.9501     | 624257      | 696846  | 11.628      | 454110     | 529893  | 16.6883   |
| 50         | 4        | 100      | 765.431      | 0.2855   | -99.9627     | 713604      | 759148  | 6.38217     | 555191     | 610142  | 9.89769   |
| 50         | 6        | 100      | 758.034      | 0.2503   | -99.967      | 784244      | 779844  | -0.561102   | 611053     | 647081  | 5.89615   |
| 60         | 2        | 100      | 1020.25      | 0.5868   | -99.9425     | 687698      | 791048  | 15.0285     | 507807     | 597894  | 17.7404   |
| 60         | 4        | 100      | 898.298      | 0.3513   | -99.9609     | 779577      | 862694  | 10.6618     | 606195     | 697122  | 14.9996   |
| 60         | 6        | 100      | 864.769      | 0.3219   | -99.9628     | 963231      | 934019  | -3.03273    | 763191     | 758973  | -0.552658 |
| 70         | 2        | 100      | 1140.01      | 0.7286   | -99.9361     | 674563      | 807255  | 19.6708     | 481717     | 614093  | 27.4801   |
| 70         | 4        | 100      | 998.288      | 0.4778   | -99.9521     | 790981      | 936318  | 18.3743     | 615378     | 749481  | 21.7918   |
| 70         | 6        | 100      | 933.56       | 0.3643   | -99.961      | 939480      | 913782  | -2.73538    | 738567     | 758156  | 2.6523    |
| 80         | 2        | 100      | 1335.19      | 0.8121   | -99.9392     | 675374      | 764500  | 13.1966     | 484876     | 571681  | 17.9026   |
| 80         | 4        | 100      | 1128.58      | 0.578    | -99.9488     | 820516      | 866474  | 5.60118     | 633781     | 690406  | 8.93458   |
| 80         | 6        | 100      | 1033.51      | 0.4622   | -99.9553     | 917152      | 881285  | -3.91062    | 711985     | 726243  | 2.00264   |
| 90         | 2        | 100      | 1545.66      | 1.1942   | -99.9227     | 720586      | 770589  | 6.93925     | 516457     | 596998  | 15.5948   |
| 90         | 4        | 100      | 1198.43      | 0.612    | -99.9489     | 800706      | 875014  | 9.2804      | 626990     | 693239  | 10.5662   |
| 90         | 6        | 100      | 1114.56      | 0.5048   | -99.9547     | 955722      | 854531  | -10.5879    | 765976     | 732872  | -4.32186  |
| 100        | 2        | 100      | 1749.61      | 1.0826   | -99.9381     | 736390      | 845929  | 14.8751     | 535605     | 624765  | 16.6465   |
| 100        | 4        | 100      | 1323.3       | 0.7647   | -99.9422     | 791795      | 890620  | 12.4811     | 616885     | 713350  | 15.6374   |
| 100        | 6        | 100      | 1193.41      | 0.6241   | -99.9477     | 870960      | 868757  | -0.252937   | 695934     | 726090  | 4.33313   |

| Req. Stat. | Vehicle | % Reussi | Temps NSGAIL. | Temps RBS | %Temps Diff. | Cout NSGAIL. | Cout RBS     | %Cout Diff. | Co2 NSGAIL.  | Co2 RBS      | %Co2 Diff. |
|------------|---------|----------|---------------|-----------|--------------|--------------|--------------|-------------|--------------|--------------|------------|
| 10         | 2       | 60       | 322.164       | 2.40733   | -99.2528     | 177745       | 283797       | 59.6653     | 1.18302e+006 | 1.41753e+006 | 19.8229    |
| 10         | 4       | 90       | 381.802       | 6.12367   | -98.3961     | 254086       | 260439       | 2.50068     | 2.20634e+006 | 1.89551e+006 | -14.088    |
| 10         | 6       | 100      | 403.808       | 7.8458    | -98.057      | 258010       | 274527       | 6.40187     | 1.97775e+006 | 1.8828e+006  | -4.80097   |
| 20         | 2       | 10       | 451.386       | 6.193     | -98.628      | 366148       | 376082       | 2.71289     | 554198       | 575099       | 3.77143    |
| 20         | 4       | 100      | 469.429       | 17.5968   | -96.2514     | 440679       | 381016       | -13.5389    | 3.75061e+006 | 3.18894e+006 | -14.9756   |
| 20         | 6       | 100      | 487.666       | 26.5654   | -94.5525     | 432128       | 378674       | -12.3237    | 3.37502e+006 | 2.87538e+006 | -14.8041   |
| 30         | 2       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 30         | 4       | 50       | 562.167       | 18.1394   | -96.7733     | 521908       | 589166       | 12.887      | 2.39992e+006 | 2.69266e+006 | 12.1981    |
| 30         | 6       | 100      | 579.464       | 40.5946   | -92.9945     | 561216       | 550031       | -1.99289    | 4.62946e+006 | 4.65815e+006 | 0.619866   |
| 40         | 2       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 40         | 4       | 60       | 675.428       | 39.084    | -94.2134     | 603018       | 714398       | 18.4704     | 3.24244e+006 | 3.97899e+006 | 22.7161    |
| 40         | 6       | 100      | 670.284       | 59.3238   | -91.1495     | 681169       | 721387       | 5.90433     | 5.90839e+006 | 6.21483e+006 | 5.18652    |
| 50         | 2       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 50         | 4       | 30       | 770.156       | 56.304    | -92.6893     | 736972       | 835687       | 13.3946     | 2.31536e+006 | 2.68056e+006 | 15.7729    |
| 50         | 6       | 90       | 759.062       | 96.0646   | -87.3443     | 773463       | 837789       | 8.31659     | 6.02845e+006 | 6.65682e+006 | 10.4234    |
| 60         | 2       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 60         | 4       | 10       | 879.06        | 101.859   | -88.4127     | 777451       | 957612       | 23.1732     | 1.15262e+006 | 1.52369e+006 | 32.1945    |
| 60         | 6       | 90       | 865.999       | 160.676   | -81.4462     | 969619       | 1.02255e+006 | 5.45861     | 7.72926e+006 | 8.12836e+006 | 5.16341    |
| 70         | 2       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 70         | 4       | 10       | 958.354       | 72.125    | -92.4741     | 792327       | 1.00375e+006 | 26.6843     | 1.19274e+006 | 1.59343e+006 | 33.5945    |
| 70         | 6       | 20       | 934.659       | 164.236   | -82.4282     | 999030       | 1.06239e+006 | 6.34222     | 2.40319e+006 | 2.53313e+006 | 5.40688    |
| 80         | 2       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 80         | 4       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 80         | 6       | 30       | 1024.37       | 168.101   | -83.5897     | 887654       | 1.14316e+006 | 28.7846     | 2.69889e+006 | 3.62507e+006 | 34.3172    |
| 90         | 2       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 90         | 4       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 90         | 6       | 10       | 1133.14       | 709.415   | -37.3939     | 906374       | 1.26039e+006 | 39.059      | 1.52216e+006 | 2.00413e+006 | 31.6636    |
| 100        | 2       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 100        | 4       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |
| 100        | 6       | 0        | -             | -         | -            | -            | -            | -           | -            | -            | -          |

# Rapport de Projet Fin d'Etude 2011/2012

---

## Problème dynamique de transport à la demande

### Résumé :

Le transport à la demande est un mode de transport en commun qui est en plein développement aujourd'hui. Il vise à s'adapter aux besoins de la population en proposant aux utilisateurs de choisir les horaires et les points de passage des véhicules. Dans ce rapport, nous présentons deux méthodes pour résoudre un problème de transport à la demande dynamique de manière heuristique. La première méthode est un algorithme de liste par insertion, la deuxième est une méthode arborescente de recherche par faisceau.

**Mots clefs** : transport à la demande, TAD dynamique, heuristique, insertion, recherche par faisceau, optimisation, multicritères.

### Abstract :

The interest for transport on demand has increased during last years. It aims to adapt the offer to population needs by proposing customers to choose "when and where" they want a vehicle. In this paper, we present two algorithms to resolve this problem of multi objectives optimisation. The first method is an heuristic by insertion, the second is a Recovering Beam Search.

**Keywords** : transport on demand, DTAD, heuristic, insertion, beam search, optimization, multi objectives.

### Etudiant :

Fabien Touchard

[fabien.touchard@etu.univ-tours.fr](mailto:fabien.touchard@etu.univ-tours.fr)

### Encadrants :

Ahmed Atahran

[ahmed.atahram@etu.univ-tours.fr](mailto:ahmed.atahram@etu.univ-tours.fr)

Christophe Lente

[christophe.lente@univ-tours.fr](mailto:christophe.lente@univ-tours.fr)

Vincent T'Kindt

[vincent.tkindt@univ-tours.fr](mailto:vincent.tkindt@univ-tours.fr)

