

Optimisation Multi-objectif des Tournées de Collecte d'Huiles Usagées avec Fenêtres de Temps, Ramassages Multiples et Tournées Multiples

Florian BERGERE*, Maxime EYNARD*, Yann LANGLO*, Amaury PETERSCHMITT*

* Étudiants en 3^{ème} année d'ingénieur à CY TECH,
option Intelligence Artificielle

4 mars 2024

[Code](#)

Résumé

Ce projet vise à optimiser les tournées d'un véhicule de collecte d'huiles usagées chez des clients en tenant compte des diverses capacités des cuves et des habitudes de remplissage variables en termes de fréquence et de quantité. Ce problème, multi-objectif, de routage de véhicule avec fenêtres de temps, ramassages multiples et tournées multiples est formulé pour maximiser la quantité d'huile collectée tout en minimisant la distance parcourue par le véhicule et en réduisant l'urgence de débordement des cuves. Nous proposons une approche algorithmique combinée à l'utilisation de techniques d'optimisation pour déterminer efficacement les tournées de collecte pour la prochaine journée de l'agent, permettant ainsi une planification précise et efficace des activités de collecte, tout en respectant les contraintes.

Mots-clés

Collecte de Déchets, Problème de Routage de Véhicule, Problème Multi-objectif, Fenêtres de Temps, Ramassages Multiples, Tournées Multiples, Recherche Locale, Optimisation Métaheuristique.

Abstract

This project aims to optimize the routes of a used oil collection vehicle at customer locations, taking into account the various capacities of the tanks and the variable filling habits in terms of frequency and quantity. This multi-objective vehicle routing problem with time windows, multiple pickups, and multiple routes is formulated to maximize the amount of oil collected while minimizing the distance traveled by the vehicle and reducing the urgency of tank overflow. We propose an algorithmic approach combined with the use of optimization techniques to efficiently determine the collection routes for the next day, enabling precise and efficient planning of collection activities while respecting constraints.

Keywords

Waste Collection, Vehicle Routing Problem, Multi-objective Problem, Time Windows, Multiple Pickups, Multiple Routes, Local Search, Metaheuristic Optimization.

1 Introduction

La collecte et le recyclage des huiles usagées jouent un rôle crucial dans la préservation de l'environnement et la promotion de la sécurité publique. Actuellement, un nombre significatif de ces huiles est malheureusement déversé de manière inappropriée, ce qui entraîne une pollution environnementale considérable ainsi que des risques pour la santé publique. Les incidents liés à la manipulation d'huiles chaudes lors de leur collecte constituent également une préoccupation majeure en termes de sécurité. Par conséquent, l'adoption de pratiques de collecte efficaces et sécurisées est essentielle pour atténuer ces risques et minimiser l'impact négatif des huiles usagées sur l'environnement.

Dans cette étude, nous développons une solution visant à optimiser la collecte des huiles usagées dans la région de Toulouse, en France. Pour cela, nous utilisons des données provenant d'un échantillon représentatif de restaurants et d'hôtels locaux, comprenant des informations détaillées telles que les habitudes de remplissage des clients, les capacités des cuves et les plages horaires de collecte.

Notre solution se divise en deux phases distinctes : Dans un premier temps, nous utilisons des techniques algorithmiques avancées pour résoudre le problème complexe d'optimisation des tournées du véhicule de collecte sur une journée, en tenant compte des diverses contraintes, notamment des fenêtres de temps. Cette phase initiale nous permet de générer des solutions de base pour les tournées de collecte. Grâce à une fonction d'évaluation pondérée par rapport à l'échelle de grandeur de chaque objectif, nous avons pu déterminer un score pour chaque solution, ce qui permet de ramener le problème multi-objectif à un problème mono-objectif. De plus, plusieurs versions de choix de cuves dans l'algorithme de cette première partie sont disponibles, notamment des approches heuristiques (semblables à [1]).

Dans un second temps, nous mettons en œuvre différentes méthodes de génération de voisins et des techniques d'optimisation, telles que le recuit simulé et le hill climbing, afin d'améliorer ces solutions de base et d'obtenir des résultats optimisés. Cette approche hybride (semblable à [2]), combinant résolution algorithmique et optimisation métaheuristique, nous permet de concevoir des stratégies de collecte robustes et efficaces pour la gestion des huiles usagées, contribuant ainsi à la préservation de l'environnement et à la promotion de la sécurité publique. Il convient également de mentionner le travail effectué dans la modélisation mathématique du problème et dans la génération des voisins dans cette deuxième partie.

2 Problème

Le problème d'optimisation de tournées de collecte d'huiles usagées (**MO-VRPTWMPMR**) que nous abordons dans cette étude est un problème complexe de routage de véhicule (semblable à [3]) avec plusieurs objectifs à optimiser. Il s'agit d'un problème d'optimisation combinatoire qui appartient à la classe des problèmes NP-difficiles. Nous restreignons ce problème à un seul dépôt et un seul agent pour simplifier l'analyse et la modélisation, tout en conservant sa complexité intrinsèque.

2.1 Modélisation Mathématique

On souhaite trouver une solution pour le jour j .

Remarque : x_{varj} correspond donc à l'état de la variable x_{var} au jour j .

2.1.1 Variables

Un **dépôt** dispose :

- d'une localisation : (dep_{lat}, dep_{lon})

Un **agent** dispose :

- d'un temps maximal de travail : w_{agent} (en min)
- d'une heure de début possible : $start_{agent}$
- d'un **véhicule** disposant :
 - d'un volume : $vehicule_{vol}$ (en L)
 - d'une capacité : $vehicule_{cap}$ (en L)
 - d'une vitesse (moyenne) : $vehicule_{vit}$ (en km/h)

*Remarque : Pour refléter au plus près la réalité, nous supposons l'existence d'un laps de temps nommé **latence**. Ce temps moyen représente la période entre l'arrivée à une cuve et le départ vers un autre emplacement, ajusté en fonction du temps de travail nécessaire à ce point spécifique (pompage pour un client et vidange pour un dépôt).*

Un client possède au moins une **cuve** disposant :

- d'une localisation : (c_{lat}, c_{lon})
- d'une quantité disponible : $c_{volj} = c_{vol(j-1)} + c_{rempj}$ (en L)
- d'une capacité : c_{cap} (en L)
- d'une **pompe** disposant :
 - d'une vitesse de pompage : p_{vit} (en L/min)
 - d'une vitesse de vidange : v_{vit} (en L/min)

Remarque : la quantité disponible au jour j (c_{volj}) résulte de la quantité au jour précédent (c_{volj-1}) additionnée avec le remplissage prévu au jour j en fonction des habitudes de remplissage en fréquence et en volume pour chaque cuve.

2.1.2 Concepts

Manoeuvre. La manoeuvre diffère en fonction du contexte : lors de l'arrivée chez un client, elle prend la forme d'un pompage, tandis qu'au retour au dépôt, après un cycle de collecte, elle se traduit par une vidange.

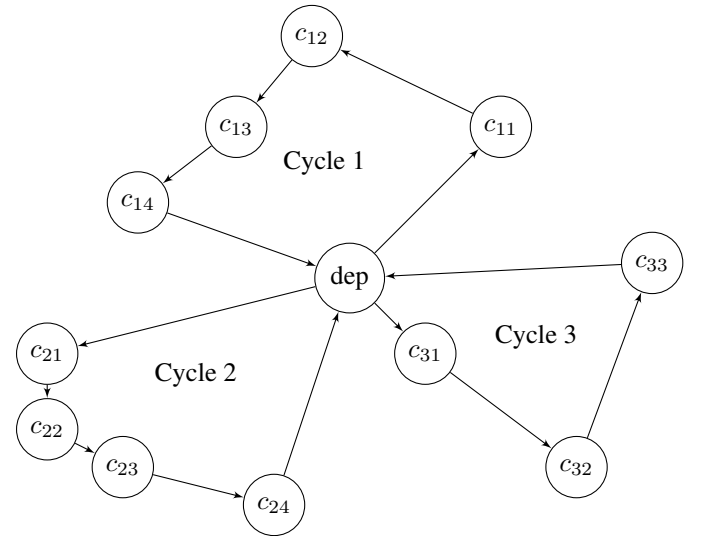
Cycle. C'est une liste ordonnée de cuves que l'agent de collecte visite en suivant un parcours défini, partant du dépôt et regagnant ce dernier après avoir récupéré des huiles usagées de chaque cuve.

Voyage. Il s'agit d'une solution au problème, définie comme une liste de cycles.

Ainsi, voici la modélisation de ces concepts :

$$\text{voyage} = [\text{cycle}_1, \text{cycle}_2, \dots, \text{cycle}_N]$$

$$\text{où : cycle}_i = [c_{i1}, c_{i2}, \dots, c_{in}]$$



Exemple : Voyage de $N=3$ cycles de tailles $n_1=4$, $n_2=4$ et $n_3=3$

2.1.3 Calculs des temps

Afin de collecter une cuve, nous avons besoin de connaître :

- le temps de la manoeuvre : t_{man}
- le temps de trajet : t_{trajet}

Dans le cas d'une arrivée chez un client (pompage), on a :

$$t_{man} = \frac{\lambda \times c_{volj}}{p_{vit}} \quad (1)$$

Dans le cas d'un retour au dépôt, on a :

$$t_{man} = \frac{vehicule_{vol}}{v_{vit}} \quad (2)$$

Remarque : Le paramètre λ , compris entre 0 et 1, représente l'effet de la troncation du volume récupéré en fonction du volume restant dans la cuve du véhicule, soit $(vehicule_{cap} - vehicule_{vol})$, et d'un seuil fixe par rapport à la capacité de la cuve c_{cap} .

Pour calculer le temps de trajet entre deux points (un point pouvant être un client ou un dépôt), on commence par calculer la distance entre ces deux points à l'aide de la formule d'Haversine :

$$dist_{point1,point2} = R \times c \quad (3)$$

où :

$$R = 6371.0 \quad (\text{Rayon de la Terre en km}) \quad (4)$$

$$c = 2 \times \text{atan2}(\sqrt{a}, \sqrt{1-a}) \quad (5)$$

$$a = \sin^2\left(\frac{\Delta lat}{2}\right) + \cos(lat1) \cos(lat2) \sin^2\left(\frac{\Delta lon}{2}\right) \quad (6)$$

$$\Delta lat = \text{radian}(lat2) - \text{radian}(lat1) \quad (7)$$

$$\Delta lon = \text{radian}(lon2) - \text{radian}(lon1) \quad (8)$$

Ainsi, nous pouvons calculer le temps de trajet entre deux points de la façon suivante :

$$t_{trajet_{point1,point2}} = K \times \frac{dist_{point1,point2} \times 60}{vehicule_{vit}} \quad (9)$$

Remarque : Le paramètre K , supérieur ou égal à 1, est introduit ici pour se rapprocher de la réalité, étant donné que nous nous basons sur une distance à vol d'oiseau pour calculer le temps de trajet. En pratique, nous utilisons $K = 2$ pour mieux estimer le temps réel de déplacement.

Par conséquent, il nous suffit maintenant d'additionner le temps de trajet, le temps de manœuvre, et la latence à partir d'une heure de départ spécifique pour déterminer l'heure de fin de la manoeuvre.

3 Algorithme Principal

L'algorithme principal conçu pour la planification des tournées de collecte d'huiles usagées s'appuie sur une série de filtres et de stratégies de sélection afin d'optimiser l'utilisation des ressources disponibles et d'assurer l'efficacité des opérations de collecte.

3.1 Principes de l'Algorithme

Dans un premier temps, notre algorithme génère un ensemble exhaustif de plages horaires de travail pour l'agent de collecte, prenant en considération ses contraintes individuelles telles que la durée maximale de travail et l'heure de début possible. Ces plages horaires représentent la gamme complète des configurations de travail possibles pour l'agent, englobant des scénarios où l'agent travaille

sans interruption, ainsi que des situations où il effectue deux sessions de travail séparées par une pause d'une durée définie.

Remarque : Il convient de noter que d'autres découpages pourraient être envisagés pour d'autres problèmes similaires, mais dans le cadre de notre étude de cas, ces alternatives n'étaient pas pertinentes.

Ensuite, les cuves disponibles font l'objet d'une première sélection basée sur les jours d'ouverture, afin de retenir celles qui sont accessibles le jour en question. Pour chaque configuration temporelle générée, l'algorithme se concentre de manière itérative sur l'optimisation des plages de travail de l'agent. Cette optimisation vise à identifier un large éventail de trajets potentiels respectant les contraintes établies, offrant ainsi une exploration exhaustive des différentes possibilités de planification des itinéraires.

Pour chaque cycle de collecte, l'algorithme entreprend un processus méticuleux pour la sélection des cuves à inclure. En premier lieu, il évalue les cuves accessibles, c'est-à-dire celles disponibles dans les plages horaires admissibles et atteignables dans un délai raisonnable par l'agent. Ensuite, parmi celles-ci, l'algorithme choisit celles qui présentent un niveau de remplissage suffisant, déterminé par le volume disponible dans la cuve mobile et un seuil prédéterminé, justifiant ainsi une collecte. Ce seuil est défini en tenant compte de la capacité du véhicule de collecte et des objectifs de collecte. Enfin, parmi les cuves atteignables et suffisamment remplies, l'algorithme identifie celles pouvant être prises en charge dans un laps de temps approprié, prenant en considération le trajet nécessaire pour se rendre à la cuve, récupérer les huiles usagées, et revenir au dépôt.

Lorsque des cuves deviennent inaccessibles en raison des contraintes d'horaires d'ouverture, et que le cycle actuel inclut déjà des cuves collectées, celui-ci est intégré au trajet. Ensuite, un nouveau cycle démarre à l'heure de fin du dernier cycle collecté. Cependant, si le cycle actuel est vide, indiquant qu'aucune cuve n'a encore été collectée, aucun cycle n'est ajouté au trajet. À la place, un nouveau cycle est lancé à la prochaine plage horaire disponible. Cette stratégie permet de maintenir la continuité des opérations de collecte tout en tenant compte des restrictions horaires liées aux cuves.

En outre, si les volumes à prélever des cuves ne sont pas assez importants par rapport à un seuil défini, le cycle courant est ajouté au voyage, et un nouveau cycle est programmé pour l'horaire de fin du dernier cycle collecté.

Enfin, si les cuves sont atteignables et que les volumes à prélever sont suffisants, mais que l'on ne peut pas garantir la collecte et le retour au dépôt dans le temps imparti, le cycle est ajouté au voyage et le voyage est clôturé.

3.2 Stratégies de Sélection

En cas de choix à faire parmi plusieurs cuves candidates, différentes stratégies de sélection sont disponibles. Parmi elles, on retrouve plusieurs options :

- **Aléatoire (R)** : Cette stratégie sélectionne une cuve candidate de manière aléatoire à chaque itération de l'algorithme.
- **Plus grand volume collectable (Q)** : Cette stratégie sélectionne la cuve ayant le plus grand volume collectable parmi les cuves candidates.
- **La plus urgente (E)** : Cette stratégie sélectionne la cuve considérée comme la plus urgente parmi les cuves candidates.
- **La plus proche (D)** : Cette stratégie sélectionne la cuve la plus proche du point de collecte actuel parmi les cuves candidates.
- **Heuristiques combinant les critères (HQE, HQD, HDE, HQDE)** : Ces heuristiques combinent différents critères, tels que le volume, la distance et l'urgence, pour évaluer les cuves candidates et sélectionner la meilleure option.

Prenons l'heuristique HQDE comme exemple. Cette heuristique combine les critères de volume, distance et urgence pour évaluer une solution candidate (un voyage). Le score de l'heuristique HQDE est calculé selon l'équation suivante :

$$\begin{aligned} \text{score}_{HQDE} = & w_q \times v_q \\ & + w_d \times v_d \\ & + w_e \times v_e \end{aligned} \quad (10)$$

où w_q , w_d et w_e sont les poids normalisés attribués aux critères de volume, distance et urgence respectivement, et v_q , v_d et v_e représentent les valeurs associées à ces critères sur un voyage, pour la solution candidate.

Remarque : Une méthode aléatoire (Random) peut être utilisée pour sélectionner aléatoirement une des stratégies présentées, à chaque itération de l'algorithme.

3.3 Pseudo-algorithme

Algorithme 1 : Algorithme principal : Génération d'un voyage

Result : Voyage

```

1 while processing_voyage do
2   Récupérer l'horaire de départ du cycle;
3   Instanciation du cycle;
4   while processing_cycle do
5     Filtrer les cuves atteignables selon les horaires
      d'ouverture;
6     if aucune cuve atteignable then
7       if cycle non vide then
8         Ajouter le cycle courant et lancer un
          nouveau cycle à la prochaine plage
          horaire;
9       else
10        Mettre à jour l'horaire de départ du
          prochain cycle et clore le cycle actuel;
11      end
12    else
13      Filtrer les cuves faisables selon le
        remplissage;
14      if aucune cuve faisable then
15        Ajouter le cycle courant et lancer un
          nouveau cycle à la prochaine plage
          horaire;
16      else
17        Filtrer les cuves candidates selon le
          temps de retour;
18        if aucune cuve candidate then
19          Ajouter le cycle courant et clore le
            voyage;
20        else
21          Sélection d'une cuve candidate
            selon la méthode de choix choisie;
22        end
23      end
24    end
25  end
26 end

```

3.4 Résultats

On se propose de projeter selon nos trois objectifs les 396 solutions que nous avons générées en combinant les 44 configurations temporelles avec les 9 méthodes de choix.

Comme illustré ci-dessous, nous obtenons différents fronts de Pareto [4]. Cependant, il est notable que le front de Pareto le plus fréquent est celui de la figure 1, correspondant à la méthode de choix D, qui se distingue nettement des autres méthodes. Ensuite viennent les méthodes heuristiques, notamment HQD et HQDE, suivies par la méthode E, les méthodes aléatoires Random et R, et enfin la méthode Q. Ces conclusions sont basées sur l'analyse de 3 semaines de solutions, totalisant $21 \times 396 = 8316$ observations.

Cependant, comme peut notamment le montrer la figure 2, il reste difficile de déterminer avec certitude la méthode qui prévaudra, car plusieurs choix se retrouvent souvent présents simultanément dans le front de Pareto. Ainsi, puisque tous les choix, même ceux qui se manifestent rarement, sont pris en compte dans le front à un moment donné, il semble essentiel de les conserver et d'explorer toutes les possibilités.

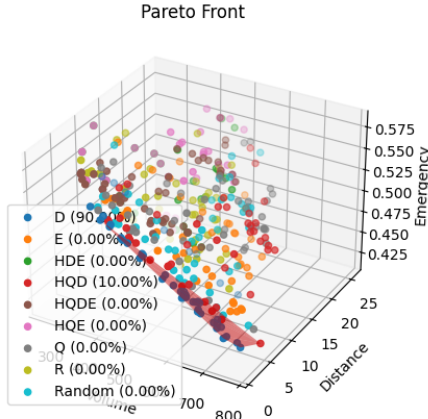


FIGURE 1 – Front de Pareto type (jour 1)

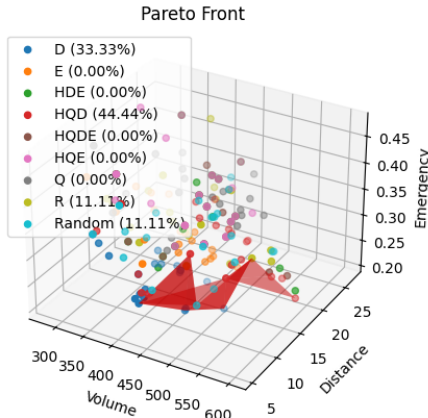


FIGURE 2 – Front de Pareto d'un cas particulier (jour 6)

Certains résultats obtenus peuvent sembler surprenants au premier abord. Par exemple, on pourrait s'attendre à ce que la méthode Q maximise la quantité collectée au détriment des autres objectifs, tandis que la méthode E minimise l'urgence au détriment des autres objectifs, et ainsi de suite.

Cependant, il est intéressant de noter que choisir une méthode différente, comme D par exemple, peut permettre de collecter davantage tout en réduisant l'urgence par rapport à d'autres choix.

Cette observation peut s'expliquer par les contraintes spécifiques de notre cas d'étude. En effet, la taille de la cuve mobile est limitée ($vehicule_{cap}$), ce qui signifie que la capacité du véhicule diminue à mesure qu'il collecte des huiles usagées. Ainsi, lorsque la cuve du véhicule se remplit, il devient nécessaire de tronquer les gros volumes afin de les rendre plus petits, ce qui peut influencer le choix de la méthode optimale dans certaines situations.

Pour illustrer ces observations, deux figures sont présentées ci-dessous. La première figure 3 met en évidence la prépondérance de la méthode D sur toutes les autres et correspond au front de Pareto de la figure 1. La seconde figure 4, met en lumière un cas particulier où la méthode D ne se positionne pas nécessairement comme la meilleure dans tous les objectifs, mais elle est néanmoins intégrée dans le front de Pareto de la figure 2.

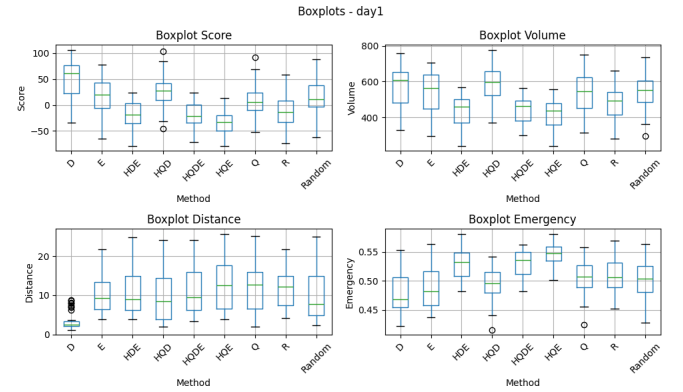


FIGURE 3 – Boîtes à moustaches types (jour 1)

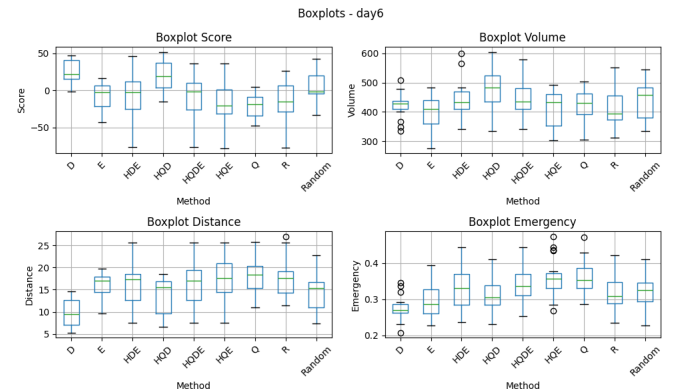


FIGURE 4 – Boîtes à moustaches d'un cas particulier (jour 6)

Avec la première partie de notre étude achevée, nous avons obtenu 396 solutions de base (sauf cas d'égalité rares) pour les tournées de collecte en utilisant des techniques algorithmiques avancées et en tenant compte des diverses contraintes telles que les fenêtres de temps ou bien le temps de travail de l'agent.

Pour évaluer ces solutions, nous avons utilisé une fonction d'évaluation pondérée par rapport à l'échelle de grandeur de chaque objectif, permettant ainsi de les comparer sur un pied d'égalité et de les ramener à un problème mono-objectif, facilitant ainsi la phase d'optimisation à venir.

4 Optimisation

Grâce à cette première partie, nous avons obtenu des solutions de base qui serviront de point de départ à la seconde phase de notre étude : l'optimisation des solutions obtenues.

Pour cela, nous utiliserons différentes méthodes de génération de voisins et des techniques d'optimisation telles que le **hill climbing** et le **recuit simulé**.

4.1 Génération de voisins

Dans le processus d'optimisation, la génération de voisins joue un rôle crucial. Nous proposons trois méthodes pour générer des voisins à partir d'une solution donnée.

Permutation. Dans cette méthode, nous sélectionnons une cuve disponible lors de la journée de collecte qui n'a pas été prise lors du voyage actuel. Ensuite, nous déterminons les positions dans le voyage auxquelles nous pouvons placer cette cuve, en tenant compte des horaires d'ouverture. Si le placement de la cuve respecte les contraintes de temps, nous l'ajoutons au voyage et lançons une optimisation pour compléter le cycle jusqu'à son horaire de fin.

Ainsi, nous avons :

- Un voyage de la forme :

$$\begin{aligned} \text{voyage} &= [cycle_1, cycle_2, \dots, cycle_N] \\ &= [[c_{11}, \dots, c_{1n_1}] \dots [c_{N1}, \dots, c_{Nn_N}]] \end{aligned}$$

- Un ensemble $\overline{\text{cuves}}$ tel que chaque élément représente une cuve disponible mais non collectée au jour de collecte. Formellement, nous pouvons l'écrire comme suit :

$$\overline{\text{cuves}} = \{c \in \text{cuves} \mid \forall cycle_i \in \text{voyage}, c \notin cycle_i\}$$

- On choisit une cuve $c_j \in \overline{\text{cuves}}$, nous déterminons les positions p_{ij} dans lesquelles nous pouvons placer la cuve dans un cycle du voyage, en tenant compte des contraintes de temps.

- Parmi ces positions possibles, nous choisissons une position p_{ij} et optimisons la suite du cycle $cycle_i$ après l'ajout de la cuve à la position p_{ij} . On a donc :

$$cycle_i = [c_{i1}, c_{i2}, \dots, c_{in}]$$

Après avoir ajouté la cuve c_j à la position p_{ij} dans le cycle $cycle_i$, le nouveau cycle résultant peut être représenté comme suit :

$$cycle'_i = [c_{i1}, \dots, c_{ip_{ij}}, c_j, c_{ip_{ij}+1}, \dots, c_{in}]$$

Remarque : En fin de compte, le voyage voisin obtenu est identique au voyage initial, à l'exception du cycle $cycle_i$ à partir de la position p_{ij} .

Échange. Dans cette méthode, nous identifions les couples de cuves dans le voyage (inter-cycles ou intra-cycle) pour lesquels un échange est possible tout en respectant les contraintes liées aux horaires d'ouverture.

Un échange entre deux cuves c_{ij} et c_{kl} dans un même cycle ($i = k$) ou deux cycles distincts ($i \neq k$) est possible si les conditions suivantes sont remplies :

- $c_{ij} \in cycle_i$ et $c_{kl} \in cycle_k$.

- Les heures d'ouverture de c_{ij} et de c_{kl} doivent permettre l'échange sans violer les contraintes de temps, c'est à dire que c_{ij} prenant la place de c_{kl} dans $cycle_k$ doit être ouverte à la collecte à l'heure prévue (et respectivement pour c_{kl} à la place de c_{ij} dans $cycle_i$).

- Dans les deux cas, l'échange doit également respecter les contraintes de volume : le volume de $cycle_i$ et le volume de $cycle_k$ ne doivent pas excéder $vehicule_{cap}$.

Après avoir identifié les couples de cuves échangeables et vérifié les conditions mentionnées ci-dessus, nous procédons à l'échange en mettant à jour les cycles concernés. Soit $cycle_i$ et $cycle_k$ les cycles concernés par l'échange, et p_{ij} et p_{kl} les positions respectives des cuves c_{ij} et c_{kl} dans ces cycles. Après l'échange, les nouveaux cycles $cycle'_i$ et $cycle'_k$ peuvent être représentés comme suit :

$$cycle'_i = [c_{i1}, \dots, c_{ip_{kl}-1}, c_{kl}, c_{ip_{kl}+1}, \dots, c_{in_i}]$$

$$\text{où } c_{ip_{kl}} = c_{kl}.$$

$$cycle'_k = [c_{k1}, \dots, c_{kp_{ij}-1}, c_{ij}, c_{kp_{ij}+1}, \dots, c_{kn_k}]$$

$$\text{où } c_{kp_{ij}} = c_{ij}.$$

Remarque : Les tailles des vecteurs entre $cycle_i$ et $cycle'_i$, ainsi qu'entre $cycle_k$ et $cycle'_k$, sont identiques. Après l'échange, la solution obtenue est optimisée en recalculant intelligemment les volumes sur les cycles concernés, tout en respectant les contraintes de volume et de temps.

Transfert. Dans cette méthode, nous supprimons une cuve sélectionnée de la solution dans la liste des cuves disponibles en entrée, puis nous relançons l'algorithme principal sans cette cuve. Cela conduit à l'obtention d'une nouvelle solution potentiellement meilleure.

Remarque importante : Nous avons défini deux modes de génération de voisins : la complexité totale et la complexité

faible. Dans la complexité totale, nous renvoyons tous les voisins possibles, tandis que dans la complexité faible, nous renvoyons un seul voisin par méthode.

4.1.1 Résultats

On se propose de présenter les résultats ici de la génération de voisins en commençant par afficher les fronts de Pareto pour trois premiers voyages générés avec l'algorithme principal en fonction des méthodes de génération de voisins employées ainsi que les boîtes à moustaches correspondantes.

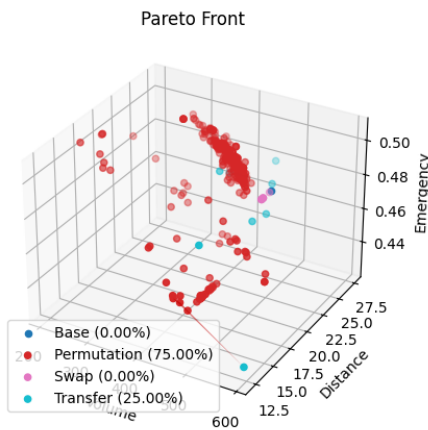


FIGURE 5 – Front de Pareto pour le voyage 1

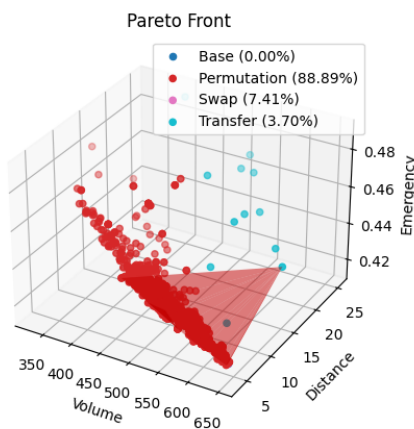


FIGURE 6 – Front de Pareto pour le voyage 2

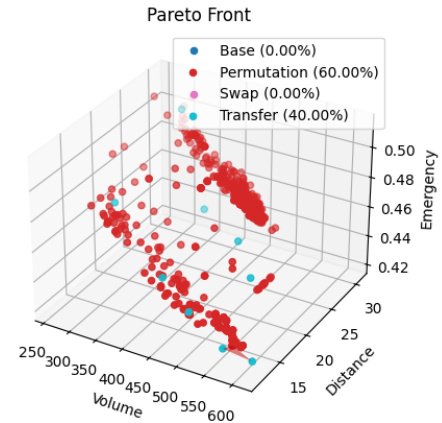


FIGURE 7 – Front de Pareto pour le voyage 3

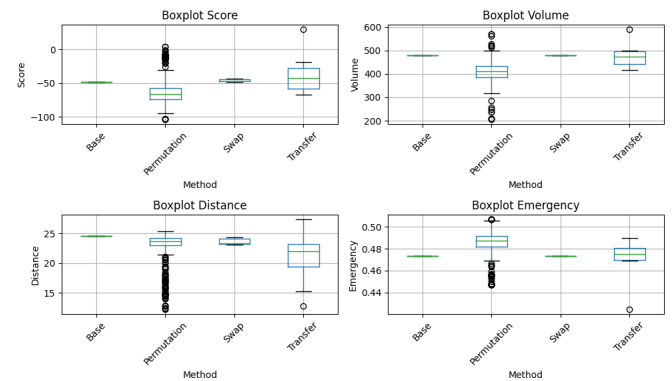


FIGURE 8 – Boîtes à moustaches pour le voyage 1

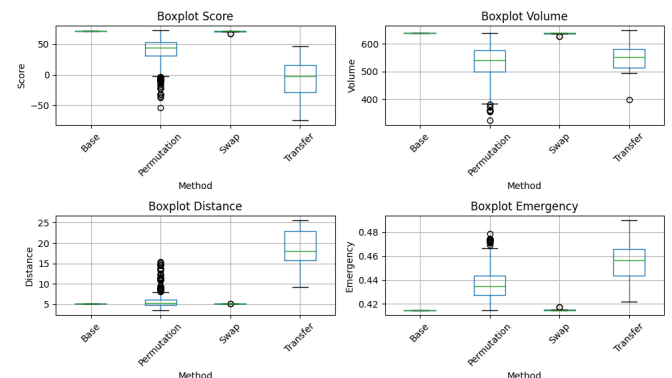


FIGURE 9 – Boîtes à moustaches pour le voyage 2

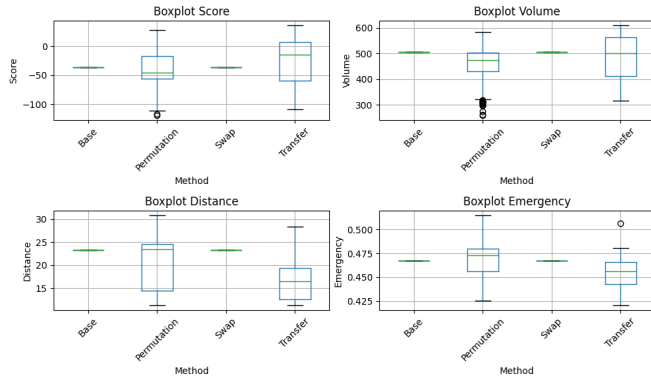


FIGURE 10 – Boîtes à moustaches pour le voyage 3

Les résultats de la génération de voisins pour les trois premiers voyages sont présentés ci-dessus. Globalement, nous observons une prédominance des voisins générés par permutation, suivis d'un nombre conséquent de voisins par transfert, et un nombre limité de voisins par l'échange. Cette observation s'explique par la complexité des contraintes liées aux horaires d'ouverture et par les seuils de remplissage des cuves, rendant difficile la réalisation d'échanges entre les cuves sans violer ces contraintes. De plus, il est important de noter que la méthode d'échange ne nécessite pas de recalcul du cycle après la réalisation de l'échange, à l'exception des quantités prélevées dans les cuves. Cela peut limiter son efficacité.

En analysant les boîtes à moustaches des voyages 1 et 3 (voir Figure 8 et Figure 10), nous constatons que les méthodes d'optimisation permettent souvent d'obtenir des solutions avec un score considérablement amélioré, en particulier pour certains objectifs. Cependant, le voyage 2 présente une tendance différente, où il est parfois difficile d'obtenir une amélioration significative par rapport à la solution de base, même si dans ce cas, une légère amélioration est possible (voir Figure 9). Ces deux tendances prédominantes indiquent la variabilité des performances des méthodes de génération de voisins en fonction des scénarios de collecte.

En examinant les fronts de Pareto des voyages 1 et 3 (voir Figure 5 et Figure 7), nous remarquons l'absence de solutions issues de l'échange, contrairement au voyage 2 (voir Figure 6). De plus, la supériorité des voisins générés par permutation dans le front est contestée par la méthode de transfert, suggérant que cette dernière peut également générer des solutions compétitives. Néanmoins, aucune tendance claire n'a été observée entre les méthodes en termes d'objectifs, soulignant ainsi l'importance de considérer toutes les techniques de génération de voisins dans l'optimisation des voyages.

4.2 Algorithmes d'Optimisation

Hill Climbing. Le hill climbing est une méthode d'optimisation locale qui consiste à explorer les voisins d'une solution actuelle et à choisir le voisin améliorant le plus la solution. Voici le pseudo-algorithme du hill climbing :

Algorithme 2 : Hill Climbing

Result : Solution optimisée

- 1 Initialiser la solution actuelle ; **while** la solution actuelle peut être améliorée **do**
 - 2 Générer tous les voisins de la solution actuelle ;
 Choisir le voisin améliorant le plus la solution actuelle ; Mettre à jour la solution actuelle ;
 - 3 **end**
-

Recuit Simulé. Le recuit simulé [5] est une méthode d'optimisation probabiliste qui permet d'éviter les optimums locaux en acceptant parfois des solutions dégradantes. Cela permet d'explorer l'espace de recherche de manière plus efficace. Voici le pseudo-algorithme du recuit simulé :

Algorithme 3 : Recuit Simulé

Result : Solution optimisée

- 1 Initialiser la solution actuelle ; Initialiser la température ; **while** la température n'a pas atteint le seuil minimum **do**
 - 2 Générer un voisin aléatoire de la solution actuelle ;
 Calculer la variation d'énergie entre la solution actuelle et le voisin ; **if** la variation d'énergie est négative **then**
 - 3 Accepter le voisin comme nouvelle solution ;
 - 4 **else**
 - 5 Accepter le voisin avec une probabilité décroissante en fonction de la température ;
 - 6 **end**
 - 7 Mettre à jour la température ;
 - 8 **end**
-

4.2.1 Résultats

Les résultats de la convergence des solutions avec les méthodes d'optimisation sont présentés pour le voyage 3 à travers trois graphiques.

La première figure (11) illustre la progression du score avec le Hill Climbing en complexité maximale, c'est-à-dire en générant tous les voisins possibles. Les deux autres graphiques (12 et 13) représentent respectivement la progression du score avec le Hill Climbing en complexité minimale (générant un seul voisin) et avec le Recuit Simulé en complexité minimale.

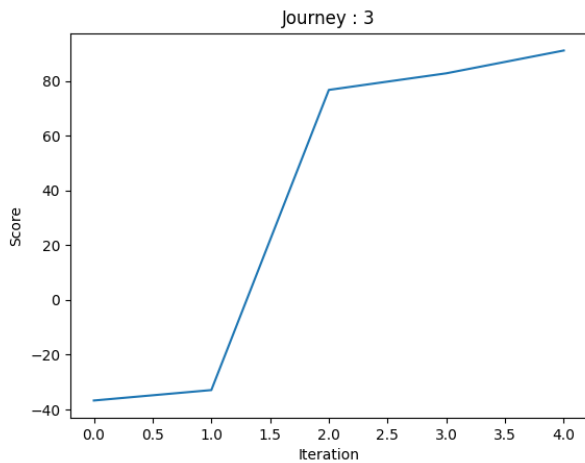


FIGURE 11 – Progression du score avec le Hill Climbing en complexité maximale pour le voyage 3

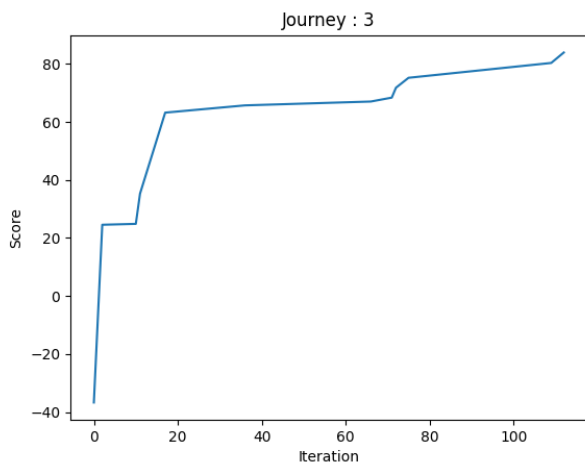


FIGURE 12 – Progression du score avec le Hill Climbing en complexité minimale pour le voyage 3

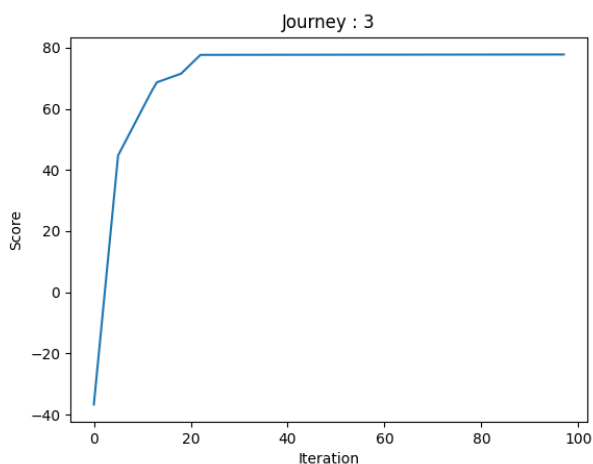


FIGURE 13 – Progression du score avec le Recuit Simulé en complexité minimale pour le voyage 3

En termes de temps d'exécution, la méthode à complexité maximale peut prendre jusqu'à 10 minutes sur un CPU pour atteindre la solution optimale, tandis que les méthodes à faible complexité, telles que le Hill Climbing et le Recuit Simulé, convergent vers des solutions approchées en une quinzaine de secondes, quel que soit le voyage en entrée. Ainsi, pour optimiser tous les voyages en faible complexité, il faut environ 1h30.

Les graphiques montrent que les méthodes d'optimisation parviennent à converger vers la solution optimale voire parfois à converger jusqu'à celle-ci, dans certains cas, avec les méthodes de génération de voisins actuelles. Ainsi, les méthodes du Hill Climbing en faible complexité et du Recuit Simulé semblent être des choix judicieux pour une optimisation rapide.

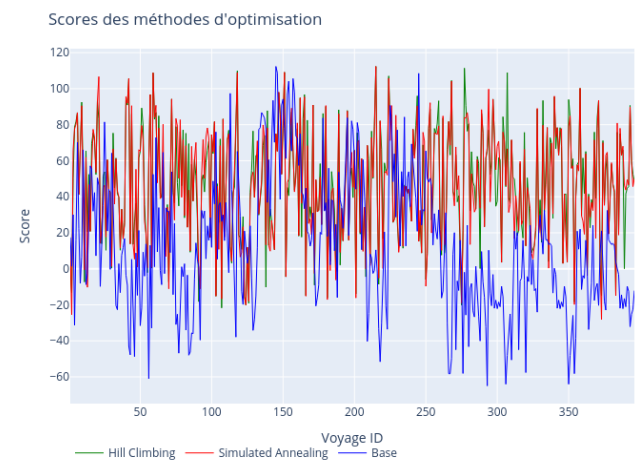


FIGURE 14 – Score maximal obtenu avec chaque méthode en complexité minimale pour tous les voyages

Le graphique 14 représente le score maximal obtenu avec chaque méthode d'optimisation en complexité minimale pour tous les voyages générés avant l'optimisation. On remarque que la méthode du Recuit Simulé semble atteindre plus de maximums que celle du Hill Climbing, bien que cette dernière soit parfois plus performante.

Cependant, bien que la plupart des voyages initiaux aient été notablement améliorés par les méthodes d'optimisation, il est important de noter que cette amélioration ne garantit pas nécessairement que les solutions optimisées seront finalement incluses dans le front de Pareto. Pour évaluer objectivement les performances des solutions générées par l'optimisation par rapport à celles générées par l'algorithme principal, une comparaison complète nécessite une projection dans l'espace des objectifs, ce qui est illustré par la figure suivante.

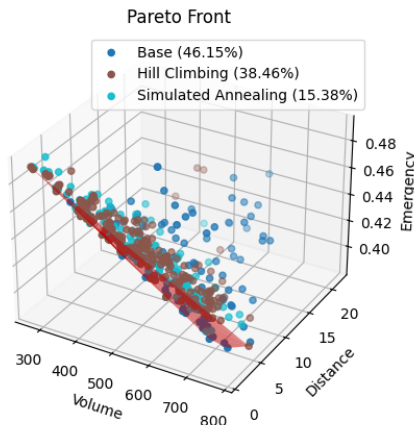


FIGURE 15 – Front de Pareto final avec tous les voyages optimisés

On observe que près de 45% des solutions appartenant au front de Pareto sont issues de l'algorithme principal, tandis qu'environ 55% proviennent des voyages améliorés par l'optimisation. Dans certains cas assez rares, aucune amélioration n'a pu être réalisée, et le score maximal obtenu reste identique à celui de la solution de base, ce qui signifie qu'aucun voisin n'a été généré ou qu'aucun des voisins générés n'améliore le score de la solution de base. Cependant, dans la plupart des cas, on parvient à améliorer considérablement les voyages initiaux grâce aux méthodes d'optimisation utilisées. Ce constat est confirmé par la figure 15, qui présente le front de Pareto final avec tous les meilleurs voyages.

5 Conclusion

En conclusion, notre approche hybride, combinant des techniques algorithmiques avancées avec des méthodes d'optimisation métaheuristiques, représente une contribution significative à la gestion efficace des huiles usagées dans la région de Toulouse. Cette approche a permis d'améliorer de manière importante les performances des tournées de collecte en tenant compte des contraintes spécifiques telles que les fenêtres de temps et les capacités des cuves. En intégrant trois méthodes de génération de voisins dans le cadre du Hill Climbing (HC) et du Recuit Simulé (SA), notre algorithme a pu explorer efficacement l'espace des solutions et obtenir des résultats optimaux dans des délais raisonnables. Ces résultats fournissent des renseignements précieux pour les gestionnaires de déchets et les décideurs politiques dans leur quête d'amélioration des pratiques de collecte d'huiles usagées, contribuant ainsi à la préservation de l'environnement et à la promotion de la sécurité publique.

6 Limites

Cependant, malgré les performances prometteuses de notre solution, quelques limites doivent être prises en compte. Premièrement, notre étude se concentre sur la région spécifique de Toulouse, ce qui rend impossible la généralisation de nos résultats à d'autres zones géographiques. De plus, notre modélisation simplifiée du problème, en ne considérant qu'un seul dépôt et un seul agent, peut ne pas capturer toutes les nuances et les complexités des systèmes de collecte dans des environnements plus vastes. De même, bien que notre approche hybride ait montré son efficacité dans l'optimisation des tournées de collecte, il reste des possibilités d'amélioration en explorant davantage les paramètres des méthodes d'optimisation et en développant des stratégies plus sophistiquées de génération de voisins. Enfin, l'évaluation des solutions obtenues est basée sur des critères prédéfinis et pondérés, ce qui peut ne pas refléter pleinement les préférences et les priorités des parties prenantes impliquées dans la gestion des huiles usagées. Ces limitations soulignent la nécessité de recherches futures pour affiner et étendre notre solution afin de mieux répondre aux défis complexes de la collecte des huiles usagées.

Remerciements

Nous tenons à exprimer notre sincère gratitude à nos encadrants, M. Le Nir et Mme. Jourdan, pour leur accompagnement tout au long de ce projet. Un remerciement spécial à M. Loubiere, dont l'expertise et les conseils éclairés ont été d'une grande valeur pour notre travail. Nous remercions également toute l'équipe impliquée et nos proches pour leur soutien constant.

Références

- [1] Fei Liu; Chengyu Lu; Lin Gui; Qingfu Zhang; Xiaoliang Tong; Mingxuan Yuanc, Heuristics for Vehicle Routing Problem : A Survey and Recent Advances *arxiv :2303.04147*, (Mar 1, 2023).
- [2] Mina Roohnavazfar; Seyed Hamid Reza Pasandideh; Roberto Tadei, A Hybrid Algorithm for the Vehicle Routing Problem with AND/OR Precedence Constraints and Time Windows *arXiv :2106.01652*, (Jun 3, 2021).
- [3] Breno Serrano; Alexandre M. Florio; Stefan Minner; Maximilian Schiffer; Thibaut Vidal, Contextual Stochastic Vehicle Routing with Time Windows, *arXiv :2402.06968* , (Feb 10, 2024).
- [4] Weitian Wu; Xinmin Yang, Obtaining properly Pareto optimal solutions of multiobjective optimization problems via the branch and bound method, *arXiv :2402.18015*, (Feb 28, 2024).
- [5] S. Kirkpatrick; C. D. Gelatt; M. P. Vecchi, Optimization by Simulated Annealing, *Science*, Vol. 220, No. 4598. (May 13, 1983), pp. 671-680.