

Optimisation des tournées et dimensionnement des équipes : Modèle appliqué

Dans ce chapitre, nous nous intéressons de nouveau au problème d'optimisation des tournées des véhicules de la logistique du CHRU, en considérant des hypothèses émises par le service de la logistique et d'autres contraintes rendant le problème plus proche de la réalité du terrain. Le tableau 4.1 résume les différences entre le modèle exploratoire qui vient d'être présenté dans le chapitre précédent et le modèle appliqué présenté dans ce chapitre. Dans ce modèle appliqué, les hypothèses ont permis de simplifier certains points. Elles ont été formulées pour obtenir des plannings des tournées les plus réalistes possibles. Une première section de ce chapitre présente ces nouvelles contraintes et hypothèses. Les algorithmes précédents ont été modifiés en fonction de ces nouveaux éléments et des perspectives d'amélioration établies dans le chapitre précédent ([119] et [122]). Ces modifications sont présentées dans une deuxième section. Puis une troisième section est dédiée à une nouvelle méthode de résolution : un algorithme mémétique [118]. Enfin, une dernière section est consacrée aux résultats expérimentaux sur la résolution de ce modèle appliqué par les trois algorithmes.

4.1 Contraintes et hypothèses du modèle appliqué

Tout d'abord, la première évolution du modèle concerne les tournées de collecte des chariots (linge sale et salubrité). A l'exception de Bretonneau, les tournées de collecte peuvent être vues comme des tournées de livraison fictives de chariots vides, sachant qu'une personne dépose autant de chariots vides qu'elle collecte de chariots pleins. A Bretonneau, la synchronisation des tournées de manutentionnaires et des tournées de collectes ne permet pas de considérer uniquement des tournées de type livraison (une tournée collectant des chariots à Bretonneau doit attendre que ces chariots soient collectés par les manutentionnaires). Cependant, nous considérons l'hypothèse que les collectes des chariots de

4.1. CONTRAINTES ET HYPOTHÈSES DU MODÈLE APPLIQUÉ

	Modèle exploratoire	Modèle appliqué
Contraintes de production ou de préparation des chariots aux dépôts	Aucune	Pour chaque flux, une date de préparation au plus tôt du premier chariot et un temps de production par chariot
Contraintes des capacités des véhicules	Borné par un volume et un nombre maximal de chariots	Borné par un nombre maximal de chariots
Contraintes des capacités des routes des manutentionnaires (fenwicks + pieds)	Borné par un poids et un nombre maximal de chariots	Borné par un nombre maximal de chariots
Contraintes de temps de déchargements/chargements des chariots à un quai	Identique quelque soit la place d'un quai	Dépend de la place du quai (haut ou bas)
Contraintes de collecte des chariots à Bretonneau	Attente des véhicules si les chariots à collecter ne sont pas encore collectés par les manutentionnaires	Hypothèse : les collectes sont réalisées la veille, les tournées de collectes sont modélisées par des livraisons fictives
Contraintes de temps de livraison/collecte	Temps de livraison par point et par flux ($tl_{i,p}$)	Temps de livraison plus détaillé : temps moyen d'accès au service, temps de déposer et dépaqueter le chariot et prise en compte de la capacité des ascenseurs.
Contraintes sur les flux à assurer à l'intérieur de Bretonneau par les manutentionnaires	Tous les flux sont à gérer par les manutentionnaires	Certains flux peuvent ne pas être pris en compte dans les tournées des manutentionnaires, mais seulement dans les tournées des chauffeurs

TAB. 4.1 – Différences entre le modèle exploratoire et le modèle appliqué

4.1. CONTRAINTES ET HYPOTHÈSES DU MODÈLE APPLIQUÉ

Bretonneau peuvent être réalisées la veille au soir pour une collecte par les véhicules le lendemain matin au quai. Pour modéliser cet aspect, il suffit de représenter les demandes à collecter à Bretonneau par des demandes de chariots à livrer avec des fenêtres de temps en fin d'après midi. En considérant uniquement des tournées de livraisons, la complexité de l'étape d'évaluation d'un individu, en particulier pour les tournées des manutentionnaires, diminue.

Comme le CHRU n'est pas certain de prendre en compte tous les flux dans le projet de la réorganisation de la logistique, il faut prévoir le cas où certains flux ne seront pas gérés, en particulier la salubrité. Pour ce cas, il suffirait de retirer dans les données toutes les demandes affectées à ce flux. Cependant, il existe actuellement une équipe appartenant à la salubrité qui se charge de collecter les chariots de linge sales et de déchets à Bretonneau. Cette équipe pourrait donc intervenir en parallèle de l'équipe de manutention, et collecterait à leur place le linge sale. Suite à ce souhait du CHRU, nous avons ajouté un attribut Br_p par type de produit p égale à 1 si le flux est à prendre en compte à l'intérieure de Bretonneau, 0 sinon. Les demandes appartenant à un flux p tel que $Br_p = 0$, seront considérées comme livrées au quai de Bretonneau, et le calcul des retards s'effectuera par rapport à leurs dates d'arrivée au quai et de leurs dates de fin livraison définies par $l_{i,p,t}$.

Après une étude approfondie du terrain de Bretonneau, il n'est finalement pas envisageable que les manutentionnaires puissent livrer certains bâtiments par fenwick extérieur. Comme présenté dans le chapitre 1 section 1.2.9, un sous-ensemble de bâtiments de Bretonneau peut être livré par des manutentionnaires (les cinq plus importants) parmi lesquels certains sont livrables uniquement par fenwick (étant donné la distance des couloirs souterrains et la possibilité d'augmenter le nombre de fenwicks), et un autre sous-ensemble peut être livré par véhicule (cf. figure 4.1). Nous avons donc changé les points de livraisons pour prendre en compte cet aspect et retiré la possibilité de livrer par fenwick extérieur.

D'autre part, la gestion des manutentionnaires et le calcul des tournées intra-Bretonneau ont été revus. Il paraît difficilement concevable de donner à chaque manutentionnaire la liste précise des chariots à livrer de chacune de ses tournées avec les dates à laquelle il doit partir du Quai. Nous avons donc décidé de déterminer les tournées intra en fonction du comportement qu'aurait un manutentionnaire vis-à-vis des chariots arrivant sur le quai. Ce comportement doit tenir compte des facteurs suivants : la priorité des chariots, des dates au plus tôt de livraison (un manutentionnaire ne prendra pas les chariots s'il sait qu'il va attendre dans le service), la destination des chariots (regroupement par bâtiment) et de la disponibilité des fenwicks. Nous détaillerons l'algorithme du comportement d'un manutentionnaire dans la section 4.2 suivante.

Un dernier point rectifié concerne le temps de livraison de chaque point $tl_{i,p}$ par les manutentionnaires ou les chauffeurs. Ce temps dépend du type de chariot pour différencier le temps de dépaquetages des chariots, et aussi des temps moyens d'accès aux services à livrer. Cependant, un élément non négligeable pour le calcul du temps de livraison d'un point est l'ascenseur. Seuls deux chariots peuvent être logés dans tous les ascenseurs des

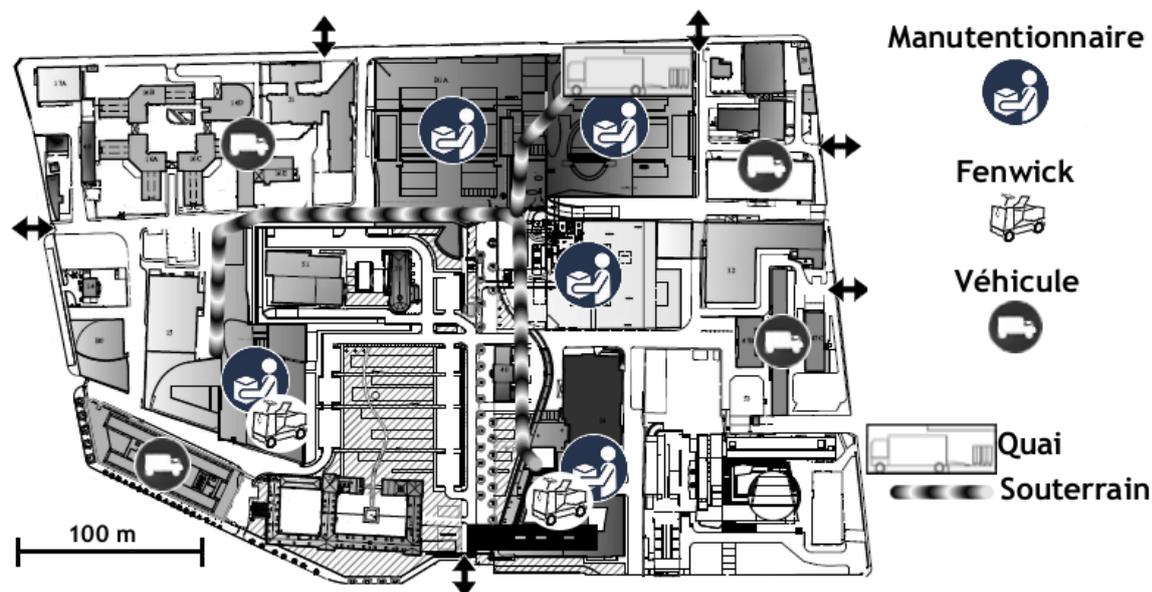


FIG. 4.1 – Plan de Bretonneau

hôpitaux. Lorsqu'un livreur arrive au pied du bâtiment, il ne peut monter les chariots que deux par deux. Pour modéliser cette nouvelle contrainte, nous remplaçons la donnée $tl_{i,p}$ par deux nouvelles :

- ta_i : le temps moyens d'accès aux services au point i .
- td_p : le temps de déposer ou dépaqueter si besoin, les chariots de produit p .

Lorsqu'un chauffeur arrive au pied d'un bâtiment d'un point i pour livrer en produits p le jour t une quantité $q_{i,p,t}$ de chariots, le temps total de livraison avec chargement et déchargement des chariots est égal à $q_{i,p,t} \times (2 \times tch_p + td_p) + 2 \times ta_i \times \lceil q_{i,p,t}/2 \rceil$. Le temps total de livraison pour un manutentionnaire est défini par la même expression sans le terme tch_p de chargement/déchargement des chariots du véhicule.

Enfin, les contraintes prises en compte dans ce modèle, contrairement au modèle précédent, sont les suivantes :

- un temps de production ou de préparation des chariots par type p modélisé par une heure de début au plus tôt h_p et une durée dp_p par chariot de production ou préparation.
- et un type de place pour chaque quai de chaque point de livraison. Ce type est soit "haut" soit "bas", et influence le temps de chargement et déchargement des chariots à un quai qui ne dépend plus d'uniquement du type de produit (le type haut permettant de charger et décharger plus rapidement). Lorsqu'un véhicule arrive à un quai, il mobilisera en priorité une place de type haut.

4.2 Impacts sur les algorithmes existants

L'une des plus grosses modifications des deux méta-heuristiques développées pour le précédent modèle concerne l'évaluation d'une solution pour prendre en compte les temps de production, les types de places et le fait que certains flux véhiculés par les camions ne sont pas pris en charge par les manutentionnaires.

Pour la simulation du comportement des manutentionnaires, nous avons mis en place un nouvel algorithme. Ce dernier est plus simple à expliquer que celui du modèle précédent puisqu'il n'intègre pas de tournées de type collecte. L'algorithme consiste à exécuter une procédure décrite par la figure 4.2 (les constantes ϵ , α et λ sont des paramètres, $\epsilon \approx 15$ min., $\alpha \approx 3$ heures, $\lambda \approx 5$ min). Cette procédure est exécutée à un instant t lorsque le nombre de tournées intra se déroulant au même moment est inférieur à NbT_{intra} et des chariots attendant sur le quai de Bretonneau peuvent être livrés. La sélection des chariots doit se faire toujours en respectant la **condition 1** définie par :

- $l_{i,p,t}$ est la plus petite valeur parmi tous les chariots sur le quai à l'instant t
- $e_{i,p,t} < t + \epsilon$ et $l_{i,p,t} < t + \alpha$

L'algorithme génétique possède la même structure que celle décrite à la section 3.3 du chapitre précédent. Cependant la génération de la population initiale proposée en section 3.3.2 a été améliorée de manière à mieux répartir les chariots dans les véhicules. Les N individus de cette population sont générés de la manière suivante :

- Découper la période considérée τ en plusieurs demi-journées.
- Pour les demandes attribuées à un jour particulier, affecter chaque demande dont l'intersection entre la fenêtre de temps correspondante et la demi-journée est la plus grande parmi les deux demi-journées concernées.
- Affecter aléatoirement les autres demandes (non attribuées à un jour) aux demi-journées.
- Enfin, pour chaque demi-journée :
 - Trier aléatoirement les demandes affectées à cette demi-journée.
 - Affecter un numéro de véhicule pour chaque demande de manière suivante :
 - Trier aléatoirement dans une liste les véhicules.
 - Prendre le premier véhicule de cette liste, et lui affecter les x premières demandes jusqu'à atteindre sa capacité maximum.
 - Positionner ce véhicule à la fin de la liste et réitérer à l'étape précédente s'il reste des demandes à affecter dans la demi-journée considérée.

Nous avons également modifié l'opérateur de croisement type PMX de telle sorte que les deux sous-segments échangés entre les deux individus soient positionnés au même niveau dans le codage des deux solutions. La recherche tabou possède, elle aussi, la même structure que celle décrite à la section 3.4 du chapitre précédent.

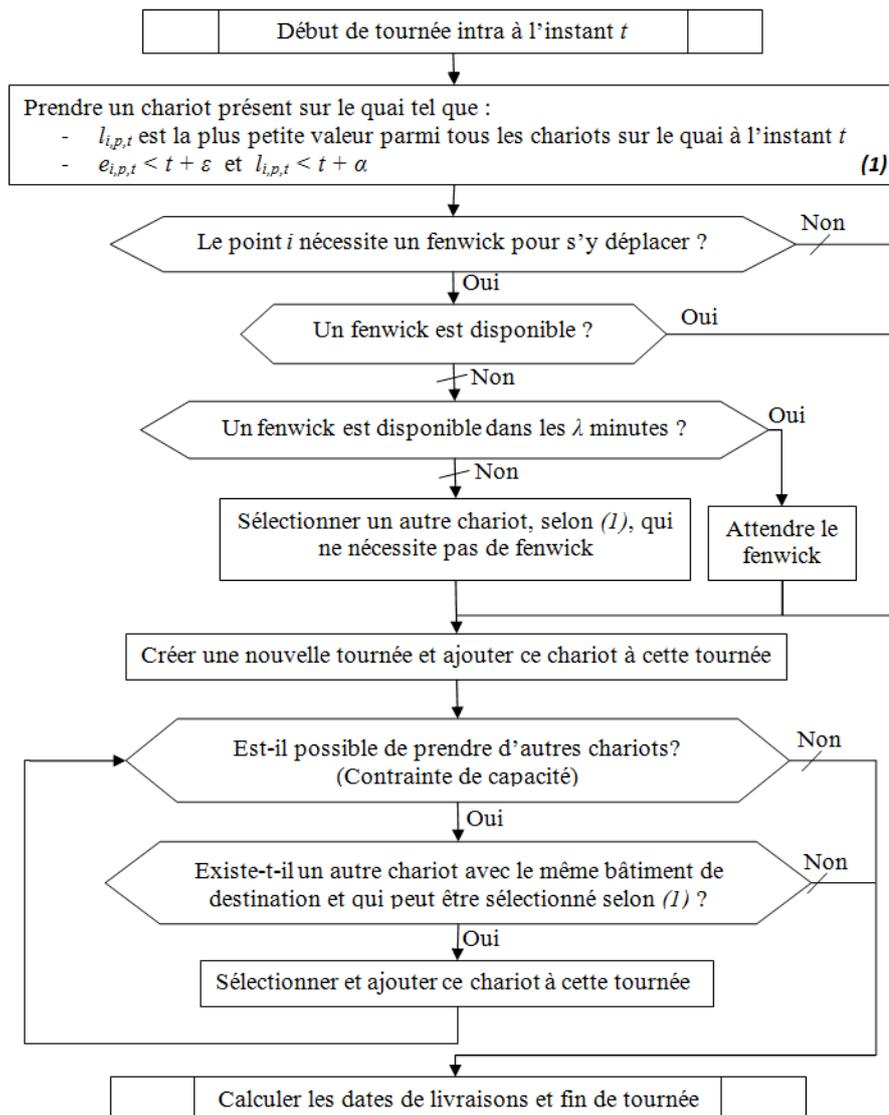


FIG. 4.2 – Comportement des manutentionnaires

4.3 Algorithme mémétique

Le terme algorithme mémétique (Moscato [149]) est souvent utilisé pour désigner les méthodes de résolution qui hybrident une recherche à base de population, comme un algorithme évolutionnaire et une recherche par voisinage, comme une recherche locale. L'algorithme mémétique, décrit ici, est le résultat d'une hybridation des deux méthodes précédentes : l'algorithme génétique et la recherche tabou. La structure générale de cette hybridation est présentée par l'algorithme 3. Nous avons ajouté à l'algorithme génétique présenté à la section 3.3, une étape d'amélioration de la population qui fait appel à la recherche tabou. Cette recherche tabou est exécutée sur chaque individu de la population courante avec une probabilité P_{TS} . Le nombre d'itérations de la recherche tabou dépend de l'itération courante I de l'algorithme génétique, il est fixé à $I/4$. Le critère d'arrêt de l'algorithme mémétique est un nombre d'itérations sans amélioration $\#ite$ de la meilleure solution trouvée.

L'algorithme mémétique utilise le même codage d'une solution que les deux précédents algorithmes développés. La construction de la population initiale, les opérateurs de sélection, de croisement et mutation sont les mêmes que l'algorithme génétique. Après l'étape de mutation, l'algorithme tabou est exécuté sur chaque individu de la population avec une probabilité P_{TS} . Seul un ensemble d'individus de la population est donc amélioré par la tabou avec un nombre d'itérations égal à $\lfloor I/4 \rfloor$. Le calcul du voisinage d'une solution et la gestion de la liste tabou pour l'étape d'amélioration des individus sont identiques à ceux de la recherche tabou. L'application systématique de la recherche tabou à tout individu demanderait trop de temps, ce qui justifie l'utilisation d'une probabilité P_{TS} . Le choix de limiter le nombre d'itérations de la recherche tabou à $I/4$ permet de répartir les rôles entre elle et l'algorithme génétique. Initialement c'est l'algorithme génétique qui prévaut, puis au fil des itérations $I/4$ augmente et la recherche tabou influence de plus en plus la recherche.

4.4 Résultats expérimentaux

Avant de présenter les résultats expérimentaux sur ces trois algorithmes, nous allons décrire la méthode de génération des nouvelles instances. Comme pour le chapitre précédent, les instances ont été générées en fonction de données réelles fournies par l'hôpital. Le langage utilisé pour l'implémentation des algorithmes est le $C++$. L'ordinateur utilisé est un Intel(R) 2 Core (TM) 2 Duo CPU 3.00 GHz, 3.48 Go de RAM.

4.4.1 Génération d'instances

Des nouvelles instances ont été générées de manière à prendre en compte les nouvelles contraintes et hypothèses de ce modèle appliqué. La méthode de génération est identique à celle du précédent modèle : une instance théorique la plus proche possible du réel est créée, puis plusieurs instances sont générées à partir de celle-ci. L'instance théorique a donc été conçue à partir de données réelles du CHRU. Même si des nouvelles données étaient plus complètes que pour la précédente version, certaines quantités de chariots demandées avec

Algorithme 3 Structure générale de l'algorithme mémétique

```
1:  $\mathcal{P}_{courant} \leftarrow$  Initialiser une population de  $N$  individus
2: Evaluer chaque individu de  $\mathcal{P}_{courant}$ 
3:  $S_{best} \leftarrow$  Le meilleur individu  $S \in \mathcal{P}_{courant}$ 
4:  $I \leftarrow 0$ 
5: Tant que  $I < \#ite$  faire
6:    $\mathcal{P}_{enfant} \leftarrow \emptyset$ 
7:   Pour  $j = 0$  à  $j = N/2$  faire
8:      $(P_1, P_2) \leftarrow$  Sélectionner deux individus parents de  $\mathcal{P}_{courant}$ 
9:      $(E_1, E_2) \leftarrow$  Croiser les deux parents  $(P_1, P_2)$  pour obtenir deux individus enfants
10:     $\mathcal{P}_{enfant} \leftarrow \mathcal{P}_{enfant} \cup E_1 \cup E_2$ 
11:   Fin pour
12:   Muter aléatoirement des individus de la population  $\mathcal{P}_{enfant}$ 
13:   Pour Chaque individu  $ind$  de  $\mathcal{P}_{enfant}$  faire
14:     Si une amélioration est décidée (décision prise avec la probabilité  $P_{TS}$ ) alors
15:        $S_{courant} \leftarrow$  Solution de  $ind$ 
16:       Liste tabou  $\leftarrow \emptyset$ 
17:       Pour  $j = 0$  à  $j = \lfloor I/4 \rfloor$  faire
18:          $\Psi \leftarrow$  Calculer le voisinage non tabou de la solution de  $S_{courant}$ 
19:          $S_{courant} \leftarrow$  Meilleure solution du voisinage  $\Psi$ 
20:         Stocker le mouvement dans la liste tabou
21:       Fin pour
22:     Fin si
23:   Fin pour
24:    $\mathcal{P}_{courant} \leftarrow \mathcal{P}_{enfant}$ 
25:   Evaluer chaque individu de  $\mathcal{P}_{courant}$ 
26:   Si il existe un individu  $S \in \mathcal{P}_{courant}$  meilleur que  $S_{best}$  alors
27:      $S_{best} \leftarrow S$ 
28:      $I \leftarrow 0$ 
29:   Fin si
30:    $I \leftarrow I + 1$ 
31: Fin Tant que
```

4.4. RÉSULTATS EXPÉRIMENTAUX

leurs fenêtres de temps ont du être estimées à partir des tournées existantes ou de distributions empiriques données par le CHRU. Les caractéristiques de cette instance théorique sont les suivantes :

- L'ensemble \mathcal{D} est constitué de 4 dépôts (Logipôle, Blanchisserie, Salubrité à Bretonneau et les Archives près de Trousseau).
- L'ensemble \mathcal{H} est composé de 9 hôpitaux (Trousseau 1, Trousseau 2, Bretonneau, Clocheville 1, Clocheville 2, Ermitage, CPTS et CPU) et de 7 bâtiments de Bretonneau livrables par véhicules. Les quais de Trousseau possèdent 6 places de type haut, Bretonneau en dispose de 2 de type haut et 2 de type bas, les quais de Clocheville et d'Ermitage sont composés 2 places de type bas, CPU et CPTS en disposent seulement d'un de type bas. Tous les bâtiments de Bretonneau sont composés d'une seule place de type bas sauf un bâtiment qui en possède un de type haut.
- L'ensemble \mathcal{B} est constitué de 4 bâtiments à Bretonneau livrables uniquement par manutentionnaire.
- La période de planification τ est toujours de 5 jours, les demandes affectées au sixième jour sont celles qui ne sont pas assignées à un jour fixe dans la semaine.
- L'ensemble \mathcal{P} est composé de 11 types de chariots ou flux (LH, Pharmacie, repas du midi, repas du soir, linge propre 1, linge propre 2, linge sale 1, linge sale 2, la stérilisation, les dossiers patients et les déchets). Les caractéristiques de ces flux sont notés dans le tableau 4.2. Nous supposons que l'ensemble des flux sont pris en charge par les manutentionnaires de Bretonneau ($\forall p \in \mathcal{P}, Br_p = 1$).

Nom	Dépôt	td_p (min)	g_p (min)	h_p	dp_p (min)
LH	Logipôle	10	-	07 : 00	1
Pharmacie	Logipôle	1	-	07 : 00	3
Repas midi	Logipôle	1	200	06 : 30	2
Repas soir	Logipôle	1	200	13 : 00	2
Linge Propre 1	Blanchisserie	1	-	07 : 00	2
Linge Propre 2	Blanchisserie	1	-	07 : 00	2
Linge Sale 1	Blanchisserie	1	-	06 : 00	0
Linge Sale 2	Blanchisserie	1	-	06 : 00	0
Stérilisation	Logipôle	1	-	07 : 00	1
Dossiers patients	Archives	20	-	07 : 00	10
Déchets	Salubrité	1	-	07 : 00	1

TAB. 4.2 – Flux logistiques de l'instance théorique

- L'ensemble des véhicules \mathcal{V} est constitué de 16 camions dont la capacité est 12 chariots pour un véhicule, 16 pour 4, 20 pour 2 et 21 pour 9.
- Il y a au total 4 fenwicks intérieurs qui peuvent tracter 5 chariots. Les fenwicks permettent de se déplacer deux fois plus vite qu'à pied.
- L'ensemble \mathcal{R} est composé de 22 personnes à répartir entre l'équipe des chauffeurs et celle des manutentionnaires. Un manutentionnaire peut tracter en moyenne 2 chariots.

4.4. RÉSULTATS EXPÉRIMENTAUX

Les temps de chargement et déchargement, le temps de livraison par point, l'ensemble des produits transportables suivant le moyen de transport, ainsi que les durées de déplacements entre hôpitaux et au sein de Bretonneau sont celles fournies par le CHRU. Comme pour la première partie du chapitre, pour les quelques flux dont le CHRU nous a fourni les distributions empiriques, nous avons retenu pour chaque service une valeur haute du nombre de chariots (correspondant à une probabilité cumulée de l'ordre de 90%). Pour les autres flux, nous nous sommes basés sur les tournées existantes pour estimer le nombre de chariots constituant chaque demande. Il en est de même pour les fenêtres de temps. Cette instance compte au total 680 demandes pour 2775 chariots. La répartition totale des chariots sur les différents points de livraisons est représentée par la figure 4.3 (2775 chariots pour les tournées inter et 793 chariots pour les tournées intra).

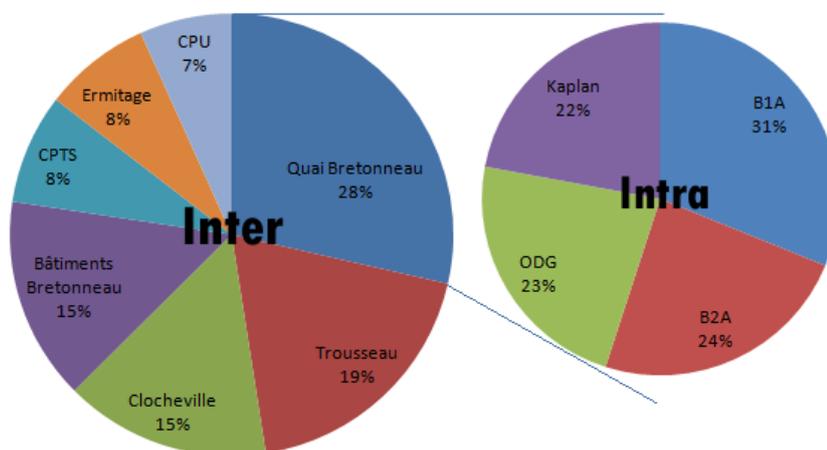


FIG. 4.3 – Répartition des chariots par point de livraison

Le tableau 4.3 présente par jour une borne inférieure du temps total des livraisons à effectuer pour les tournées intra-Bretonneau et inter-hôpitaux (le jour 5 désigne les demandes affectées à aucune journée). Ces bornes inférieures sont obtenues en effectuant la somme des éléments suivants :

- les temps de dépaquetages des chariots de toutes les demandes,
- les temps de chargements et déchargements de tous les chariots sur les quais les plus rapides pour cette opération, dépendant donc de leurs points de livraison,
- les temps moyens d'accès aux services de toutes les demandes suivant leurs points de livraison et modulo 2 chariots (contrainte des ascenseurs),
- et les temps d'aller/retour minimum entre les dépôts et les points de livraison en prenant les moyens de transports ayant les plus grandes capacités.

A partir de cette instance théorique, cent instances ont été générées pour réaliser les tests. Les quantités de chariots des demandes de l'instance théorique étant surestimées, ces instances ont été construites en diminuant aléatoirement certaines quantités de chariots demandées de 1 à 2 suivant que le nombre théorique est inférieur ou supérieur à 4. Les fenêtres de temps sont également modifiées aléatoirement, la date de livraison au plus tôt et

4.4. RÉSULTATS EXPÉRIMENTAUX

Jour t	Tournées inter (min)	Tournées intra (min)
0	4608	1157
1	4398	1195
2	4329	1068
3	4317	1175
4	4458	1059
5	1051	314

TAB. 4.3 – Bornes inférieures du temps total des livraisons

la date de livraison au plus tard sont décalées de plus ou moins une heure indépendamment.

4.4.2 Algorithme génétique

Après des tests préliminaires, la taille de la population a été fixée à $N = 200$, la probabilité de mutation P_{mut} à 60% et dans le cas d'une mutation l'opérateur agissant sur le numéro de véhicule est choisi avec une probabilité de 40% contre 60% pour l'opérateur agissant sur le déplacement d'un gène. Les paramètres importants correspondant au nombre maximal des tournées inter-hôpitaux et intra-Bretagne se déroulant au même moment ont été fixés à $NbT_{inter} = 8$ et $NbT_{intra} = 3$. Comme pour le chapitre précédent, nous nous avons ensuite étudié l'influence de l'opérateur de sélection (sélection des 50% meilleurs (M), sélection par roulette (R) et sélection par tournoi (T)) et l'influence de l'opérateur de croisement (par recopie (C) ou par insertion (I)). Un couple (sélection, croisement) est encore appelé une configuration. Les tableaux 4.4 et 4.5 résument les résultats sur 100 instances de chacune des six configurations pour $\#ite = 200$ et $\#ite = 400$. Δ_i est la moyenne sur toutes les instances des écarts relatifs de la solution trouvée par la configuration i à la meilleure solution trouvée par l'ensemble des 6 configurations. L'écart type est indiqué par $\sigma(\Delta_i)$. Nb_{min} est le nombre de fois où la configuration i trouve la meilleure solution et Tps indique le temps moyen de résolution en secondes.

Crois.	C	C	C	I	I	I
Sélec.	M	R	T	M	R	T
Δ_i (%)	4,5	70,8	16,4	90,8	92,2	90,1
$\sigma(\Delta_i)$	9,4	7,1	15,7	1,9	1,5	2,0
Nb_{min}	72	0	28	0	0	0
Tps (s)	3177,35	2016,48	5343,24	681,63	315,88	301,41

TAB. 4.4 – Résultat pour l'AG avec $\#ite = 200$

L'opérateur de croisement I semble être beaucoup moins performant que l'opérateur C quelque soit l'opérateur de sélection et le nombre d'itérations sans amélioration. Tandis que les opérateurs de sélection M et T semblent largement dominer l'opérateur R. Enfin, parmi les deux configurations dominantes (C, M) et (C, T), (C,M) permet d'obtenir de

4.4. RÉSULTATS EXPÉRIMENTAUX

Crois.	C	C	C	I	I	I
Sélec.	M	R	T	M	R	T
Δ_i (%)	7,3	68,5	13,4	91,0	92,7	90,5
$\sigma(\Delta_i)$	11,0	6,4	16,0	1,6	1,4	1,8
Nb_{min}	56	0	44	0	0	0
Tps (s)	4881,26	3326,59	5140,86	1249,85	598,16	537,91

TAB. 4.5 – Résultat pour l'AG avec $\#ite = 400$

meilleures solutions quelque soit le nombre d'itérations sur la moyenne des écarts relatifs. Par rapport aux résultats expérimentaux du chapitre précédent, l'opérateur de croisement I est devenu inutile.

Les tableaux 4.6 et 4.7 présentent une moyenne sur les 100 instances de la fonction objectif, de la somme des retards en minutes, de la somme des dépassements des autonomies des chariots repas, des bornes inférieure et supérieure du nombre de manutentionnaires, des bornes inférieure et supérieure du nombre de chauffeurs, du nombre de demandes en retard et du nombre de demandes en chariot repas pour lesquelles l'autonomie d'au moins un chariot a été dépassée.

Crois.	C	C	C	I	I	I
Sélec.	M	R	T	M	R	T
Fonction objectif	2226,12	7371,24	2591,63	23046,34	26933,70	21250,97
Somme des retards	1997,84	7093,34	2356,03	22613,62	26427,77	20847,08
Somme des dépassements des autonomies	11,88	49,90	16,20	195,22	267,48	166,89
Borne inférieure (manutentionnaires)	5,98	5,99	5,98	6,04	6,02	6,01
Borne supérieure (manutentionnaires)	6,00	6,00	6,00	6,08	6,02	6,03
Borne inférieure (chauffeurs)	14,35	15,43	14,79	16,00	15,76	15,90
Borne supérieure (chauffeurs)	16,95	18,18	17,11	19,38	19,89	19,46
Nombre de demandes en retard	39,11	97,39	45,31	161,11	197,70	155,43
Nombre de demandes avec autonomie dépassée	1,19	3,20	1,31	7,41	9,71	7,13

TAB. 4.6 – Résultats des six configurations de l'AG avec $\#ite = 200$

Les tableaux 4.6 et 4.7 confirment la dominance des configurations (C,M) et (C,T) sur les autres configurations. Nous pouvons également noter que lors de l'évaluation d'une

4.4. RÉSULTATS EXPÉRIMENTAUX

Crois.	C	C	C	I	I	I
Sélec.	M	R	T	M	R	T
Fonction objectif	2168,09	6364,21	2342,84	22065,21	27018,94	20784,64
Somme des retards	1939,24	6085,55	2112,88	21666,30	26530,49	20379,70
Somme des dépassements des autonomies	12,75	51,41	13,11	161,81	249,70	168,39
Borne inférieure (manutentionnaires)	5,89	6,00	5,96	6,02	6,02	6,04
Borne supérieure (manutentionnaires)	6,00	6,00	6,00	6,06	6,05	6,12
Borne inférieure (chauffeurs)	14,41	15,43	14,43	16,05	15,81	15,95
Borne supérieure (chauffeurs)	16,92	18,02	16,98	19,29	19,87	19,20
Nombre de demandes en retard	37,55	89,03	41,43	156,21	196,97	152,77
Nombre de demandes avec autonomie dépassée	1,20	3,42	1,33	6,67	9,22	6,96

TAB. 4.7 – Résultats des six configurations de l'AG avec $\#ite = 400$

solution, la borne supérieure et la borne inférieure du nombre de manutentionnaires sont égales dans la plupart des cas. Cependant, la différence entre la borne supérieure et la borne inférieure du nombre de chauffeurs est souvent égale à deux dans la plupart des cas. L'influence du nombre d'itérations sans amélioration n'est pas importante pour la configuration (C, M), contrairement à la configuration (C, T) où une plus grande amélioration des solutions est visible. Cette configuration permettrait d'obtenir les mêmes résultats que (C, M) mais convergerait plus lentement. Une nette amélioration est aussi notable pour la configuration (C, R).

4.4.3 Algorithme tabou

Nous avons testé l'algorithme tabou sur ces 100 instances de la même manière que l'algorithme génétique. Les différentes combinaisons testées dépendent de trois paramètres : le nombre d'itérations sans amélioration $\#ite$ (5000 et 10000), l'opérateur de voisinage de type déplacement d'une demande Op_{mov} (60 et 80) et la taille de la liste tabou T_{lt} (1, 25, 50, 75 et 100). Les tableaux 4.8 et 4.9 présentent les mêmes indicateurs que pour les tests de l'algorithme génétique : la moyenne Δ_i des écarts relatifs entre la meilleure solution trouvée pour toutes les combinaisons et la meilleure solution trouvée par la configuration i , l'écart type $\sigma(\Delta_i)$, le nombre d'instances pour lesquelles la combinaison i trouve la meilleure solution (Nb_{min}) et le temps moyen de résolution en secondes (Tps). Pour une comparaison plus fine, nous avons ajouté le nombre d'instances pour lesquelles l'écart relatif est inférieur à 10% ($Nb_{\Delta \leq 10}$) et 20% ($Nb_{\Delta \leq 20}$).

4.4. RÉSULTATS EXPÉRIMENTAUX

Op_{mov}	60	60	60	60	60	80	80	80	80	80
T_{lt}	1	25	50	75	100	1	25	50	75	100
$\Delta_i(\%)$	19,7	16,0	21,1	29,0	33,3	19,9	13,3	20,0	24,6	29,4
$\sigma(\Delta_i)(\%)$	16,8	12,3	13,0	12,0	13,0	15,0	12,0	14,0	14,2	12,6
Nb_{min}	19	15	10	0	0	15	19	14	6	2
$Tps(s)$	530,8	960,9	759,1	725,4	725,8	618,2	887,7	908,0	895,2	853,4
$Nb_{\Delta \leq 10}$	37	37	23	6	5	31	46	27	18	8
$Nb_{\Delta \leq 20}$	57	60	47	21	19	50	74	50	36	22

TAB. 4.8 – Résultats pour la recherche tabou avec $\#ite = 5000$

Op_{mov}	60	60	60	60	60	80	80	80	80	80
T_{lt}	1	25	50	75	100	1	25	50	75	100
$\Delta_i(\%)$	19,2	13,8	22,8	28,7	33,2	19,9	12,8	19,9	26,1	33,2
$\sigma(\Delta_i)(\%)$	14,3	11,2	13,2	13,6	13,0	14,6	12,4	14,0	13,8	13,6
Nb_{min}	15	23	4	3	0	12	27	14	3	0
$Tps(s)$	1076	1441	1429	1484	1381	1022	1666	1824	1809	1707
$Nb_{\Delta \leq 10}$	28	39	21	10	3	27	49	26	11	6
$Nb_{\Delta \leq 20}$	52	73	39	27	16	51	68	51	33	16

TAB. 4.9 – Résultats pour la recherche tabou avec $\#ite = 10000$

Les résultats montrent globalement peu de différences pour la valeur de l'opérateur de voisinage. Néanmoins, une faible taille de la liste tabou est préférable à une grande taille. Nous pouvons remarquer que deux combinaisons semblent dominer les autres : ($Op_{mov} = 60, T_{lt} = 25$) et ($Op_{mov} = 80, T_{lt} = 25$) avec un temps de résolution moins important pour la première combinaison. Ensuite, comme pour l'algorithme génétique, nous avons relevé sur les 6 meilleures combinaisons (en enlevant les combinaisons avec des tailles de liste tabou égales à 74 et 100) la moyenne sur 100 instances de la fonction objectif, de la somme des retards en minutes, de la somme des dépassements des autonomies des chariots repas, des bornes inférieure et supérieure du nombre de manutentionnaires, des bornes inférieure et supérieure du nombre de chauffeurs, du nombre de demandes en retard et du nombre de demandes en chariots repas pour lesquelles l'autonomie d'au moins un chariot a été dépassée. Les tableaux 4.10 et 4.11 présentent ces résultats.

Les résultats des tableaux 4.10 et 4.11 confirment la dominance des deux combinaisons ($Op_{mov} = 60, T_{lt} = 25$) et ($Op_{mov} = 80, T_{lt} = 25$). D'autre part, en doublant le nombre d'itérations sans amélioration, le temps de résolution est aussi doublé mais la valeur moyenne de la fonction objectif des solutions ne diminue qu'au plus de 7,5% quelque soit la combinaison. De plus, le nombre d'itérations sans amélioration influence très peu les bornes inférieures et supérieures du nombre des manutentionnaires et des chauffeurs.

4.4. RÉSULTATS EXPÉRIMENTAUX

Op_{mov}	60	60	60	80	80	80
T_{lt}	1	25	50	1	25	50
Fonction objectif	2785,8	2565,7	2728,4	2770,1	2490,1	2694,6
Somme des retards	2480,8	2272,7	2441,6	2465,9	2201,0	2410,9
Somme des dépassements des autonomies	79,06	68,56	62,30	77,80	64,57	58,59
Borne inférieure (manutentionnaires)	5,99	6,00	6,00	6,00	6,00	6,00
Borne supérieure (manutentionnaires)	6,00	6,00	6,00	6,00	6,00	6,00
Borne inférieure (chauffeurs)	15,53	15,46	15,41	15,59	15,44	15,52
Borne supérieure (chauffeurs)	17,68	17,43	17,48	17,68	17,47	17,51
Nombre de demandes en retard	94,79	90,18	89,52	95,87	88,53	89,39
Nombre de demandes avec autonomie dépassée	4,49	4	3,75	4,49	3,71	3,75

TAB. 4.10 – Résultats des six meilleures combinaisons avec $\#ite = 5000$

Op_{mov}	60	60	60	80	80	80
T_{lt}	1	25	50	1	25	50
Fonction objectif	2534,9	2295,56	2553,3	2550,3	2273,1	2475,6
Somme des retards	2233,3	2014,2	2274,2	2250,1	1990,6	2196,6
Somme des dépassements des autonomies	75,35	56,14	54,24	74,96	58,99	54,77
Borne inférieure (manutentionnaires)	6,00	6,00	5,99	6,00	6,00	6,00
Borne supérieure (manutentionnaires)	6,00	6,00	6,00	6,00	6,00	6,00
Borne inférieure (chauffeurs)	15,73	15,51	15,52	15,51	15,33	15,48
Borne supérieure (chauffeurs)	17,52	17,53	17,46	17,54	17,37	17,36
Nombre de demandes en retard	90,83	85,48	86,07	92,03	84,21	85,97
Nombre de demandes avec autonomie dépassée	4,15	3,42	3,35	4,17	3,40	3,51

TAB. 4.11 – Résultats des six meilleures combinaisons avec $\#ite = 10000$

4.4. RÉSULTATS EXPÉRIMENTAUX

Sélection	M	R	T	M	R	T	M	R	T
T_{lt}	1	1	1	10	10	10	25	25	25
$\Delta_i(\%)$	27,0	48,4	34,6	25,1	40,5	31,1	20,3	41,2	30,9
$\sigma(\Delta_i)(\%)$	20,5	17,4	21,0	21,1	20,0	19,7	21,0	17,6	19,8
Nb_{min}	12	1	9	24	4	8	33	0	9
$Tps(s)$	3732	4244	3795	4067	5025	4489	3826	4333	3838
$Nb_{\Delta \leq 10}$	29	1	15	35	7	17	42	5	18
$Nb_{\Delta \leq 20}$	41	3	27	45	18	30	57	13	32

TAB. 4.12 – Résultats pour l’algorithme mémétique avec une population de 100 individus

Sélection	M	R	T	M	R	T	M	R	T
T_{lt}	1	1	1	10	10	10	25	25	25
$\Delta_i(\%)$	27,3	48,2	33,5	25,4	45,0	33,8	22,3	43,7	29,5
$\sigma(\Delta_i)(\%)$	20,0	19,6	19,2	20,8	16,4	21,1	20,1	19,0	20,7
Nb_{min}	18	4	7	20	1	8	25	3	15
$Tps(s)$	9418	10835	9151	10979	11535	9188	12540	14039	11920
$Nb_{\Delta \leq 10}$	27	5	11	29	3	15	38	5	23
$Nb_{\Delta \leq 20}$	37	9	24	45	8	28	52	12	36

TAB. 4.13 – Résultats pour l’algorithme mémétique avec une population de 200 individus

4.4.4 Algorithme mémétique

Suite à des tests préliminaires et aux résultats de l’algorithme génétique et de l’algorithme tabou, nous avons fixé le nombre d’itérations sans amélioration de l’algorithme mémétique à 100. La taille de la population est composée de 100 individus puis 200 individus. L’opérateur de croisement est C et les trois opérateurs de sélection ont été testés. Les probabilités de mutation sont identiques à celles de l’algorithme génétique. L’opérateur de voisinage de type déplacement d’une demande pour l’algorithme tabou a été fixé à 60 et les trois valeurs de la taille de liste tabou testées sont 1, 10 et 25. Les tableaux 4.12 et 4.13 présentent les mêmes indicateurs que pour les tests des algorithmes précédents : la moyenne Δ_i des écarts relatifs entre la meilleure solution trouvée pour toutes les combinaisons et la meilleure solution trouvée par la configuration i , l’écart type $\sigma(\Delta_i)$, le nombre d’instances pour lesquelles la combinaison i trouve la meilleure solution (Nb_{min}) et le temps moyen de résolution en secondes (Tps). Pour une comparaison plus fine, nous avons également ajouté le nombre d’instances pour lesquelles l’écart relatif est inférieur à 10% ($Nb_{\Delta \leq 10}$) et 20% ($Nb_{\Delta \leq 20}$).

Les résultats montrent globalement une dominance de l’opérateur de sélection M quelque soit la taille de la liste tabou. De plus, indépendamment de l’opérateur de sélection, une taille de la liste tabou égale à 25 paraît le meilleur compromis. La combinaison (M, 25) est donc la plus performante. Mais l’écart entre cette combinaison et toutes les autres combinaisons diminue légèrement lorsque le nombre d’individus dans la population augmente.

4.4. RÉSULTATS EXPÉRIMENTAUX

Ensuite, comme pour les autres algorithmes, nous avons relevé sur les 6 meilleures combinaisons (en enlevant les combinaisons avec l'opérateur de sélection R) la moyenne sur 100 instances de la fonction objectif, de la somme des retards en minutes, de la somme des dépassements des autonomies des chariots repas, des bornes inférieure et supérieure du nombre de manutentionnaires, des bornes inférieure et supérieure du nombre de chauffeurs, du nombre de demandes en retard et du nombre de demandes en chariot repas pour lesquelles l'autonomie d'au moins un chariot a été dépassée. Les tableaux 4.14 et 4.15 présentent ces résultats.

Sélection	M	T	M	T	M	T
T_{tt}	1	1	10	10	25	25
Fonction objectif	2995,5	3316,0	2892,9	3151,6	2739,0	3086,7
Somme des retards	2750,6	3073,4	2649,5	2906,7	2499,6	2844,9
Somme des dépassements des autonomies	22,22	19,58	21,43	21,84	16,93	18,18
Borne inférieure (manutentionnaires)	6,00	5,98	6,00	6,00	5,99	6,00
Borne supérieure (manutentionnaires)	6,00	6,00	6,00	6,00	6,00	6,00
Borne inférieure (chauffeurs)	15,13	15,11	15,09	15,15	15,15	15,25
Borne supérieure (chauffeurs)	17,41	17,52	17,32	17,47	17,37	17,47
Nombre de demandes en retard	63,29	67,75	63,00	65,42	60,49	65,41
Nombre de demandes avec autonomie dépassée	1,81	1,75	1,84	1,74	1,61	1,62

TAB. 4.14 – Résultats des six meilleures combinaisons avec une population de 100 individus

Les tableaux 4.14 et 4.15 montrent aussi la dominance de la combinaison (M, 25) néanmoins l'écart avec toutes les autres combinaisons se réduit lorsque le nombre d'individus dans la population augmente. Dans le cas d'une population de 200 individus, le nombre moyen de demandes en retard et les bornes inférieures et supérieures du nombre de personnes sont sensiblement les mêmes sur toutes les combinaisons. Les différentes combinaisons se distinguent donc entre elles sur principalement le critère de la somme des retards. Enfin, l'augmentation de la population permet d'améliorer nettement la qualité des solutions trouvées par les différentes combinaisons, au détriment du temps de résolution qui est parfois triplé.

4.4.5 Comparaison des trois algorithmes

Nous avons ensuite comparé les meilleures combinaisons identifiées précédemment, relativement aux trois algorithmes. Pour l'algorithme génétique, les deux combinaisons (C,

4.4. RÉSULTATS EXPÉRIMENTAUX

Sélection	M	T	M	T	M	T
T_{lt}	1	1	10	10	25	25
Fonction objectif	2471,0	2664,7	2403,9	2697,5	2274,5	2529,1
Somme des retards	2229,6	2421,4	2163,3	2456,8	2033,6	2291,6
Somme des dépassements des autonomies	19,85	21,6	18,95	18,31	19,05	16,69
Borne inférieure (manutentionnaires)	5,99	6,00	6,00	6,00	6,00	6,00
Borne supérieure (manutentionnaires)	6,00	6,00	6,00	6,00	6,00	6,00
Borne inférieure (chauffeurs)	15,09	15,05	15,08	15,16	15,09	14,92
Borne supérieure (chauffeurs)	17,23	17,29	17,25	17,31	17,28	17,24
Nombre de demandes en retard	56,01	55,88	56,43	56,69	54	55,06
Nombre de demandes avec autonomie dépassée	1,6	1,7	1,77	1,55	1,73	1,43

TAB. 4.15 – Résultats des six meilleures combinaisons avec une population de 200 individus

M) et (C, T) pour un nombre d'itérations sans amélioration égal à 200, ont été sélectionnées. Pour l'algorithme tabou, les combinaisons ($Op_{mov} = 60$, $T_{lt} = 25$) et ($Op_{mov} = 80$, $T_{lt} = 25$) ont été choisies avec un nombre d'itérations sans amélioration égal à 10 000. Enfin, pour l'algorithme mémétique, l'opérateur de sélection est M , le nombre d'individus dans la population est égal à 200 et les deux tailles de la liste tabou testées sont 10 et 25. Le tableau 4.16 présente la comparaison de ces algorithmes en conservant pour chacun deux combinaisons. Pour les 100 instances, nous déterminons la moyenne Δ_i des écarts relatifs entre la meilleure solution trouvée pour toutes les combinaisons des algorithmes et la meilleure solution trouvée par la configuration i , l'écart type $\sigma(\Delta_i)$, le nombre d'instances pour lesquelles la combinaison i trouve la meilleure solution (Nb_{min}) et le temps moyen de résolution en secondes (Tps). Pour une comparaison plus fine, nous avons également ajouté le nombre d'instances pour lesquelles l'écart relatif est inférieur à 10% ($Nb_{\Delta \leq 10}$) et 20% ($Nb_{\Delta \leq 20}$).

Parmi ces algorithmes, la première combinaison de l'algorithme génétique et la dernière combinaison de l'algorithme mémétique semblent dominer les autres combinaisons et la recherche tabou. Cependant, l'algorithme tabou est la méthode la plus rapide et obtient des solutions dégradées d'au plus 4% en écart relatif moyen. L'algorithme mémétique est similaire à l'algorithme génétique aux vues des résultats de ce tableau 4.16. A priori, le temps passé par l'algorithme mémétique à améliorer les individus de la population par la méthode tabou au détriment d'un nombre moins important de générations de nouvelles populations ne permet pas d'obtenir des résultats significativement meilleurs que l'algorithme génétique. De plus, le temps de résolution de l'algorithme mémétique est au minimum doublé

4.4. RÉSULTATS EXPÉRIMENTAUX

Algo.	génétique	génétique	tabou	tabou	mémétique	mémétique
Combi.	(C, M)	(C, T)	$Op_{mov} = 60$ $T_{lt} = 25$	$Op_{mov} = 80$ $T_{lt} = 25$	$T_{lt} = 10$	$T_{lt} = 25$
$\Delta_i(\%)$	22,6	27,7	26,6	26,2	26,9	23,7
$\sigma(\Delta_i)(\%)$	16,4	18,0	14,7	14,1	18,7	18,4
Nb_{min}	19	6	7	7	19	20
$Tps(s)$	4881,26	5140,86	1441,36	1665,63	10979,82	12540,20
$Nb_{\Delta \leq 10}$	27	17	14	16	22	29
$Nb_{\Delta \leq 20}$	45	40	32	29	32	43

TAB. 4.16 – Comparaison des trois algorithmes

par rapport à l'algorithme génétique.

Enfin, les derniers résultats expérimentaux comparent la meilleure combinaison de l'algorithme mémétique avec une nouvelle hybridation assez simple entre l'algorithme génétique et la recherche tabou. Cette hybridation consiste à exécuter dans un premier temps l'algorithme génétique puis à appliquer la recherche tabou dans un deuxième temps à la meilleure solution trouvée par le premier algorithme. Les paramètres de l'hybridation correspondent aux paramètres de la meilleure combinaison de l'algorithme génétique, colonne 1 du tableau 4.16, et de la recherche tabou précisé par la colonne 4 du même tableau. Les résultats obtenus sont présentés dans le tableau 4.17 de la même manière que les tableaux précédents.

Le tableau 4.17 montre que l'hybridation à deux de temps de l'algorithme génétique puis de la recherche tabou est plus performant que l'algorithme mémétique. La valeur de l'écart relatif moyen est diminuée de 10,3% grâce à cette nouvelle hybridation. Pour seulement un tiers de ces 100 instances, l'algorithme mémétique trouve une meilleure solution que cette hybridation avec un temps de résolution légèrement supérieur. Sur l'ensemble des critères de la fonction objectif, l'hybridation est principalement plus performante sur la somme des retards des demandes. Cependant, il serait intéressant d'augmenter encore le nombre d'itérations sans amélioration de l'algorithme mémétique pour étudier sa convergence vers de meilleures solutions. Soit l'algorithme n'a pas assez de temps pour continuer à améliorer la solution soit l'algorithme, dans l'état actuel, ne permet pas de trouver de meilleures solutions que l'hybridation. Dans ce cas, une étude sur la diversité de la population tout au long de la résolution pourrait montrer une convergence trop prématurée de la méthode, essentiellement causée par l'amélioration des individus par la recherche tabou. Dans ce cas, des techniques de gestion de diversité de la population, comme par exemple ceux menés dans les travaux de Prins et al. [161] et Sörensen et Sevaux [181], permettraient d'améliorer l'algorithme mémétique.

4.4. RÉSULTATS EXPÉRIMENTAUX

Algorithme	mémétique	génétique puis tabou
$\Delta_i(\%)$	16,2	5,9
$\sigma(\Delta_i)(\%)$	16,1	11,2
Nb_{min}	33	67
$Tps(s)$	12540,20	10479,63
Fonction objectif	2274,50	1949,73
Somme des retards	2033,60	1701,02
Somme des dépassements des autonomies	19,05	27,66
Borne inférieure (manutentionnaires)	6,00	5,99
Borne supérieure (manutentionnaires)	6,00	6,00
Borne inférieure (chauffeurs)	15,09	14,93
Borne supérieure (chauffeurs)	17,28	17,29
Nombre de demandes en retard	54,00	60,51
Nombre de demandes avec autonomie dépassée	1,73	1,99

TAB. 4.17 – Comparaison entre l'algorithme mémétique et une hybridation à deux temps

4.5 Conclusion

Dans ce chapitre, nous nous sommes intéressés au problème d'optimisation des tournées des véhicules de la logistique du CHRU en considérant des hypothèses émises par le service de la logistique du CHRU et d'autres contraintes rendant le problème plus appliqué au contexte de l'étude. Les algorithmes présentés au chapitre précédent ont été modifiés de manière à prendre en compte ces nouvelles contraintes et hypothèses et à améliorer leurs performances. Un nouvel algorithme est également proposé, un algorithme mémétique, fruit d'une hybridation entre un algorithme génétique et une recherche tabou. Pour tester ces méthodes de résolutions, des nouvelles instances ont été générées en fonction de données réelles fournies par l'hôpital. Les résultats expérimentaux des trois algorithmes montrent que l'algorithme génétique est plus performant que la recherche tabou et domine légèrement l'algorithme mémétique. Au niveau du ratio de la qualité des solutions et du temps de résolution, la recherche tabou paraît être la meilleure méthode. De plus, des tests sur une hybridation à deux temps (algorithme génétique puis recherche tabou) montrent une efficacité supérieure à l'algorithme mémétique pour trouver de bonnes solutions.

D'autres améliorations peuvent être apportées aux algorithmes présentés dans ce chapitre. Des techniques de gestion de diversité de la population permettraient d'améliorer l'algorithme mémétique mais aussi l'algorithme génétique. Ces techniques consistent à s'assurer que les algorithmes ne convergent pas trop vite vers une unique solution ou "super-individu". Pour cela, il existe plusieurs méthodes : ajouter des nouveaux individus, refuser des individus enfants trop similaires aux individus déjà présents dans la population courante ou encore augmenter l'action de l'opérateur de mutation. D'autres choix des paramètres d'hybridation de l'algorithme mémétique peuvent être testés comme améliorer seulement les mauvais individus de la population par la recherche tabou ou encore changer le nombre d'itérations de cet algorithme en fonction de la qualité des individus. Pour la recherche tabou, des techniques d'intensification ou de diversification de l'espace de voisinage d'une solution permettraient de l'améliorer. Une phase d'intensification consiste à augmenter le nombre de voisins d'une solution lorsque la qualité de la solution courante est meilleure que la qualité moyenne des solutions parcourues. Et dans le cas contraire, les ensembles de voisinages des solutions courantes de moins bonnes qualités sont diminués. La phase de diversification permet de diriger la recherche vers des ensembles de voisinages pas ou peu visités. Elle consiste par exemple à appliquer plusieurs fois aléatoirement un opérateur de voisinage lorsqu'on estime que le voisinage courant ne contient plus beaucoup de nouvelles solutions intéressantes.

Ces méthodes de résolution permettent d'obtenir des solutions d'organisation des tournées de livraison et de répartition du nombre de chauffeurs et manutentionnaires. Cependant, le défaut de ces solutions est qu'elles sont basées sur des instances dont les données sont fixes. Dans le réel, les quantités des chariots de chaque demande sont souvent variables et les durées des trajets et temps de livraison ne sont pas constants. Il serait donc intéressant d'évaluer la robustesse de ces solutions dans un environnement dynamique avec des incertitudes. L'une des techniques adaptées à cette problématique est la simulation, objet du chapitre suivant.