

Rapport de projet tutoré de 5ème année

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

Victor MAINTENANT

Victoria SCHWINDENHAMMER

Séraphie MAURY

Client : Dimitri DUVAL

Tuteur école : Hervé PINGAUD

Remerciements

Nous souhaitons tout d'abord remercier vivement M.PINGAUD pour toute l'aide qu'il nous a apportée sur ce projet, pour le temps qu'il nous a consacré, pour son expertise et ses conseils.

De la même façon nous remercions Mme.MEGDICHE pour son implication dans le projet et son expertise sur le sujet d'étude.

Nous adressons également nos remerciements à l'entreprise Atout Majeur Concept et en particulier à M.DUVAL sans qui nous n'aurions pu travailler sur ce projet. Nous remercions également Adrien MARCOS et Damien DE QUINA qui, avec M.DUVAL, nous ont fourni tous les documents nécessaires pour mener à bien ce projet et se sont rendus disponibles tout du long.

Enfin nous voulions remercier M.ZHANG, qui a pris le temps de nous répondre pour nous transmettre un peu de sa connaissance sur l'outil OptaPlanner.

Résumé

Dans le cadre de notre projet tutoré de 5ème année à ISIS nous avons choisi le projet proposé par l'entreprise Atout Majeur Concept qui est la recherche opérationnelle pour l'optimisation des flux de brancardiers. Cette dernière est séparée en deux étapes, la prise de connaissances des solutions déjà existantes chez AMC, soit la méthode heuristique, qui est actuellement la solution utilisée par l'entreprise et la méthode exacte, qui a été déployée en stage l'année dernière par deux étudiants d'ISIS et la création d'une solution métaheuristique. La deuxième étape étant l'amélioration de la méthode exacte et la comparaison des méthodes heuristique, métaheuristique et exacte concernant leurs performances (rapidité, précision). Cette comparaison a pour but de permettre à AMC de choisir la solution la plus adaptée au problème de gestion des flux des brancardiers afin de la mettre en œuvre chez eux pour proposer à leurs clients la meilleure alternative possible.

Abstract

In the framework of our 5th year tutored project at ISIS, we chose the project proposed by the company Atout Majeur Concept which is the operational research for the optimization of stretcher-bearer flows. The latter is separated in two steps, the knowledge of the already existing solutions at AMC, that is to say the heuristic method, which is currently the solution used by the company and the exact method, which was deployed in internship last year by two ISIS students. The second step is the improvement of the exact method and the comparison of the heuristic, metaheuristic and exact methods regarding their performances (speed, accuracy). This comparison aims at allowing AMC to choose the most adapted solution to the stretcher-bearer flow management problem in order to implement it at the company to propose to their customers the best possible alternative.

Table des matières

Remerciements.....	2
Résumé	3
Table des matières	4
Table des figures	5
Glossaire.....	6
Introduction	7
1. Contexte.....	7
2. Objectif	7
Présentation du sujet	8
1. Brancardage.....	8
2. Vehicle Routing Problem	8
3. Méthodes d'optimisation.....	10
Développement.....	11
1. Principe et Hypothèse	11
2. Méthode approchée : Heuristique.....	12
3. Méthode exacte : Programmation linéaire	14
A. Présentation de l'existant.....	14
B. Améliorations apportées	14
4. Méthode approchée : Méta-heuristique	15
A. Présentation d'OptaPlanner	15
B. Solution envisagée et problèmes rencontrés.....	18
5. Indicateurs.....	19
A. Construction des indicateurs	19
B. Modélisation du processus de brancardage.....	22
6. Comparaisons	22
A. Jeux de données	22
B. Comparaison des modèles	27
7. Solution Optimale	29
Gestion de projet.....	30
Conclusion	32
Bibliographie	33
Annexes.....	34

Table des figures

Figure 1. Tableau des différents types de VRP	9
Figure 2. Schéma de composition des missions.....	11
Figure 3. Exemple d'enchaînement des missions	12
Figure 4. Schéma du processus de récupération des données des missions.....	13
Figure 5. Ordre de priorité des services	13
Figure 6. Schéma du processus d'affectation des brancardiers aux missions	13
Figure 7. Interface d'OptaPlanner version "Quickstarts"	16
Figure 8. Interface d'OptaPlanner version "Examples"	17
Figure 9. Comparaison des arborescences des versions d'exécution "Examples" (gauche) et "Quickstarts" (droite) d'OptaPlanner	17
Figure 10. Exemple de résolution d'un VRP sur Optaplanner.....	18
Figure 11. Tableau des criticités en fonction des situations.....	20
Figure 12. Cartographie du processus de tournée de brancardage.....	22
Figure 13. Enchaînement des missions dans le premier jeu de données	23
Figure 14. Test des indicateurs sur le premier jeu de données fictif	23
Figure 15. Enchaînement des missions dans le deuxième jeu de données.....	24
Figure 16. Test des indicateurs sur le deuxième jeu de données fictif.....	24
Figure 17. Enchaînement des missions dans le quatrième jeu de données	25
Figure 18. Test des indicateurs sur un jeu de données réel	25
Figure 19. Enchaînement des missions dans le troisième jeu de données.....	26
Figure 20. Test de valeurs d'indicateurs supérieures aux seuils.....	26
Figure 21. Tableau des résultats de la méthode exacte	27
Figure 22. Tableau des résultats de la méthode heuristique	29
Figure 23. Diagramme de Gantt initial	30
Figure 24. Diagramme de Gantt final	31

Glossaire

AMC : Atout Majeur Concept

CPLEX : outil informatique d'optimisation commercialisé par IBM. Son nom fait référence au langage C et à l'algorithme du simplexe.

Programmation linéaire (PL) : sont des problèmes d'optimisation où la fonction objective et les contraintes sont toutes linéaires.

Mathématiques heuristiques : En algorithmique, une heuristique est une méthode de calcul qui fournit rapidement une solution réalisable, pas nécessairement optimale ou exacte, pour un problème d'optimisation difficile.

IBM CPLEX ILOG Studio : IDE développé par IBM, propre à la Programmation linéaire.

Modèle mathématique : traduction d'une observation dans le but de lui appliquer les outils, les techniques et les théories mathématiques, puis généralement, en sens inverse, la traduction des résultats mathématiques obtenus en prédictions ou opérations dans le monde réel.

Intelligence artificielle : l'ensemble des théories et des techniques mises en œuvre en vue de réaliser des machines capables de simuler l'intelligence humaine

Vehicles Routing Problem (VRP) : Les problèmes de tournées de véhicules, est une classe de problèmes de recherche opérationnelle et d'optimisation combinatoire.

Vehicles Routing Problem with Soft Time Window (VRPSTW) : est un VRP tourné autour des fenêtres de temps d'action, ici défini comme souples, qui peut donc accepter des retard et sur lesquelles il est possible d'intervenir.

Optimisation : Raisonnement ou calcul permettant de trouver les valeurs d'un ou plusieurs paramètres qui correspondent au maximum d'une fonction.

Introduction

1. Contexte

AMC fait une percée remarquée dans le monde de l'édition logicielle en santé avec ses outils de gestion de ressources implantées dans le module SESAME. Un facteur clé du succès est lié à l'architecture de ces solutions gravitant autour du parcours patient à l'hôpital. Riche de ce patrimoine de données, et entretenant des relations privilégiées avec certains établissements hospitaliers innovants, AMC soutient une activité de recherche en optimisation de la gestion des ressources et, plus spécifiquement dans notre cas, pour les flux des brancardiers au sein des établissements de santé.

2. Objectif

Le projet "Recherche opérationnelle d'optimisation pour l'optimisation des flux de brancardiers à l'hôpital" a pour objectif de comparer plusieurs modèles, méthodes, démarches et outils de recherche opérationnelle en optimisation. La comparaison se focalise principalement sur les trois méthodes heuristique, métaheuristique et exacte, pour l'optimisation des flux de brancardiers.

Il s'agit de poursuivre le développement des outils de gestion optimisée des tournées de brancard qui a déjà été fait par des étudiants stagiaires d'ISIS. Le travail devra faire le point sur les limites des modèles développés par les travaux de stage par rapport à l'étendue des besoins. Une autre facette du travail sera de montrer, de manière pédagogique, les capacités de résolution de trois approches méthodologiques en recherche opérationnelle : par heuristique avec le programme fourni par AMC, par méta-heuristique via l'outil OptaPlanner, et exacte en programmation linéaire.

Cette comparaison permettra au client de choisir quelle méthode donne les meilleurs résultats en un temps minimum (inférieur à une dizaine de minute). L'optimisation choisie devra minimiser une fonction objectif choisi qui poursuit deux critères : le temps d'attente pour le patient et l'équité du temps de travail des brancardiers.

Actuellement c'est la méthode heuristique qui est utilisée par le client. La méthode exacte a été étudiée en première instance, mais a une durée d'exécution trop longue en temps de calcul pour des problèmes de taille réelle. Enfin la méthode métaheuristique n'a pas encore été initiée. La comparaison des trois méthodes sous l'hypothèse de modèles équivalents dans la mise en œuvre de chacune permettra, soit de conforter le client dans l'utilisation de la méthode heuristique et d'éclairer la pertinence des règles choisies tout comme l'algorithme qui les déploie, soit de changer de politique en poussant le travail de R&D pour lui proposer une solution satisfaisante. Dans le premier cas, en prouvant qu'elle est la meilleure en tenant compte du temps d'exécution et de la fiabilité de l'optimisation, dans le second en prouvant qu'une des 2 autres méthodes surpasse la méthode déjà en place en bénéfice de temps et/ou d'optimisation.

Présentation du sujet

1. Brancardage

Le sujet portant sur les flux de brancardiers, il semble important de rappeler ce qu'est le brancardage et quels en sont les enjeux. Le brancardage est l'un des éléments clés de maîtrise des flux de patients et des phénomènes d'engorgement dans les établissements de santé.

Le brancardage recouvre des transports de natures différentes : transport interne dans le service (transports entre box et salles d'attente), transport en lien avec les plateaux médico-techniques (bloc opératoire, imagerie, ...), transport vers les services cliniques d'hospitalisation.

C'est l'un des domaines les plus importants dans la prise en charge des patients dans les établissements de santé, car il possède un impact direct sur les soins fournis aux patients.

L'acte de brancardage est composé de plusieurs éléments¹ :

- Le transport des lits, brancard et autres matériels jusqu'aux bons patients
- La prise en charge de ces derniers dans un temps rapide mais sans jamais les brusquer afin d'éviter tout accident quel qu'il soit
- Le transport soigneux des patients jusqu'au bon service
- Le dépôt des patients.

Malgré son importance, le brancardage est aussi un acte à part des autres services. En effet, il est à la frontière de tout service et peut se retrouver exclu des études de services dans leur fonctionnement, alors qu'il en est le moyen d'entrée et de sortie dans la plupart des cas. Faire le choix de ne pas négliger la gestion et l'optimisation des flux de brancardages c'est donc s'assurer un grand impact positif, car en réduisant les retards et les erreurs c'est la qualité des soins du patient qui en est indirectement améliorée.

On parle de "flux de brancardiers" pour décrire en fait des tournées de brancardages au sein d'un ou plusieurs établissements. Dans le cadre de notre projet nous avons appréhender des jeux de données modélisant ces tournées possédantes chacune un nombre de brancardiers par jour avec respectivement leurs horaires de début, de fin de journée et de pause déjeuner, les durées et les horaires de début de chaque mission et le temps de déplacement d'une mission à une autre. Un flux de brancardier typique ce sont donc tous les mouvements qu'effectue chaque brancardier de leur début à leur fin de journée pour réaliser plusieurs missions qui leurs ont été attribuées.

2. Vehicle Routing Problem

Le Vehicle Routing Problem, VRP, est une classe de problèmes de recherche opérationnelle et d'optimisation combinatoire. Il s'agit de déterminer les tournées d'une flotte de véhicules

¹ Bouabdallah, M.N., Rached, M., Fondreville, J., Bahroun, Z., (2013). "Organization and management of hospital patient transportation system".

afin de livrer une liste de clients, ou de visites (visites médicales, commerciales, etc.). Ce genre de problème se rapproche fortement du nôtre car nous souhaitons optimiser le flux des brancardiers et déterminer les meilleurs chemins à emprunter.

Le but est de minimiser le coût de livraison des biens, d'optimiser le temps et de définir des plannings. Il est issu d'une version étendue du problème du voyageur de commerce (TSP, Travel Salesman Problem).

Il existe plusieurs formes de VRP, dont voici une liste non exhaustive :

- VRP, Vehicle Routing Problem
- CVRP, Capacited Vehicle Routing Problem
- VRPTW, Vehicle Routing Problem with Time Windows
- PVRP, Period Vehicle Routing Problem
- IRP, Inventory Routing Problem
- M-VRPTW, M-Vehicle Routing Problem with Time Windows

Toutes ces formes possèdent des différences et ne sont donc pas utilisables dans toutes les situations.

Modèles	Différence avec VRP
CVRP	Ajout d'une capacité associée à un véhicule
VRPTW	Sert dans une fenêtre de temps d'action définie Assure la cohérence temporelle des tournées
PVRP	Ajout d'une fréquence de visite d'un client
IRP	Intègre des problématiques de gestion de stock
M-VRPTW	Intègre une limite de dans le nombre de véhicules

Figure 1. Tableau des différents types de VRP

Ces différences sont définies en ajoutant des contraintes et variables supplémentaires à la fonction de base, qu'est celle de VRP. Par la suite, après avoir bien défini le problème et le cadre de l'utilisation de la fonction, certaines contraintes viennent s'ajouter sur les contraintes du modèle choisi, comme des contraintes de préemption, de précedence, de synchronisation, etc.

3. Méthodes d'optimisation²

En ce qui concerne l'optimisation combinatoire, il y a différentes méthodes de résolutions que l'on peut classer en 2 grandes familles : les méthodes exactes et les méthodes approchées.

Les méthodes exactes, appelées aussi complètes, produisent une solution optimale pour une instance de problème d'optimisation donnée. Elles se reposent généralement sur la recherche arborescente et sur l'énumération partielle de l'espace de solutions. Elles sont utilisées pour trouver au moins une solution optimale à un problème. L'inconvénient majeur de ces méthodes est que le nombre de combinaisons augmente avec l'augmentation de la dimension du problème. L'efficacité de ces algorithmes n'est donc prometteuse que pour les instances de problèmes de petites tailles. Il y a trois méthodes exactes : programmation linéaire, programmation dynamique et évaluation-séparation.

Contrairement aux méthodes exactes, les méthodes approchées ne fournissent pas forcément une solution optimale, mais seulement une bonne solution dans un temps raisonnable. Il existe deux méthodes approchées : heuristique et métaheuristique. Une heuristique est un algorithme qui permet de trouver dans un temps polynomial une solution réalisable, en prenant en compte une fonction objective, pas nécessairement optimale ou exacte, pour un problème d'optimisation difficile. Ce type de méthode traduit une stratégie s'appuyant sur la connaissance du problème. Une heuristique est spécifique au problème et ne peut pas être généralisée. Les métaheuristiques sont des méthodes généralement inspirées de la nature. Contrairement aux heuristiques, elles s'appliquent à plusieurs problèmes de nature différente. Pour cela on peut dire qu'elles sont des heuristiques modernes, de plus haut niveau, dédiées particulièrement à la résolution des problèmes d'optimisation. Leur but est d'atteindre un optimum global tout en échappant les optima locaux. Elles sont très utiles pour des problèmes de grandes tailles car leur façon de fonctionner est telle qu'elles éliminent successivement et rapidement des solutions non optimales.

Afin de répondre à notre problème d'optimisation de tournées de brancardiers au sein d'un établissement de santé, nous avons étudié trois des cinq méthodes de résolution évoquées ci-dessus : heuristique, métaheuristique et programmation linéaire. La première, les heuristiques, qui vont approcher ce que l'on serait amené à faire en réalité pour résoudre ce problème de manière manuelle. La seconde, la méthode exacte, qui va chercher la résolution la plus optimale à notre problème en fonction de contraintes que l'on définit mathématiquement. Et enfin la dernière, la méthode méta-heuristique qui nous donnera une approximation de la meilleure solution possible en un temps réduit.

² Dr. Kherici N. (2020). "Méthode de résolution en optimisation combinatoire".

Développement

1. Principe et Hypothèse

Comme expliqué plus haut, nous avons travaillé sur trois méthodes de résolution pour notre VRP afin de trouver celle qui répond le mieux aux attentes de notre client. Pour chaque méthode le problème posé reste fondamentalement le même sur un certain nombre de principes et d'hypothèses. Il se définit de la façon suivante, un établissement de santé a un certain nombre de missions de brancardage programmé au sein de ses services et compte un certain nombre de brancardiers. Cependant pour pouvoir les comparer et donner un avis, il faut que nous définissions des règles communes de résolution et de prise de décision qui les feront se ressembler. En comparant les différentes méthodes qui existaient déjà, heuristique et exacte, nous en avons tiré les règles et principes suivants :

- Chaque mission est composée d'une heure de départ et d'une durée de trajet qui prend en compte la prise en charge du patient ainsi que son transport entre deux services.
- Il existe deux types de mission, celles nécessitant un seul brancardier et celles à deux brancardiers.
- Chaque brancardier a une durée de travail journalière maximale à ne pas dépasser.
- Chaque brancardier n'est pas affectable à une mission si elle n'est pas dans ses horaires de travail journaliers.
- Il est important de répartir au mieux les missions entre brancardiers afin qu'ils aient un temps de travail ou un nombre de missions affectées homogène.
- Les deux brancardiers affectés à une même mission doivent être obligatoirement synchronisés au départ de la mission.
- Une mission donnée à un brancardier ne peut pas commencer pendant une mission préalablement donnée au même brancardier, Il doit y avoir une continuité du temps entre les missions des brancardiers.
- Il faut faire en sorte que chaque brancardier diminue son temps de trajet à vide entre deux missions.

Une fois les règles définies, on a schématisé la composition de chaque mission (figure 1) ainsi que le déroulement de quelques-unes avec plusieurs brancardiers (figure 2) afin de visualiser ce que nous cherchons exactement.

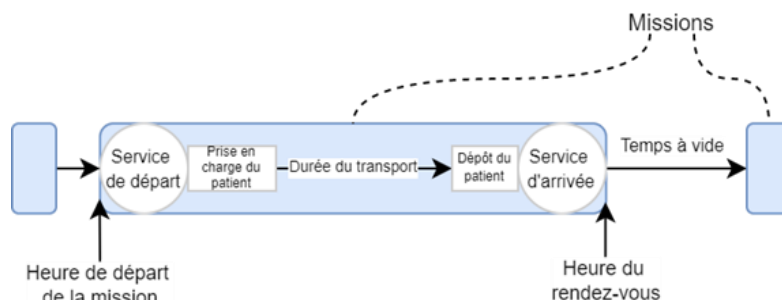


Figure 2. Schéma de composition des missions

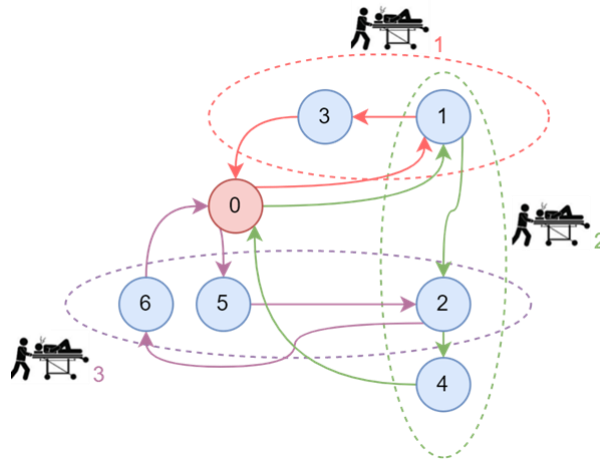


Figure 3. Exemple d'enchaînement des missions

2. Méthode approchée : Heuristique

Les heuristiques sont des méthodes approchées qui ne vont pas forcément fournir la solution la plus optimale à un problème mais une bonne solution. Une heuristique est un algorithme qui permet de trouver dans un temps raisonnable une solution réalisable, en prenant en compte une fonction objective. Ce type de méthodes traduit une stratégie de fonctionnement proche du réel et de la manière dont on résoudrait le problème manuellement.

La méthode heuristique utilisée pour faire nos comparaisons avec les autres méthodes de résolution est la solution existante utilisée par le logiciel de AMC. Cette méthode est écrite en Python et vient traiter des données de plusieurs Excel comportant les informations des différentes missions d'une journée, les informations des différents brancardiers disponibles sur la journée, et une matrice des différents temps de trajets entre chaque service de l'établissement de santé.

Nous nous sommes donc principalement penchés sur son code et sa documentation pour vérifier qu'elle respecte bien les principes énoncés plus haut, et comprendre son fonctionnement.

L'algorithme fonctionne en deux grandes parties comprenant la récupération des données et l'affectation de certaines valeurs pour les missions. Puis l'affectation des "meilleurs" brancardiers pour les missions. Le processus global se rapproche des fonctionnements suivants :

- 1) Traitement des missions (figure 3)
 - Récupération des missions et de leurs caractéristiques de prise de décision comme les services de provenance et de destination, l'heure du rendez-vous ou encore le nombre de brancardiers à y affecter.
 - Affectations des temps de trajets pour les missions en fonction des deux services de la mission et de la matrice de temps, ainsi que d'une priorité aux missions en fonction là aussi des services, car ils possèdent tous un ordre de priorité comme on peut le voir avec la figure 4.
 - Détermination du temps de départ des missions.

- Tri des missions selon leur priorité et leur heure de départ.

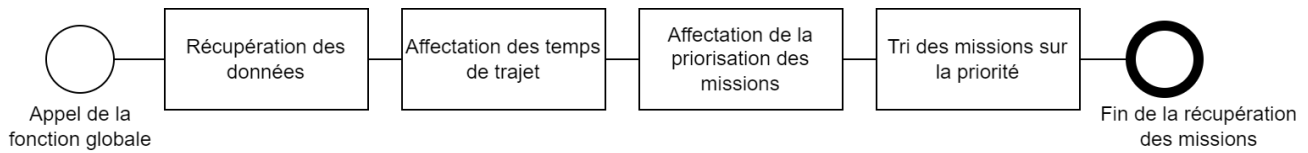


Figure 4. Schéma du processus de récupération des données des missions



Figure 5. Ordre de priorité des services³

2) Affectation des brancardiers (figure 5) :

- Recherche des brancardiers disponible pour une mission en fonction de l'heure de départ et de leurs horaires de travail et s'ils ne sont pas actuellement sur une mission. Tant qu'aucun brancardier n'est disponible pour le moment, on vient ajouter du retard à la mission et on recommence à chercher des brancardiers.
- Affectation d'un score à chaque brancardier permettant de déterminer celui qui sera le plus à même de prendre la mission. Le score se calcule en fonction de la charge de travail qu'à déjà effectué le brancardier dans la journée, de sa proximité avec la mission et de sa capacité à prendre en charge le transport dans les temps.
- Tri des brancardiers par score et on affecte celui ou ceux avec le meilleur score sur le transport. Si aucun brancardier n'est trouvé, on vient exclure les brancardiers qui seraient potentiellement disponibles pour le prochain trajet de la liste et on prend le ou les premiers brancardiers restants pour les affecter à la mission.
- Une fois que tous les trajets ont eu leur affectation, on extrait les résultats dans un excel.

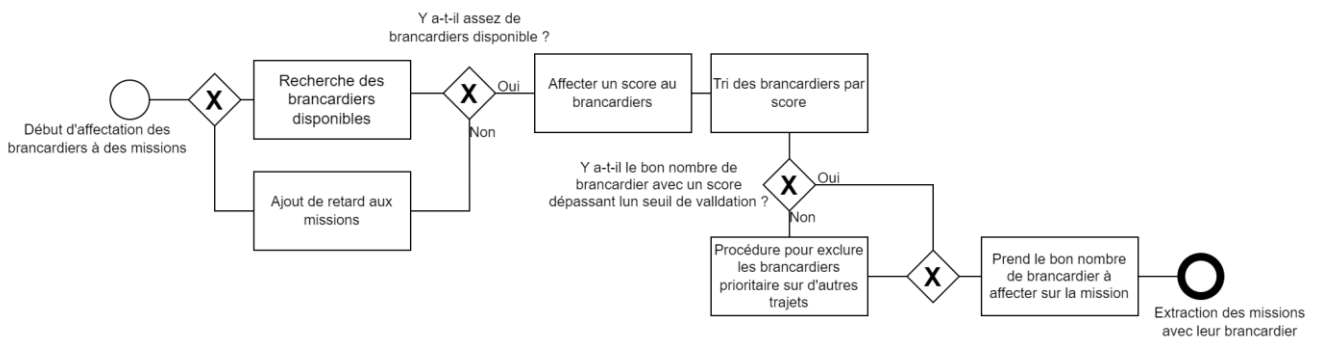


Figure 6. Schéma du processus d'affectation des brancardiers aux missions

³Coubetergues A, De Quina D. (2022). "Projet régulation brancardages"

De par son fonctionnement, la méthode heuristique fournie par AMC n'est pas à modifier pour pouvoir la comparer à nos autres modèles car elle respecte bien les principes que nous avons énoncé précédemment.

3. Méthode exacte : Programmation linéaire

A. Présentation de l'existant

Comme expliqué en introduction de cette partie, les méthodes exactes permettent de donner la solution la plus optimale parmi les nombreuses solutions qu'on peut trouver à un problème posé. Pour le travail que nous avons fourni dans ce projet, nous nous sommes concentrés sur la méthode exacte par programmation linéaire. La programmation linéaire consiste à écrire mathématiquement et linéairement une fonction objective ainsi que les contraintes et règles associées afin de résoudre notre problème. Le terme "programmation linéaire" suppose que les solutions à trouver doivent être représentées en variables réelles. Pour notre cas, il est nécessaire d'utiliser des variables discrètes dans la modélisation du problème, on parle alors de programmation linéaire en nombres entiers. Il est important de savoir que ces derniers sont nettement plus difficiles à résoudre que les programmations linéaires à variables continues.

Nous ne sommes pas partis de zéro pour le modèle linéaire mathématique, nous avons là encore récupéré de la documentation de AMC sur le sujet et particulièrement le travail réalisé en stage par Nelson ROGERS. Pour résoudre ce problème de brancardage, il était parti sur un VRPTW, un VRP avec fenêtre de temps souple. Ce type de VRP donne la possibilité d'un temps d'action qui n'est plus fixe mais qui accepte une fenêtre de temps. Dans notre cas, cela nous donnait la possibilité d'ajouter du retard à une mission et de ne plus avoir une heure fixe pour les rendez-vous, qui pouvait nous poser problème à certains moments de la journée. Concernant les fonctions objectives qu'il avait définies pour résoudre son problème, elles étaient basées sur la minimisation des retards ainsi que la minimisation de l'écart du nombre de missions entre les brancardiers.

B. Améliorations apportées

Nous avons donc gardé ce modèle, mais nous sommes venus y ajouter certaines contraintes et changer la fonction objective car elle ne reflétait plus ce que nous souhaitions faire et ne correspondait plus aux principes que nous avons définis plus haut.

Tout d'abord il a fallu venir modifier la fonction objective en ajoutant un point sur la réduction du temps de trajet à vide. La nouvelle fonction objective ressemble donc à l'équation ci-dessous, avec la minimisation de $Dmoyen_{ij} * x_{ij}^k$, qui est la distance de trajet à vide entre deux missions i et j pour un brancardier k . Ainsi que la minimisation de la variable liée au retard, $late_{ik}$. Et la minimisation de l'écart de travail entre les brancardiers représenté par $t^- + t^+$ pour chaque brancardiers k .

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

Il a fallu aussi revoir plusieurs des contraintes et en ajouter certaines pour que le modèle soit fidèle à nos besoins.

La synchronisation qu'il y avait dans le modèle ne se faisait pas bien et n'était pas totale. En effet, elle se réalisait à la fin de la mission au lieu du début et laissait donc en attente des brancardiers qui ne travaillaient pas pendant un long moment car ils attendaient un autre brancardier pendant plusieurs missions. La synchronisation que nous avons défini permet une synchronisation totale entre les brancardiers avant les missions⁴, synchronisation qui sera lourde, car elle pourra faire attendre les brancardier mais nécessaire pour le bon fonctionnement du modèle. On vient donc vérifier que le départ $Tdep_i$ d'une mission nécessitant deux brancardiers, soit le même pour les deux brancardiers k et l .

$$Tdep_{i \neq 0}^k \leq Tdep_{i \neq 0}^l + M * (1 - x_{ij}^k) \quad \forall i, j \in N^2, ij > 0, \forall k, l \in B^2$$

Un autre point à changer dans le modèle, a été la manière de définir l'heure de départ en mission. En effet, on vient définir une fenêtre de temps dans laquelle l'heure de rendez-vous est acceptable. Cette fenêtre est comprise entre l'heure initiale du rendez-vous et une durée maximale qui est définie en fonction de la priorité des missions. Plus une mission est prioritaire, moins son temps de retard maximal accepté sera grand. L'heure de départ de la mission est donc définie par rapport à cette intervalle de temps et en fonction de la durée de la mission, comme on peut le voir dans l'équation suivante.

$$(Tdep_i^k \geq Trdv_i - D_i) \&\& (Tdep_i^k \leq Trdv_i + R_i + D_i) \quad \forall i \in N, \forall k \in B$$

Nous n'avons détaillé dans cette partie que les contraintes que nous avons modifié ou ajouté au modèle que nous avons récupéré de AMC, le modèle complet est trouvable dans l'annexe 1 avec le tableau récapitulant tous les paramètres et toutes les variables de décision nécessaires au fonctionnement du modèle.

4. Méthode approchée : Méta-heuristique

A. Présentation d'OptaPlanner

OptaPlanner est un solveur de contraintes Open Source développé par Red Hat en 2006. Il résout les problèmes de satisfaction de contraintes avec des algorithmes heuristiques de construction et méta-heuristiques. La promesse d'OptaPlanner est de proposer une solution quasi-optimale en un temps raisonnable à des problèmes d'optimisation complexes et sur des grands jeux de données. Il est possible d'ajouter deux types de contraintes, soft et hard, contrairement aux autres méthodes heuristique et exacte qui n'ont qu'un seul type de contraintes. Avoir deux types de contraintes permet de les classer et de mettre les contraintes qui sont obligatoires à respecter, en hard et celles qui sont plus souples, en soft. Par exemple, pour notre projet, le fait que chaque brancardier doit travailler le même nombre d'heures en moyenne est une contrainte hard alors que le fait qu'ils ne fassent pas d'heures supplémentaires est une contrainte soft. L'algorithme va alors tester itérativement plusieurs

⁴ Malapert, A. (2006). "Optimisation de tournées de véhicules pour l'exploitation de Réseau Telecom".

solutions possibles sur un jeu de données et éliminer en priorité celles dont les contraintes hard ne sont pas satisfaites puis celles dont les contraintes soft ne sont pas respectées. Il convergera obligatoirement vers une solution même si celle-ci n'est pas la solution optimale elle en sera une approximation avec le plus de contraintes hard et soft satisfaites. OptaPlanner fonctionne notamment avec un système de score par rapport aux contraintes définies, chaque contrainte hard et soft se voyant attribuer des points de pénalités si elles ne sont pas satisfaites. Le score est donc séparé en deux, un pour les contraintes soft et un pour les contraintes hard. La solution finale sera donc en priorité celle avec le score de contraintes hard le plus bas car c'est lui qui prime puis avec le score soft le plus bas également.

Au niveau de son installation, OptaPlanner est accessible en ligne à tous et il ne s'agit pas de télécharger un outil à part entière mais de récupérer des fichiers de codes préexistants contenant plusieurs exemples à exécuter ensuite sur un IDE Java. Ces fichiers sont en fait des exemples de plusieurs problèmes d'optimisation comme un TSP, un VRP, un emploi du temps scolaire, une répartition de tâches... Ils se composent de scripts java classiques auxquels ont été ajoutés des dépendances optaplanner permettant la résolution méta-heuristique des problèmes d'optimisation proposés. L'objectif étant de permettre à l'utilisateur, après récupération, de modifier ces fichiers pour les adapter aux besoins des utilisateurs et pour les exécuter ensuite sur leurs jeux de données qu'ils peuvent intégrer. OptaPlanner est compatible avec Java, Kotlin, Python et Scala, soit avec la programmation orientée objet et la programmation fonctionnelle. Les fichiers d'exemples téléchargeables via le site officiel constituent le moyen le plus simple d'entrée en matière avec OptaPlanner car il suffit d'adapter les fichiers préexistants cependant il est également possible de créer un projet à partir de rien dans le cas où aucun exemple ne s'apparenterait au problème à résoudre. Dans ce cas, c'est principalement l'utilisation des frameworks Quarkus et Spring Boot qui seront à privilégier pour créer un projet et y ajouter les dépendances nécessaires à la résolution du problème d'optimisation et décrites sur la documentation d'OptaPlanner. OptaPlanner est aussi un outil collaboratif car en récupérant les fichiers on peut également récupérer des jeux de données que d'autres utilisateurs ont testé avec les exemples proposés avant nous.

Lors du téléchargement des fichiers 2 possibilités d'exécution s'offrent à l'utilisateur : "quickstarts" une version plus ergonomique car sans doute plus récente s'ouvrant en local sur votre navigateur ou une version "examples" avec plus d'exemples mais visuellement plus ancienne s'ouvrant sur une fenêtre java.

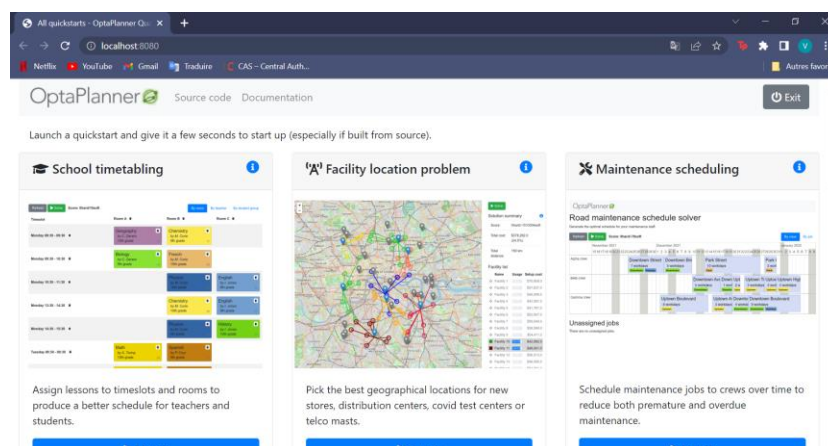


Figure 7. Interface d'OptaPlanner version "Quickstarts"

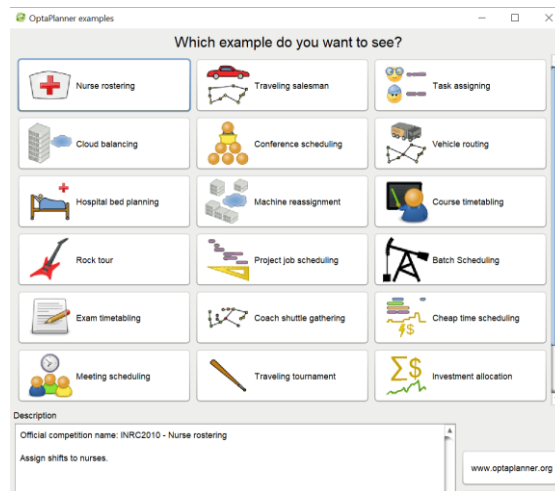


Figure 8. Interface d'OptaPlanner version "Examples"

Ces deux versions sont d'arborescence quelque peu différentes et c'est la deuxième option que nous avons choisi d'explorer au cours du projet car sa construction était plus simple de compréhension.

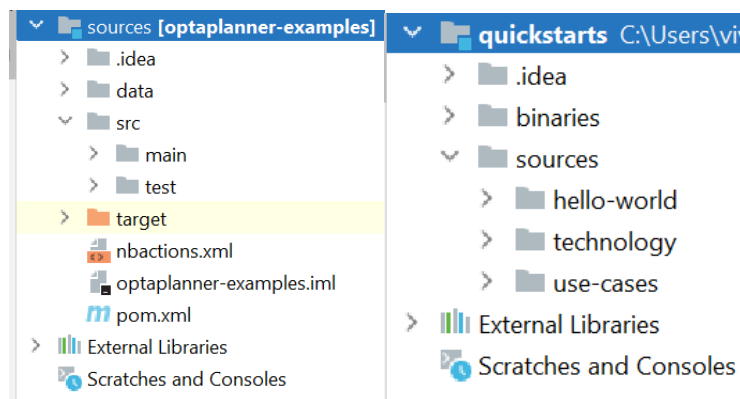


Figure 9. Comparaison des arborescences des versions d'exécution "Examples" (gauche) et "Quickstarts" (droite) d'OptaPlanner

En effet, comme on peut le voir sur la figure 8, l'arborescence de la version "examples" comporte à la fois le dossier source "src" où se trouve les fichiers utilisés pour la résolution des problèmes proposés ainsi qu'un dossier "data" dans lequel on trouve tous les jeux de données précédemment testées par d'autres utilisateurs d'OptaPlanner ainsi que nos propres jeux.

Dans le cadre de notre projet et pour en faciliter la continuité et la reprise par un futur groupe d'étudiants nous avons également réalisé un document complémentaire expliquant plus en détail ce que nous avons compris de la structure et du fonctionnement d'OptaPlanner.

B. Solution envisagée et problèmes rencontrés

Le but de l'utilisation d'OptaPlanner est donc d'essayer une nouvelle méthode de résolution du problème de gestion des flux de brancardiers qui se devrait plus proche de la réalité que la méthode heuristique et plus rapide que la méthode exacte sur un grand jeu de données. Pour se faire nous avons choisi de nous baser sur l'exemple de résolution de VRP déjà existant fourni par OptaPlanner car c'était celui qui se rapprochait le plus de notre problème d'optimisation.

La principale difficulté que nous avons rencontrée dans l'assimilation de ces deux modèles et ce qui nous a bloqué ensuite pour l'implémentation de notre problème dans Optaplanner a été la différence de dimension à minimiser. En effet, dans l'exemple classique de VRP proposé on cherche à minimiser la distance parcourue par chaque véhicule pour effectuer sa tournée comme on peut en voir un exemple sur la figure 9, or dans notre problème les deux objectifs sont de minimiser le temps d'attente des patients, en minimisant notamment les retards pris sur les missions et d'assurer l'équité dans le temps de travail de chaque brancardier. Plusieurs solutions ont été envisagées pour pallier ce problème de dimensions.

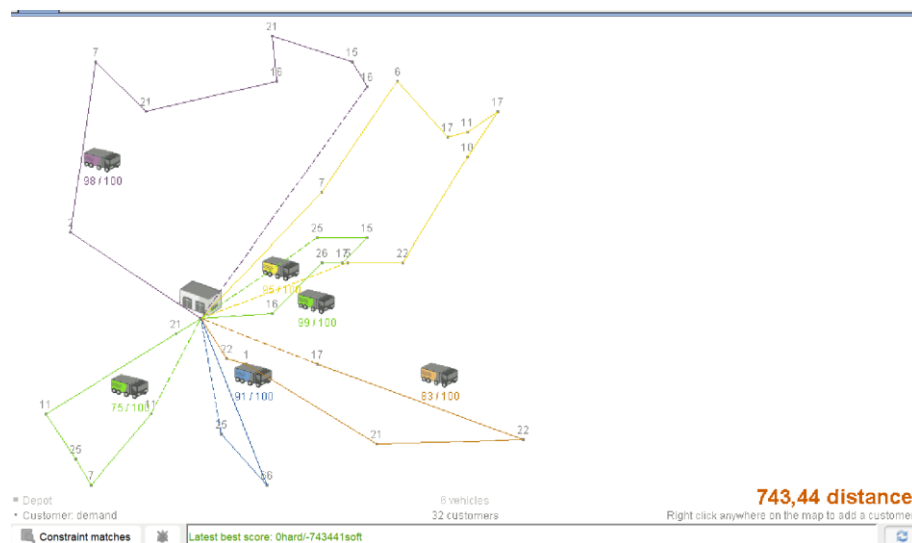


Figure 10. Exemple de résolution d'un VRP sur Optaplanner

Nous avons d'abord pensé à conserver la structure de l'objectif du VRP d'OptaPlanner en réfléchissant en valeur et non dimension, en plaçant nos missions de telle sorte à ce que la valeur de la distance entre tous les points soit égale à la valeur de temps de déplacement qu'il y a entre chaque missions (1 mètre = 1 minute). L'idée était de minimiser des données sans se soucier des dimensions en jeu. Cependant, cette solution a été rapidement éliminée pour 2 raisons, la première était justement ce problème de dimension car même s'il pouvait à priori être omis pour le calcul de minimisation il se poserait sûrement dans d'autres fonctions implémentées dans d'autres fichiers, c'était donc une simplification trop risquée. De plus, pour le placement des missions pour simplifier la tâche il aurait fallu les placer en ligne sur un l'axe des x par exemple pour pouvoir facilement gérer leurs distances entre elles et par rapport à la mission 0. Cependant, cette disposition aurait été illisible et aurait enlevé l'une des caractéristiques intéressantes d'OptaPlanner qu'est la visualisation en temps réel de résolution du problème. Sinon nous aurions dû faire des calculs de distance à partir des coordonnées de chaque mission pour les placer sur le plan de sorte à faire correspondre leur

distance aux autres avec la matrice de temps à vide. Cela s'avérait déjà compliqué à réaliser avec notre jeu de données créée avec 13 missions, c'était donc inenvisageable de le faire à la main sur un jeu de données réel d'une centaine de missions. A cela s'ajoutait aussi le fait que dans les conditions réelles de brancardage le temps aller d'une mission A à une mission B n'est pas forcément le même que le temps retour.

Après avoir éliminé cette option, nous nous sommes penchés sur la conversion de distance en temps. En effet, nous avons eu l'idée de changer de modèle et prendre l'exemple de l'emploi du temps mais M.Zhang, qui connaît bien l'outil OptaPlanner et l'utilise sur ses projet, nous a conseillé de rester sur l'exemple du VRP et de calculer le temps entre deux missions en partant de la formule $\text{vitesse} = \text{distance} / \text{temps}$. Pour ce faire, il nous fallait la vitesse moyenne d'un brancardier et la distance entre deux missions. Cependant, cela implique de connaître ces distances, il faudrait donc récupérer pour chaque établissement un plan à l'échelle et même avec celui-ci, persisteraient les problèmes des conditions réelles qui font que l'aller et le retour entre deux points n'empruntent pas le même chemin et n'ont donc pas la même distance. On peut ajouter à cela, que sur OptaPlanner les coordonnées sont planes (axe x et y uniquement), hors dans un hôpital les missions de brancardage sont souvent sur différents étages. A ce stade du projet nous n'avons donc pas réussi à adapter la fonction objective du VRP existant d'OptaPlanner à nos besoins, ce qui nous a également freiné dans l'écriture des contraintes qui en découlent et que nous n'aurions pas pu visualiser. Une des pistes à explorer pour la suite du projet serait d'implémenter la matrice de temps préalablement écrite. De ce que nous en avons vu et de notre compréhension de l'outil, nous pouvons dire qu'OptaPlanner est en effet un outil avec un grand potentiel pour la résolution de problèmes d'optimisation de grande taille et utilisé à bon escient il pourrait vraiment être un atout pour les travaux d'optimisation d'AMC. Cependant, il est aussi simple d'utilisation qu'il est complexe à s'approprier, à comprendre et à modifier.

5. Indicateurs

A. Construction des indicateurs

En parallèle de nos travaux sur les méthodes d'optimisation et leur comparaison nous nous sommes penchés sur la construction d'indicateurs qui permettraient de caractériser un jeu de données avant même de chercher à l'optimiser pour tester sa faisabilité. Cette démarche vise à faire gagner du temps et de l'argent à notre client en lui évitant de se lancer dans un problème non résoluble initialement.

Nous avons débuté en nous interrogeant sur les raisons réelles pour lesquelles une tournée ne pourrait pas être correctement réalisée. Il en est ressorti 8 raisons :

- 1) Une charge de travail trop élevé (trop de missions) par rapport au nombre de brancardiers
- 2) Une durée totale des missions sur une journée par brancardier plus longue que la durée de travail maximale autorisée par brancardier par jour
- 3) Plus de chevauchement de mission en simultanée que de nombre de brancardiers
- 4) Trop de chevauchement de mission nécessitant 2 brancardiers
- 5) Trop de chevauchement de missions de priorité élevée
- 6) Trop de missions juste avant, après ou pendant la pause

- 7) Trop de missions en début de journée
- 8) Trop de missions en fin de journée

En effet, ces situations peuvent poser problème. Dans le premier cas, la tournée peut être infaisable, pour le savoir il faudra regarder plus en détail la durée des missions, la localisation des missions et le chevauchement. Dans le second, on est sûr que la tournée ne pourra être réalisée car il n'est pas envisageable qu'un brancardier travaille plus que la durée maximale autorisée et ici on ne compte même pas le temps de déplacement entre les missions. Dans tous les autres cas, ces situations pourraient provoquer des retards sur la tournée et c'est justement ce qu'on cherche à minimiser pour éviter de l'attente au patient. Le dernier cas peut également causer des heures supplémentaires aux brancardiers. On remarque donc que selon leurs conséquences sur les tournées, l'apparition de ces situations est plus ou moins critique. Un peu comme la notion de contraintes hard et soft on peut ici donner un niveau de criticité à ces situations. Une criticité de niveau 1 indiquerait une situation rendant infaisable le dataset. Une criticité de niveau 2 indiquerait des situations qui iraient à l'encontre de contraintes fondamentales que doivent respecter les tournées. Enfin une criticité de niveau 3 servirait à classer des situations indésirables par rapport aux objectifs de l'optimisation. On peut ainsi appliquer à chacune des situations ci-dessus une criticité comme indiqué sur la figure 10.

Situations	N°1	N°2	N°3	N°4	N°5	N°6	N°7	N°8
Criticité	1 ou 2	1	3	3	2	3	3	2

Figure 11. Tableau des criticités en fonction des situations

A partir de ces constats nous avons pu écrire mathématiquement nos indicateurs. Les indicateurs qui découlent de ces situations sont tous des indicateurs de pertinence, c'est-à-dire qu'ils évaluent les moyens mis en œuvre par une entreprise pour atteindre ses objectifs. Tous les indicateurs issus de ces situations peuvent être considérés comme des indicateurs charge/capacité sauf le deuxième qui est plus un indicateur temporel. Tous ces indicateurs servent à poser un diagnostic sur la faisabilité des jeux de données a priori. A ce stade du projet nous avons pu écrire les trois premiers complètement avec leur formule et leur seuil de comparaison. Pour le premier, il s'agissait d'évaluer en moyenne combien de missions chaque brancardier allait avoir par jour et de comparer à un seuil correspondant à une moyenne de missions réellement réalisées par brancardier en une journée en s'appuyant sur plusieurs jeux de données. Le tout sous l'hypothèse de brancardiers à temps plein. Nous en avons déduit l'indicateur suivant :

$$\left(\sum_{x=1}^{N_1} x + 2 * \sum_{y=1}^{N_2} y \right) / B \leq 5,15$$

Avec N_x = nb de mission à x brancardiers,
 B = nombre total de brancardiers.

Le seuil de 5,15 a été calculé selon les données d'un jeu réel ayant pu être réalisé, il devra être amélioré en prenant en compte plusieurs jeux de données.

Le second indicateur correspond à la comparaison entre le temps moyen que chaque brancardier passait à effectuer ses missions en une journée avec la durée de travail maximale autorisée par brancardier en moyenne. On peut le traduire ainsi :

$$\left(\sum_{x=1}^{N_1} tpsMiss_x + 2 * \sum_{y=1}^{N_2} tpsMiss_y \right) / B < \sum_{k=1}^B Tra_{\max}^k / B$$

Avec B = nombre total de brancardiers,

$\sum_1^N tpsMiss_x$ = durée totale des missions à x brancardiers en une journée,

Tra_{\max} = durée maximale de travail autorisé en une journée pour un brancardier.

Pour le troisième indicateur, l'objectif est de trouver le nombre maximal de chevauchement en simultanée à un instant t et de le comparer au nombre de brancardiers total. Pour ça on calcule pour chaque mission son chevauchement avec toutes les autres missions. Si chevauchement il y a entre une mission donnée et une autre mission on ajoute 1 à la somme des chevauchements de cette mission donnée. On obtient ainsi le nombre total de missions avec lesquelles cette mission se chevauche. En faisant de même pour chaque mission et en prenant le maximum de ces sommes on obtient l'indicateur voulu. On obtient la formule suivante :

$$\max \left(\sum_{y=1}^N \sum_{i=1}^N \frac{\max(\min(d_y, d_i) - \max(Hrdv_y, Hrdv_i), 0)}{\min(d_y, d_i) - \max(Hrdv_y, Hrdv_i)} \right) < B$$

Avec $d_x = Hrdv_x + tpsMoy_x$ l'heure de fin de la mission x ,

$Hrdv_x$ = heure de début de la mission x ,

$tpsMoy_x$ = durée moyenne de la mission x ,

B = nombre total de brancardiers.

Les deux premiers indicateurs sont portés sur des moyennes, en effet on regarde le nombre moyen de missions par brancardier par jour et la durée moyenne d'une journée de missions par brancardier. Ceux-ci sont à nuancer par rapport aux durées des missions, car s'il y a de grand écart entre ces durées il faudra y prêter attention dans l'affectation des missions pour ne pas se retrouver avec deux brancardiers avec le même nombre de missions mais l'un avec que des missions courtes et l'autre avec que des missions longues par exemple. Pour être alerté de cela, il est intéressant de regarder un indicateur de dispersion des valeurs comme l'écart-type. En calculant l'écart-type des durées des missions, si celui-ci est élevé on peut en déduire qu'il y a des gros écarts dans le temps que dure chaque missions, il faudra donc être vigilant lors de l'attribution, ou si au contraire celui-ci est faible cela signifie que les durée des missions sont toute proche de la moyenne.

Pour vérifier la construction de nos indicateurs et leur pertinence nous les avons testé sur des jeux de données en les implémentant sur Excel. Les résultats de ces tests seront présentés dans la prochaine partie lors de la présentation des jeux de données utilisés.

B. Modélisation du processus de brancardage

Une autre façon de visualiser ces indicateurs serait de les voir comme des capteurs sur le processus d'une tournée de brancardage. Pour cela, nous avons commencé par modéliser ce processus. Nous l'avons cartographié (figure ?) en reprenant les grandes étapes de réalisation de la tournée ainsi que les consignes de transports que nous avons assimilé à la partie pilotage du processus. On retrouve ainsi l'arrivée des brancardiers sur leur lieu de travail (la mission 0 de laquelle ils partent), puis la prise de connaissance par les brancardiers des missions à réaliser dans la journée dont les horaires et les modalités ont été planifiées à l'avance. Puis le déroulement classique de la journée avec la réalisation des missions du matin qui ont été affectés aux brancardiers, la pause déjeuner, les missions de l'après-midi puis la fin de la journée avec le départ des brancardiers, le tout chapeauté pour réguler la charge de travail et les retards pris sur les missions des brancardiers. Ces étapes sont décrites plus en détail dans les modèles orientés flux présentés en annexe 2.

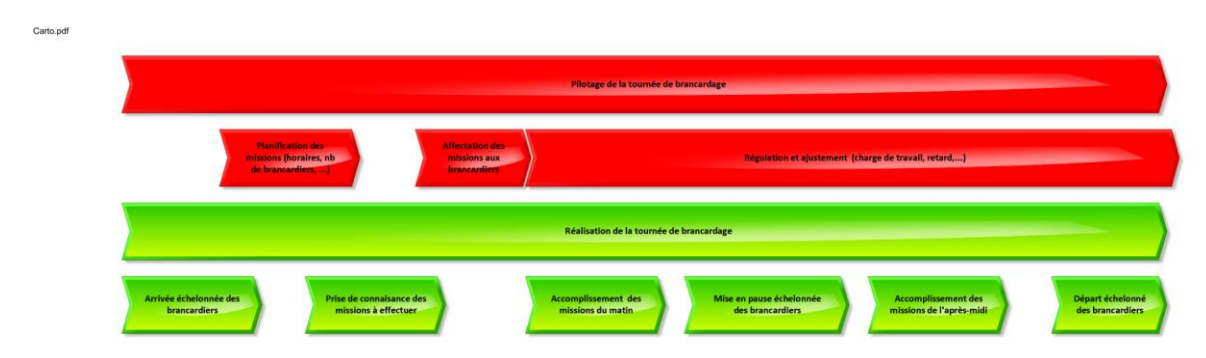


Figure 12. Cartographie du processus de tournée de brancardage

En posant des capteurs de temps d'exécution et de nombre d'exécution sur ces modèles on peut alimenter nos indicateurs de telle façon que si l'un des capteurs dépassent le seuil qui lui avait été attribué il passe au rouge et le processus s'arrête si c'est un indicateur très critique ou à l'orange et le processus est en péril si l'indicateur n'empêche pas sa poursuite. Cette vision s'apparente à une ligne de vie du processus qui suit son cours mais est endommagée voire arrêter en fonction des résultats des capteurs/indicateurs. On peut ainsi cibler précisément les endroits et les moments où le processus est le plus fragile et ainsi où et quand il est nécessaire de renforcer les ressources.

6. Comparaisons

A. Jeux de données

Une fois nos modèles définis nous nous sommes concentrés sur la comparaison de ces derniers. Il a fallu dans un premier temps définir différents jeux de données sur lesquels travailler.

Le premier que nous avons créé est un jeu de données de 13 missions pour 5 brancardiers. Ce jeu de données nous a permis avant même de le comparer, de vérifier que notre modèle linéaire fonctionne comme nous l'avons vu plus haut. Ce jeu de données est donc composé de 13 missions, répartis entre 6 missions à 1 brancardier et 7 missions nécessitant 2

brancardiers. Chaque mission possède ses caractéristiques propres (annexe 3) et la journée de travail se base sur un référentiel fictif allant de 0 à 100, mais sans unité de temps définie.

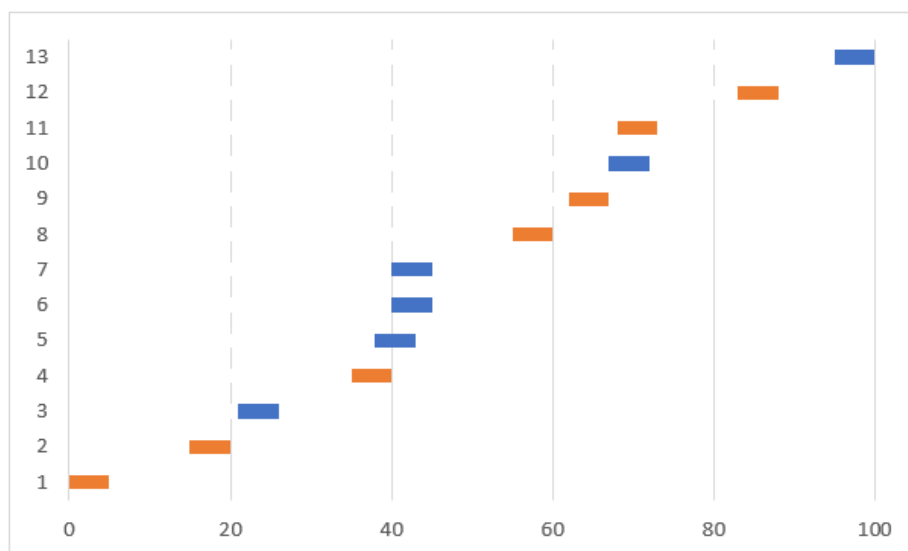


Figure 13. Enchaînement des missions dans le premier jeu de données

Nous avons pu tester nos indicateurs sur ce premier petit jeu de données fictif que nous avons créé de telle sorte à ce qu'il soit solvable. Cette faisabilité a été vérifiée par les indicateurs 1,2 et 3 dont les résultats sont présentés sur la figure 13 où l'on décrit également les seuils auxquels chaque indicateur est comparé. Dans ce jeu de donnée toutes les missions avaient une durée de 5 min, l'écart type était donc de 0 et montrait bien une homogénéité dans les durées.

	A	B	C	D	E
1	Indicateurs	N°1	N°2	N°3	Ecart-type
2		4	20	3	0
3	Seuil	5,15	300	5	

Figure 14. Test des indicateurs sur le premier jeu de données fictif

Le second jeu de données est aussi un jeu de données fictif, mais qui possède plus de missions. Il est composé de 25 missions pour 5 brancardiers, avec 16 missions à 1 brancardier et 9 missions à 2 brancardiers. Ce second jeu de données est lui aussi sur un axe de temps fictif allant de 0 à 100 aussi. Ce jeu de données a été créé en essayant de ressembler au mieux aux données réelles fournies par AMC, que nous verrons par la suite, afin de pouvoir comparer au mieux le fonctionnement et les résultats sur des données réelles.

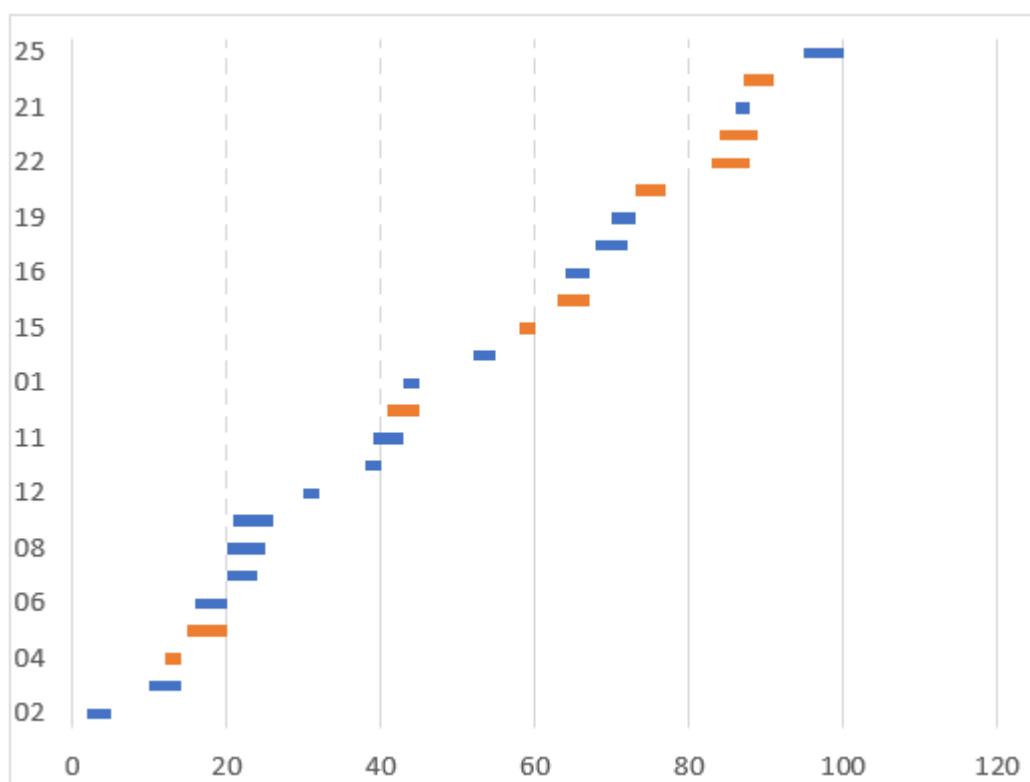


Figure 15. Enchaînement des missions dans le deuxième jeu de données

Nous avons à nouveau testé nos indicateurs sur ce jeu de données (figure 15). Les indicateurs 2 et 3 confirment la faisabilité du jeu, quant au premier il est légèrement supérieur mais est à nuancer car le seuil a été construit à partir d'un unique jeu réalisé. L'écart type ici est plutôt faible, il n'y a donc pas trop d'écart entre les durées des missions ne variant que de 2 à 5 ce qui conforte la fiabilité de nos indicateurs.

	A	B	C	D	E
1	Indicateurs	N°1	N°2	N°3	Ecart-type
2		6,8	25	3	1,11803399
3	Seuil	5,15	300	5	

Figure 16. Test des indicateurs sur le deuxième jeu de données fictif

Le troisième jeu de données est celui fourni par AMC et qui est une journée type de l'hôpital de Maubeuge, un client de AMC. Il est composé de 80 missions avec 57 missions à un brancardier et 23 à deux brancardiers. Les missions s'étendent sur des horaires de 7h40 à 17h15.

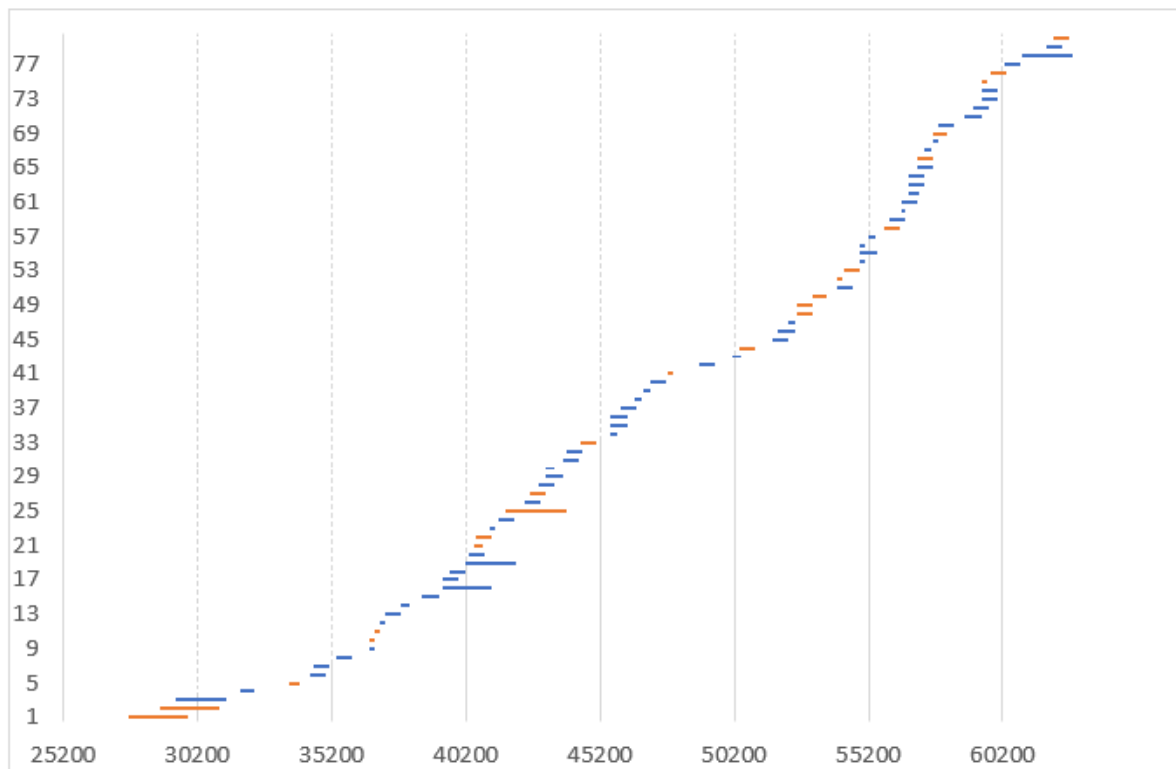


Figure 17. Enchaînement des missions dans le quatrième jeu de données

Ce jeu étant le résultat d'une journée réellement réalisée il était par définition faisable et cela a été confirmé dans le calcul des indicateurs. A nouveau on présente les résultats de ceux-ci et les seuils de comparaison en figure 17. Ici, l'écart type est très élevé, il l'est d'autant plus que la durée des missions est écrite en seconde mais même ramené en minute on obtient un écart type de l'ordre de 7 ce qui reste éloigné de 0. C'est donc un signal d'alerte qui nous indique que les missions peuvent avoir des durées très différentes, en regardant de plus près on constate qu'elles vont effectivement de 7 à 41 minutes.

	A	B	C	D	E
1	Indicateurs	N°1	N°2	N°3	Ecart-type
2		5,15	4494	8	470,821461
3	Seuil	5,15	25200	20	

Figure 18. Test des indicateurs sur un jeu de données réel

Le dernier jeu de données est issu de celui de la journée type à Maubeuge. Il a été créé en supprimant certaines missions de la journée officielle. Il a fallu, pour garder une bonne proportion de missions, supprimer les missions qui se superposent énormément, et alléger la période de midi afin d'être sûr que le modèle fonctionne. Ce jeu de données est donc composé de 50 missions comprenant 35 missions nécessitant un brancardier et 15 missions à deux brancardiers. Les missions, étant en partie les vraies, sont sur un axe de temps non fictif allant de 28800 secondes à 62100 secondes, que représente une journée de travail allant de 8h00 à 17h15.

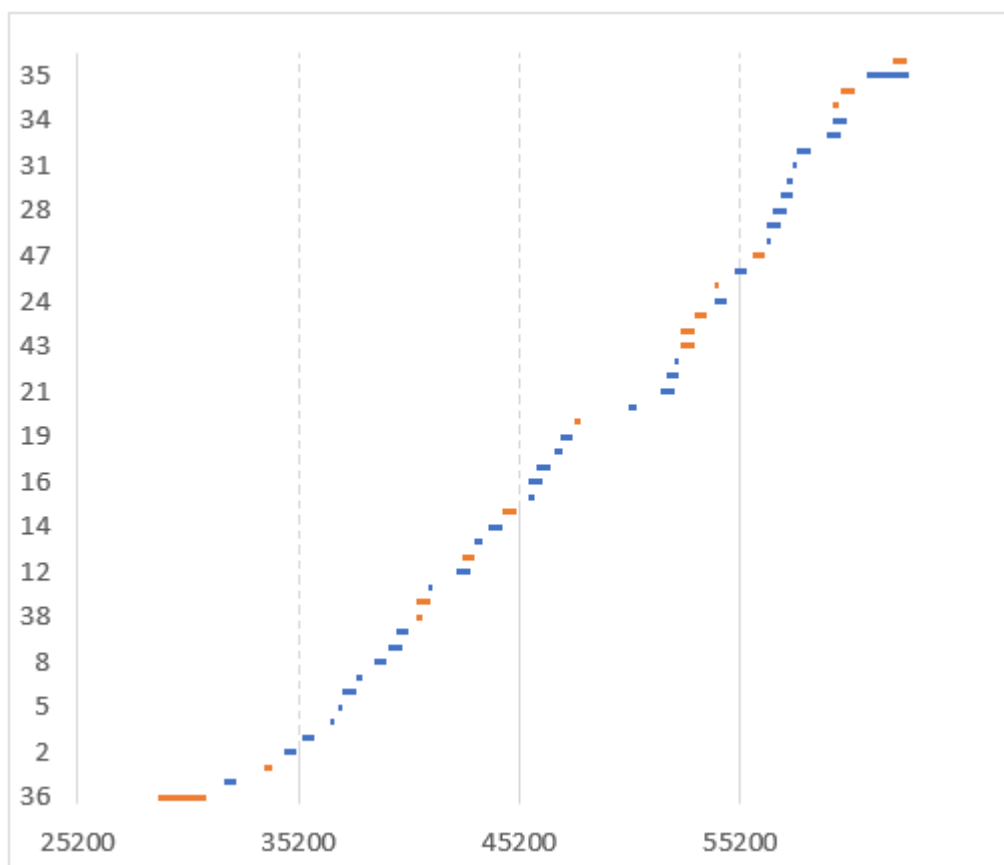


Figure 19. Enchaînement des missions dans le troisième jeu de données

Nous n'avons pas vérifié la faisabilité de ce jeu de données avec nos indicateurs car ce n'est qu'une version réduite en nombre de mission du jeu de données réel de Maubeuge avec le même nombre de brancardiers, il est forcément faisable par extension.

Pour montrer ce que donnerait les indicateurs sur un jeu de données difficilement faisable nous avons repris le premier jeu de données fictif de 13 missions et nous avons diminué le nombre de brancardiers à 3 et leur temps de travail moyen à 30. Les indicateurs se colorent lorsqu'ils dépassent leur seuil, le code couleur sert de signal d'alarme pour les utilisateurs en fonction de la criticité de l'indicateur (de jaune pour la criticité 3 à rouge pour la criticité 1). (figure 19)

	A	B	C	D	E
1	Indicateurs	N°1	N°2	N°3	Ecart-type
2		6,66666667	33,33333333	4	0
3	Seuil	5,15	30	3	

Figure 20. Test de valeurs d'indicateurs supérieures aux seuils

B.Comparaison des modèles

Les comparaisons ont été faites en se concentrant sur chaque jeu de données et modèle, un par un, et en recueillant les informations primordiales et nécessaires à la comparaison finale. Les critères de comparaison sont composés du temps de résolution en fonction du nombre de brancardiers nécessaire pour réaliser la journée. D'autres critères, proches des principes que nous avons évoqué plus haut, vont aussi jouer dans notre prise de décision, comme l'homogénéité du travail des brancardiers et le retard accumulé à la journée.

a. Modèle exacte

Pour utiliser notre modèle exact, nous avons mis en place nos équations mathématiques sur l'IDE IBM ILOG CPLEX Optimization Studio. Cet IDE, développé par IBM, est spécialisé dans la résolution de problèmes linéaires ou tout autre problème d'optimisation. Le langage qu'il utilise est l'OPL, Optimization Programming Language, qui est proche du C et Python dans la manière de fonctionner. Nous avons créé un projet différent pour chaque jeu de données. Ces projets sont composés d'un fichier .mod dans lequel on vient définir notre modèle mathématique en OPL et qui sera le même pour chaque projet, et un fichier .dat, ce sont ces derniers que nous devons changer pour indiquer sur quel jeu de données travailler. Pour les petits jeux de données, c'est nous qui les avons inscrits en brut dans le fichier .dat et pour les jeux de données réelles, il a fallu faire des liens entre le fichier .dat et les fichier Excel que AMC nous avait fourni, comme on peut le voir en annexe 3.

Les résultats obtenus sont présentés dans le tableau de la figure 20. On remarque que le temps d'exécution augmente de manière très significative plus le nombre de contraintes augmente. La répartition du travail des brancardiers est plutôt bonne, on remarque cependant que quand le nombre de brancardiers est supérieur à ce qui est nécessaire alors la répartition est beaucoup moins homogène comme on peut le voir à la ligne des 25 missions pour 6 brancardiers. Quant au retard cumulé sur la journée, il est cohérent par rapport aux données, mais il va falloir le comparer avec les autres modèles pour pouvoir en tirer une conclusion.

Nombre de mission	1B	2B	Nombre de brancardier	Temps d'exécution (seconde)	Répartition (nb de mission)	Retard cumulé (seconde)
13	6	7	4	7	4,4,5,3	25
13	6	7	5	6	3,3,3,3,3	39
13	6	7	6	10	1,2,5,2,1,2	47
25	13	12	4	38	6,8,9,7	175
25	13	12	5	45	7,1,7,8,6	187
25	13	12	6	71	12,8,0,1,4,3	171
50	35	15	4	590	14,13,13,17	12260
50	35	15	5	185	13,14, 9,13, 8	5808
50	35	15	6	150	8,10,12, 8,11, 8	7260
80	57	23	4	Erreur de calcul		
80	57	23	5			
80	57	23	6			
80	57	23	7			
80	57	23	8	3243	11,11,14,12,12,11,13,8	15540

Figure 21. Tableau des résultats de la méthode exacte

Lors des essais sur différents jeux de données, comme on peut le voir dans la figure 20, on a remarqué que notre modèle linéaire prenait beaucoup trop de temps de calcul et qu'il finissait surtout par s'arrêter avec une erreur d'exécution OPL pour le plus gros jeu de données à 80 missions. Nous avons donc essayé de chercher d'où pouvait venir le problème, et nous avons émis plusieurs hypothèses.

La première hypothèse vient se porter sur les contraintes objectives. En effet, avoir trois demandes de minimisation dans une même fonction vient demander plus de ressources à contenter afin de trouver le résultat le plus optimal. Il faudrait alors peut-être revoir cette fonction ou venir normaliser les résultats que peuvent prendre ces variables de décisions afin d'alléger les calculs et les grands nombres que l'on peut retrouver avec notre modèle, surtout concernant le temps que l'on exprime en seconde. De plus, certaines contraintes ne doivent pas être bien formulées et viennent alourdir la prise de décision.

La seconde est que la puissance de calcul de nos machines n'est pas assez performante, car le jeu de données de 80 missions vient créer plus de 200000 contraintes à résoudre ce qui est très lourd.

La dernière de nos hypothèses est que les modèles linéaires ne sont pas faits pour résoudre ce genre de problème trop complexe qui viendront créer trop de contraintes. Ils sont donc un bon point d'entrée pour la définition de contraintes et de principes mais qui doivent être repris dans d'autres types de modèles afin d'être le plus conciliant entre performance et résultat.

b. Modèle heuristique

Comme énoncé plus haut, ce modèle a été réalisé en Python et nous n'avons pas eu à revoir son fonctionnement. Nous nous sommes concentrés sur les grands jeux de données pour ce modèle et surtout sur les données fournies par AMC, qui sont des données réelles et possèdent donc toutes les informations nécessaires pour faire tourner le code. Les données fictives n'étaient pas assez renseignées pour fonctionner sur la méthode heuristique sans avoir à retoucher énormément le code et les fichiers Excel. Il fallait revoir toute la matrice de temps à vide, la manière de définir le temps moyen des missions en définissant des services de provenance et de destination pour chaque mission, créer la priorité des services, etc. Le fait de créer ces données aurait eu une incidence sur les calculs et la comparaison ne serait plus vraiment possible entre les deux méthodes.

Voici donc les résultats que nous avons obtenus pour cette méthode, on remarque que le temps d'exécution du code est toujours excessivement rapide même avec le nombre de missions le plus élevé. La répartition n'est pas très bonne, même si on voit que dans plusieurs cas les brancardiers travaillent de manière assez homogène, il y a des cas, comme celui de 80 missions à 5 brancardiers, où la moyenne de missions par brancardiers est de 20 ± 1 missions par brancardier, moyenne à laquelle peu de brancardiers se rapprochent, et un écart maximal de 17 missions entre 2 brancardiers ce qui est très important. Quant au retard cumulé, il semble cohérent là encore par rapport aux données et à la méthode mais il reste à comparer.

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

Figure 22. Tableau des résultats de la méthode heuristique

c. Modèle Métaheuristique

Nous n'avons pas réussi à créer le modèle métaheuristique pour les raisons évoquées précédemment et n'avons donc pas de résultats sur ce modèle. Nous ne le prendrons pas en compte pour l'évaluation finale entre tous les modèles.

7. Solution Optimale

Une fois nos résultats obtenus, nous avons pu les comparer pour voir lequel est le plus adapté pour notre client.

Sur énormément de points, la méthode heuristique semble bien plus performante. En effet, du point de vue rapidité, son temps d'exécution est largement plus rapide, seulement un dixième de seconde pour la journée type alors que la méthode exacte en est incapable. En ce qui concerne la répartition du travail, cela n'est pas le plus efficace pour les deux méthodes. On remarque que le partage moyen des activités est respecté pour les deux modèles, mais qu'en fonction du nombre de brancardiers affectés, les écarts entre le minimum et le maximum peuvent être assez importants. C'est ce qu'on peut voir avec le dernier cas de la méthode heuristique avec les 80 missions à 6 brancardiers où la moyenne des missions par brancardiers est entre 15 et 16 missions et l'écart maximal entre deux brancardiers est de 14 missions (22 missions pour le brancardier 1 et 8 pour le brancardier 6). Il serait donc bon de retravailler cette répartition pour les deux modèles. Si l'on se penche maintenant sur le cumul du retard sur la journée, on voit que la méthode exacte est bien plus performante. Presque deux fois moins de temps sur la journée pour les jeux de données à 50 missions et un écart d'environ 2h45 pour le cas des 80 missions pour 8 brancardiers. Sur ce point, la méthode exacte est donc à considérer comme « meilleure » solution, car c'est le but que recherchait au début AMC en changeant la manière de fonctionner de sa solution.

Néanmoins, au vu des résultats que nous avons obtenus pour nos deux méthodes, il semblerait donc logique pour AMC de privilégier la méthode qu'ils utilisent déjà, à savoir, l'heuristique. En effet, cette méthode coche le plus de leurs attentes même si ce ne sont pas forcément les plus importantes.

Premièrement, sa vitesse d'exécution. C'est le point le plus intéressant de ce type de méthode et elle l'a bien montré. De plus, on utilise ici que de petits jeux de données, un établissement de santé peut en avoir énormément plus, il est donc nécessaire d'utiliser une méthode pouvant calculer rapidement s'il y a des changements dans la journée.

Deuxièmement, cette méthode ne demande pas de ressources énormes pour fonctionner. La méthode exacte nécessite l'utilisation d'un IDE extérieur, ici CPLEX ILOG Studio, qui n'est pas open source et qui demande une puissance de calcul suffisante à l'ordinateur pour ne pas perdre plus de temps. Inverser dans une licence CPLEX représenterait un coût supplémentaire pour AMC et les établissements de santé pour mettre en place la méthode exacte ce qui n'est pas envisageable. Là encore, la méthode heuristique est donc à privilégier.

Troisièmement, la pseudo-optimisation du retard cumulé à la journée est quand même assez performant. En effet, même si sur ce point la méthode exacte est bien plus performante que l'heuristique, cette dernière n'est pas en reste et reste bien plus performante que la gestion à la main. Avec cette solution, AMC est passé d'un retard cumulé de 15h à environ 5h pour le même nombre de brancardiers sur la journée de 80 missions. Il serait donc bon de venir optimiser cette partie du code afin d'être encore plus performant.

Gestion de projet

En ce qui concerne la gestion de projet, nous nous sommes organisés en commençant par faire un plan projet dans lequel nous avons établi un diagramme de Gantt de départ sous forme de planning. Mais avant cela nous avons établi les limites du projet. En écrivant le contexte, les objectifs attendus, les parties prenantes et les documents de références auxquels nous avons accès. Tout cela avait pour but de vérifier avec M. Pingaud que nous avions bien compris le projet et que nous allions dans la bonne direction.

C'est dans ce plan projet que nous avons désigné Victor comme étant le chef de projet de ce projet tutoré et que nous nous sommes répartis les différentes tâches à effectuer en fonction des heures attribuées au projet tutoré dans notre emploi du temps.

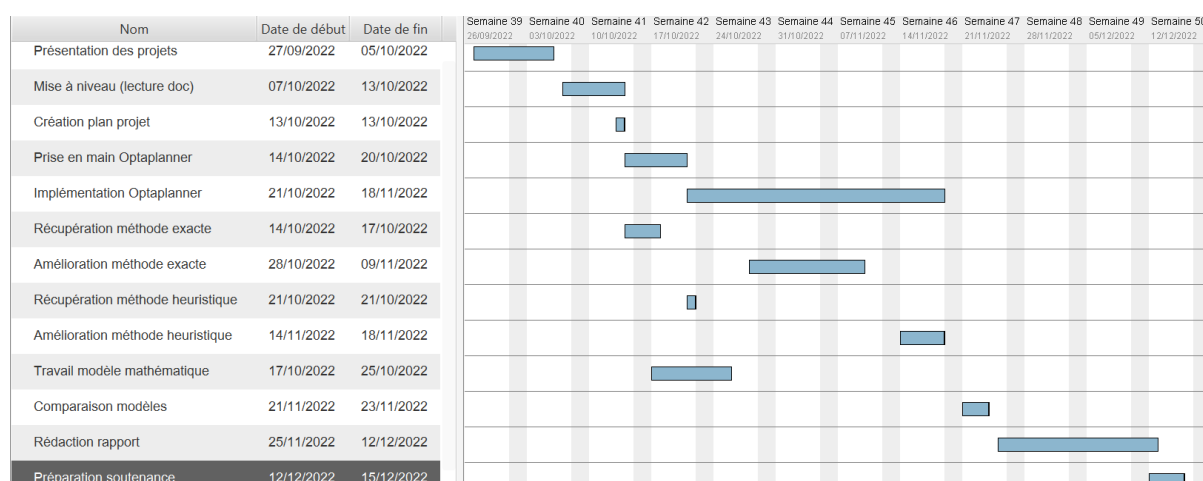


Figure 23. Diagramme de Gantt initial

Nous avons également, dans ce plan projet, effectué une gestion des risques partielle. Afin de comprendre l'envergure du projet et les problèmes que nous pourrions rencontrer. Nous avons relevé, par exemple, l'impossibilité de comparer les solutions, un calcul d'optimisation trop long, un manque d'efficacité avec un trop grand nombre de données. Par la suite nous avons donc envisagé ces risques dans la conception de nos différentes solutions. Concernant la comparaison des solutions, nous avons choisi des indicateurs communs aux trois solutions

qui permettent une comparaison simple. Pour le temps d'optimisation trop long nous avons minimiser le nombre de contraintes et optimisé au maximum les modèles. Nous n'avons malheureusement pas prévu toutes les éventualités et c'est pour cela que nous n'avons pas toujours respecté notre planning de départ.

Nous avons essayé de respecter au mieux ce planning mais malheureusement nous avons dû faire face à quelques problèmes qui nous ont ralenti dans notre projet et quelques modifications comme l'ajout d'indicateurs.

Concernant les problèmes rencontrés, il y en certains que nous aurions pu prévoir comme le temps de prise en main d'Optaplanner mais d'autres qui ne dépendaient pas de nous comme le retard quant au commencement réel du projet. En effet, avant de pouvoir commencer le projet nous avons perdu plus de 20h avec la présentation des projets puis l'affectation des groupes sur les projets. Ensuite, Victoria et Séraphie ont dû rattraper le retard qu'elles avaient sur Victor qui avait déjà travaillé sur le sujet en stage. Il a fallu, pour elles, lire toutes les thèses, les travaux de recherches, les rapports de stages et les documents du stage, cités en annexes, mais également suivre des cours sur Cplex, Optaplanner et plus généralement sur l'optimisation.

Ensuite, nous avons dû apporter quelques changements au modèle exact pour le faire correspondre à notre choix de contraintes et donc modifier la fonction objective. Nous n'avons pas modifié le modèle heuristique déjà existant chez AMC même s'il ne correspond pas exactement à ce que nous avons fait pour les deux autres méthodes car il est impossible de faire correspondre exactement les trois méthodes qui sont trop différentes au niveau de l'écriture et de la réalisation de l'optimisation.

Pour finir, Victor a été malade et n'a donc pas pu travailler pendant 1 semaine ce qui a décalé la comparaison des modèles et donc par conséquent la finalisation du projet. En soit notre diagramme de Gantt initial était en grande partie bon, il y a eu seulement quelques changements sur la fin du projet.

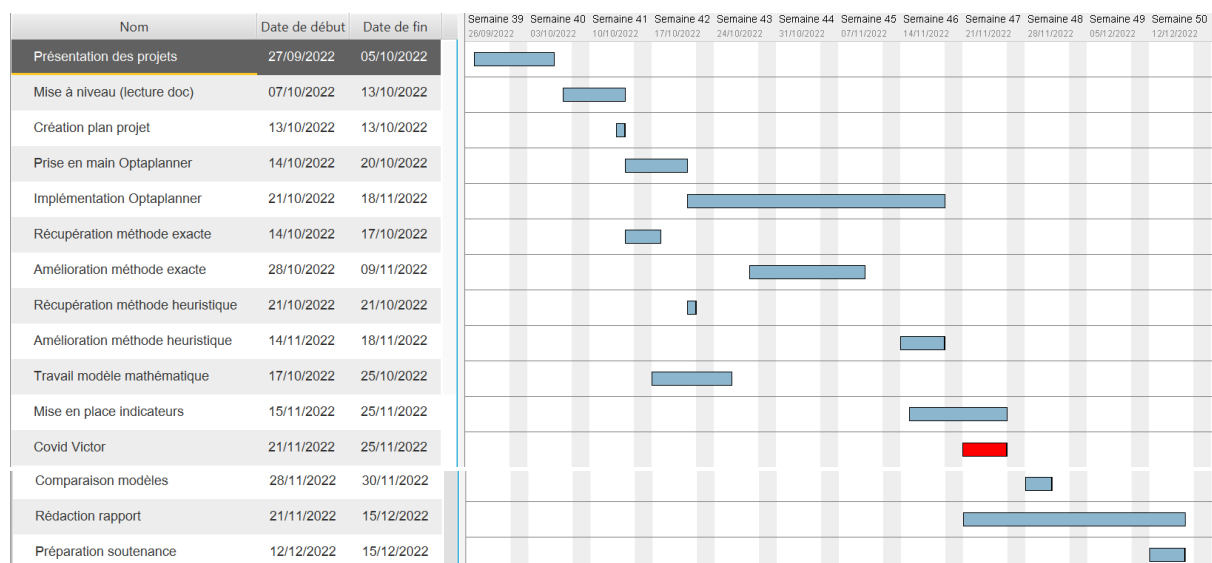


Figure 24. Diagramme de Gantt final

Conclusion

Dans le cadre de ce projet, notre objectif était de comparer différentes méthodes d'optimisation de flux de brancardiers pour notre client, l'entreprise AMC, afin qu'ils puissent choisir la solution qui répondrait le plus à leurs besoins et à ceux de leurs clients. Pour cela nous avons étudié la méthode déjà utilisée au sein de l'entreprise la méthode heuristique, une méthode précédemment initiée dans l'entreprise, la méthode exacte en programmation linéaire et enfin la méthode méta-heuristique avec l'outil OptaPlanner. Sur ces trois méthodes nous avons dû faire face à quelques problèmes.

Pour la méthode exacte le modèle a été à modifier pour le faire correspondre à 100% aux attentes du client. Cependant la taille et la complexité du problème (grand nombre de contraintes et fonctions objectives à 3 critères) a fini par rendre l'utilisation de cette méthode impossible sur des grands jeux de données où les calculs n'aboutissent pas.

Pour la méthode métaheuristique, de par la complexité de l'outil OptaPlanner nous avons pris beaucoup de temps à le comprendre et nous avons ensuite fait face à des problèmes pour modifier un modèle préexistant de VRP et l'adapter à nos besoins. Nous n'en avons pas moins conscience des possibilités qu'offriraient OptaPlanner à l'entreprise AMC, de par sa performance sur des grands jeux de données et d'autant plus que c'est un outil open source, contrairement aux outils nécessaires à la méthode exacte. Nous avons ouvert la voie sur cet outil, mais il reste un grand travail à faire pour l'implémentation puis l'utilisation de cette méthode sur laquelle il serait selon nous bénéfique de se pencher.

La méthode heuristique porte des désavantages surtout sur la minimisation des retards et sur ses limites lorsqu'il s'agit de très grands jeux de données.

Finalement, la solution qui nous semble optimale par rapport aux attentes du client reste la méthode déjà utilisée, soit la méthode heuristique. En effet, elle est plus performante en vitesse d'exécution, moins coûteuse, nécessite moins de matériel performant.

Pour les suites à donner au projet, il s'agirait donc de continuer le travail sur OptaPlanner, d'essayer de décomplexifier le problème pour la méthode exacte et enfin de formuler mathématiquement les indicateurs restants en leur trouvant des seuils pertinents auxquels les comparer.

Bibliographie

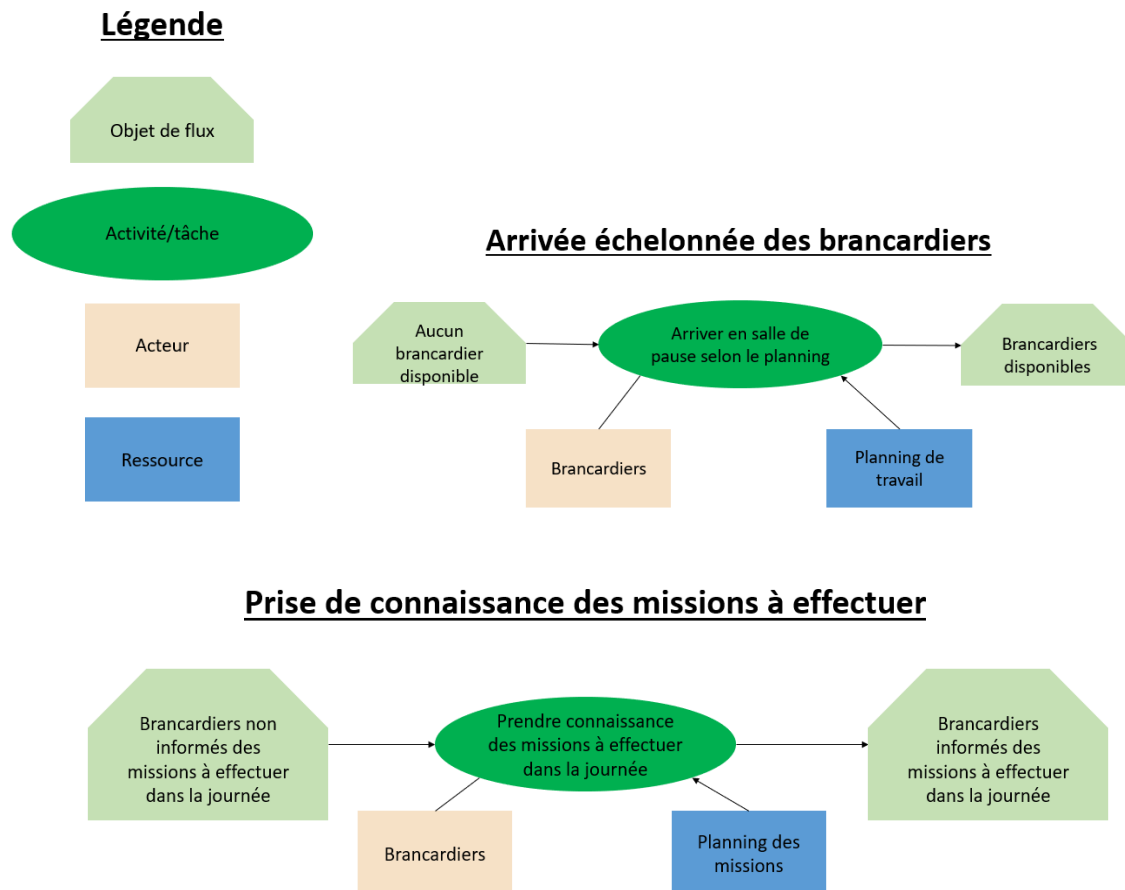
- [1] Bouabdallah, M.N., Rached, M., Fondrevelle, J., Bahroun, Z., (2013). "Organization and management of hospital patient transportation system".
- [2] Malapert, A. (2006). "Optimisation de tournées de Véhicules pour l'exploitation de Réseau Telecom".
- [3] Zhang L. (). "De la vision métier à la génération assistée de plannings pour la coordination centralisée de services de soins à domicile".
- [4] Ducomman S. (2018). "Optimisation de tournées de véhicules par programmation par contraintes : conception et développement d'un solveur industriel".
- [5] Maintenant V. (2022). "Programmation Linéaire et Optimisation du brancardage".
- [6] Rogers N. (2022). "Optimisation du flux de brancardiers".
- [7] Coubetergues A, De Quina D. (2022). "Projet régulation brancardages".
- [8] AMC (2022). "Compte-rendu Sarreguemines Brancardier".
- [9] Ducomman S. (2018). "Optimisation de tournées de véhicules par programmation par contraintes : conception et développement d'un solveur industriel".
- [10] Dr. Kherici N. (2020). "Méthode de résolution en optimisation combinatoire".

Annexes

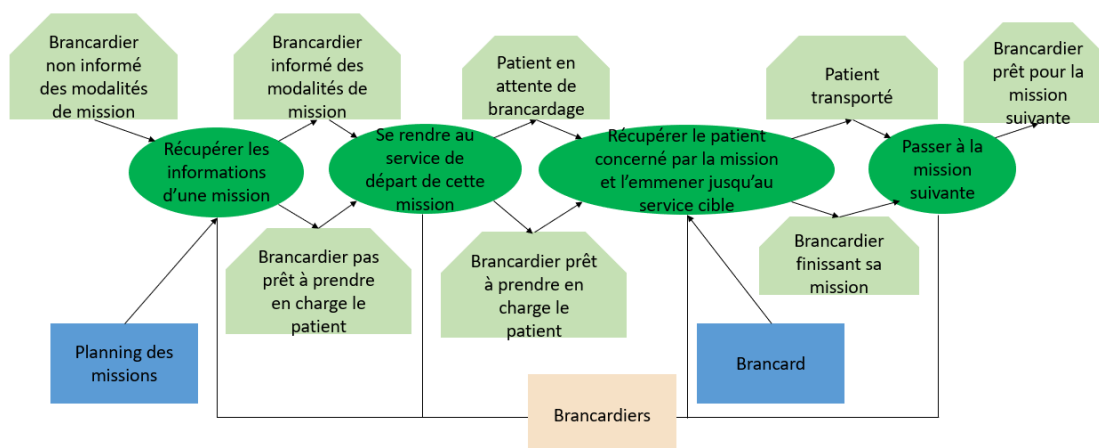
Annexe 1 : Modèle mathématique

CF fichier pdf dans le dossier des annexes nommé : "Annexe1_ModeleMathematique.pdf"

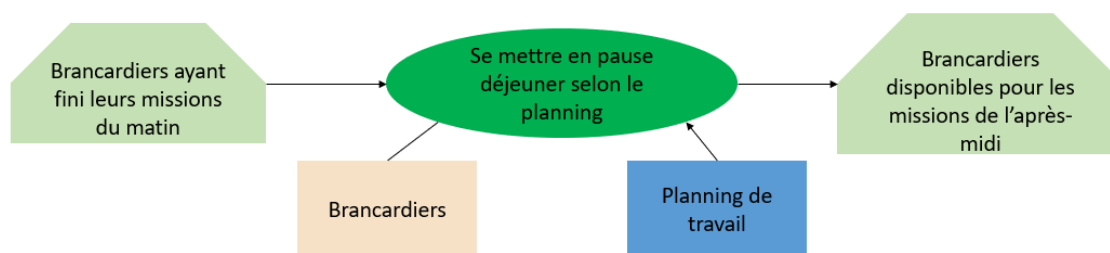
Annexe 2 : Modèles orienté flux du processus de réalisation d'une tournée de brancardage



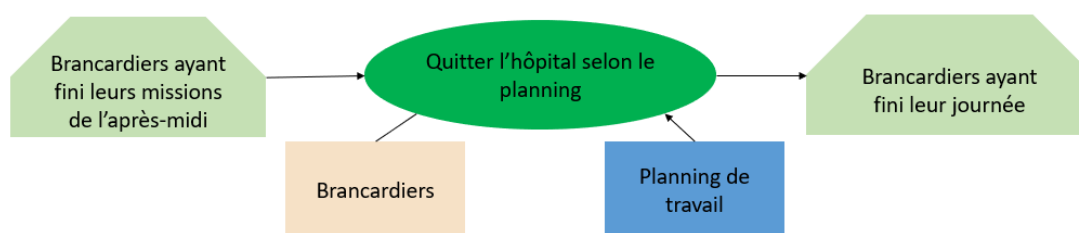
Accomplissement des missions du matin/de l'après-midi



Mise en pause échelonnée des brancardiers



Départ échelonné des brancardiers



Annexe 3 : Récupération des données

```

NbMissions = 13; // Nb de missions
NbMissions1 = 6; // Nb de missions
NbBrancardiers = 5; // Nb de brancardiers
R = [0,20,20,20,20,5,20,20,20,5,20,5,5,15]; // Retard ou avance max accepté
Trdv = [0, 45, 72, 26, 45, 100, 43, 88, 40, 67, 73, 5, 20, 60]; // heures de rdv des missions
D = [0, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5]; // Durée estimée des missions
Hdeb_mat = [0, 0, 2, 0, 5]; //, 5, 10, 30];//, 5, 10, 20];
Hfin_mat = [40, 40, 45, 55, 35]; //, 35, 35, 45];//, 100, 100, 100];
Hdeb_apm = [60, 60, 65, 70, 55]; //, 55, 55, 65];//, 5, 10, 20];
Hfin_apm = [100, 100, 100, 100, 100]; //, 100, 100, 100];//, 100, 100, 100];

Dmoyen=[ [ 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0], // tps de trajet à vide entre 2 missions
[ 0, 0, 2, 2,10, 2, 2, 2, 2, 2, 2, 2, 2],
[ 0, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2,40, 2, 2],
[ 0, 2, 2, 0,10, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[ 0,56, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2, 2],
[ 0, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 2],
[ 0,30, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2],
[ 0, 2, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2],
[ 0,10, 2, 2,10, 2, 5, 2, 0, 2, 2, 2, 2, 0],
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 0, 2, 2, 2],
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2],
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0,15, 2],
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2],
[ 0, 2, 2, 2, 2, 2, 2, 2, 2, 0, 2, 2, 2, 0]];

TraMax = [300, 300, 300, 300, 300]; //, 300, 300, 300];//, 25200, 25200, 25200];

```

Écriture en brut dans le fichier .dat sur CPLEX pour les petits jeux de données

```

SheetConnection donnees("journee_maubeuge.xlsx");
SheetConnection brancardiers("brancardiers.xlsx");
SheetConnection matrice("matrice.xlsx");

NbMissions from SheetRead(donnees, "M2"); // Nb de missions
NbMissions1 from SheetRead(donnees, "L2"); // Nb de missions à 1 brancardier
NbBrancardiers = 5; //from SheetRead(donnees, "N2"); // Nb de brancardiers

R from SheetRead(donnees, "O2:O52");

Hdeb_mat from SheetRead(brancardiers, "F2:F6"); // Heure de début de travail
Hfin_mat from SheetRead(brancardiers, "H2:H6"); // Heure de début de travail
Hdeb_apm from SheetRead(brancardiers, "J2:J6"); // Heure de début de travail
Hfin_apm from SheetRead(brancardiers, "L2:L6"); // Heure de fin de travail

D from SheetRead(donnees, "F2:F52"); // Durée estimée des missions

Trdv from SheetRead(donnees, "I2:I52"); // heures de rdv des missions

Dmoyen from SheetRead(matrice, "J4:BH54"); // tps de trajet à vide entre 2 missions

```

Récupération des différentes informations des trajets et autres dans les fichiers Excel de AMC