



HAL
open science

Optimisation de la préparation de commandes dans les entrepôts de distribution

Nicolas Lenoble

► **To cite this version:**

Nicolas Lenoble. Optimisation de la préparation de commandes dans les entrepôts de distribution. Gestion et management. Université Grenoble Alpes, 2017. Français. NNT : 2017GREAI052 . tel-01709845

HAL Id: tel-01709845

<https://theses.hal.science/tel-01709845>

Submitted on 15 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Génie Industriel**

Arrêté ministériel : 7 août 2006

Présentée par

Nicolas LENOBLE

Thèse dirigée par **Yannick FREIN** et
Codirigée par **Ramzi HAMMAMI**

préparée au sein du **Laboratoire G-SCOP**
dans l'**École Doctorale I-MEP²**

Optimisation de la préparation de commandes dans les entrepôts de distribution

Thèse soutenue publiquement le 23 Octobre 2017,
devant le jury composé de :

Mme. Atidel HADJ-ALOUANE

Professeur à l'Ecole Nationale d'Ingénieur de Tunis, Président

Mme. Olga BATAIA

Professeur à l'Institut Supérieur de l'Aéronautique et de l'Espace de
Toulouse, Rapporteur

Mr. Uche OKONGWU

Professeur à Toulouse business school, Rapporteur

Mme. Gülgün ALPAN-GAUJAL

Maître de conférences, HDR, Grenoble INP, Examinatrice

Mr. Yannick FREIN

Professeur à Grenoble INP, Directeur de thèse

Mr. Ramzi HAMMAMI

Professeur à Rennes School of Business, Co-directeur de thèse

Mr. Djamal SIMOHAND

Responsable R&D société KLS à Meylan, Invité



Remerciements

Cette thèse touche à sa fin, et c'est avec plaisir que je souhaite remercier de nombreuses personnes.

Tout d'abord, je remercie chaleureusement toutes les personnes qui m'ont aidé pendant l'élaboration de ma thèse et notamment mes directeurs de thèse Yannick Frein et Ramzi Hammami, pour l'intérêt qu'ils ont porté à mon travail et le soutien qu'ils m'ont apporté, ainsi que leur grande disponibilité et les nombreux conseils qu'ils m'ont donné durant la rédaction de ma thèse. Je tiens aussi à les remercier pour leur patience et leur compréhension.

Ce travail n'aurait pas été possible sans le soutien du laboratoire G-SCOP ainsi que l'entreprise KLS Logistic, qui m'ont permis de me consacrer sereinement à l'élaboration de ma thèse durant ces trois années.

Je remercie également l'ensemble du personnel de KLS. Tout d'abord j'exprime ma gratitude à Gilbert Garcia, qui m'a permis de venir dans cette entreprise pour effectuer ma thèse dans les meilleures conditions. Il m'a aussi permis de m'épanouir au sein de la société en me donnant de précieux conseils et en faisant part d'une grande implication pour mon travail. Mes remerciements vont aussi à Djamel Simohand, pour son temps et ses précieuses idées qui m'ont permis de concrétiser cette thèse.

Je remercie également le personnel de l'entreprise KLS pour m'avoir permis de passer trois ans dans les meilleures conditions et dans un climat sympathique, à mes collègues de bureau Shadi, Aurélie et Gilles. Ils m'ont apporté support et conseils durant ces trois années, sans compter une bonne ambiance quotidienne permettant de réaliser cette thèse dans les meilleures conditions. A Maria, Sandrine, Christelle et Cathy, pour leur sympathie et leur aide toujours indispensable ! Je n'oublie pas, Grégoire, Thomas, Didier, Adrien, Benoit, Julien, Gilles et tous les autres pour les diverses activités réalisées au cours de ses trois ans, de leurs précieux conseils au cours de la thèse.

Je remercie tous les membres du laboratoire G-SCOP. Les nombreuses discussions que j'ai pu avoir avec chacun m'ont beaucoup apporté, autant sur le plan personnel que professionnel. Merci aussi pour le temps qu'ils ont passé avec moi, leur compétence, leur gentillesse et leurs conseils. Un grand merci à

Marie Josèphe Perruet pour sa sympathie, sa disponibilité ainsi que son sourire toutes les fois que je venais au laboratoire.

Mes remerciements vont aussi à mes amis. Ainsi je remercie Aurélien, Hugo, Toulouse et Lévy, pour les weekends à Paris me permettant de me changer les idées, pour le soutien qu'ils m'ont apporté. Leurs attentions et encouragements m'ont accompagné tout au long de ces années. Je remercie aussi Pierre, pour m'avoir fait visiter Clermont Ferrand et pour nos vacances en Italie en compagnie de Mejdì, Matthieu et Aurélien. Je remercie grandement Chopi et Benoit, Eliot et les Jérèms (le grand et le petit), Sophie et Anaïs, ainsi que mon Jojo, pour les weekends à Montpellier me permettant de décompresser. Ils n'ont eu de cesse de m'épauler et de m'aider durant ces trois années, afin que je puisse finaliser cette thèse. Grand MERCI à eux ! Je remercie également mes amis de l'étranger, des Pays Bas et de Luxembourg, à savoir Stathis et Pablo pour les discussions que l'on a eues !! Aussi je tiens à remercier Nadine Gruchot, qui a été là pour moi, quand cela n'allait pas, quand je suis venu la voir à Berlin. Elle est cependant partie trop tôt mais je tiens à lui dire merci pour tout ce qu'elle a fait pour moi.

Mes remerciements vont également à mes amis de Grenoble, à savoir mon coloc à l'heure où j'écris ce mémoire, Alex, qui m'a fait passer de superbes soirées à Grenoble. Il m'a épaulé pendant ces trois ans, et je le remercie d'avoir été là durant les périodes de doutes. Je remercie également William, Amélie et Aurélien pour leurs chaleureux encouragements. Pour les repas, soirées, jeux de sociétés et mölky que l'on a fait !

Je remercie ma famille, à commencer par mes parents, toujours là pour me réconforter quand je n'allais pas bien en période d'écriture, et durant ces trois années, pour leur accompagnement et conseils toujours judicieux. Je remercie également grandement ma grand-mère, pour toute sa tendresse et sa disponibilité ! Je remercie ma sœur et tout le reste de ma famille, pour leurs soutiens moral et matériel et leur confiance indéfectible dans mes choix.

Mes remerciements vont finalement à Berengère, pour toute l'affection qu'elle me porte, pour sa présence et son authenticité, pour sa gentillesse apporté lors de moments difficiles. Je la remercie pour son soutien quotidien et son enthousiasme à l'égard de mes travaux comme de la vie en général.

Au terme de ce parcours, je remercie enfin celles et ceux qui me sont chers et que j'ai quelque peu délaissé ces derniers mois pour achever cette thèse.

Pour finir, mes remerciements ne vont pas à youtube, à Blizzard, ni à Mark Zuckerberg, qui m'ont fait perdre beaucoup de temps.

Résumé

La préparation de commandes est une activité primordiale dans les entrepôts de distribution (près de 60% des coûts opérationnels dans les entrepôts traditionnels). Un des moyens de réduire ces coûts est de collecter plusieurs commandes simultanément, plutôt qu'une par une ; cela permet d'éviter des déplacements inutiles des opérateurs dans le cas de collecte manuelle des produits, et de moins solliciter les machines, dans le cas de collecte automatisée. Nous considérons dans cette thèse des entrepôts de distribution où les produits sont stockés dans des machines appelées stockeurs automatisés composés de plateaux sur lesquels les produits sont stockés ; nous prenons en compte deux types de stockeurs : les VLMs et Carousels. Ces deux types de machines, en plein développement, se différencient par le temps nécessaire pour passer d'un plateau au plateau suivant à visiter (constant dans le cas d'un VLM, dépendant du nombre d'étages entre les 2 plateaux visités dans le cadre des carousels). L'objectif de la thèse est donc de développer des méthodes pour faire des regroupements de commandes en lots dans des entrepôts automatisés afin de collecter un ensemble donné de commandes le plus efficacement possible selon des critères que nous allons définir.

Nous étudions tout d'abord un premier type de regroupement de commandes en lots, pour lequel chaque lot sera collecté entièrement (toutes les commandes composant le lot seront traitées) avant de passer au suivant. Nous parlerons simplement de batching. Nous évaluons le temps de préparation de commandes, dans le cas où l'opérateur effectue la collecte sur une ou sur plusieurs machines. L'intérêt d'utiliser plusieurs machines est de permettre la recherche de plateaux en temps masqué (le stockeur effectue un changement de plateaux pendant que l'opérateur est occupé à effectuer la collecte sur d'autres stockeurs). Cette évaluation du temps de préparation de commandes nous permet d'extraire un critère d'optimisation et d'établir des modèles d'optimisation exacts pour les VLMs puis les carousels. Ces modèles sont ensuite testés avec des données réelles d'entreprise, grâce au partenariat avec l'entreprise KLS Logistic, éditeur du WMS Gildas. Enfin, nous étendons la résolution aux méthodes approchées de type métaheuristiques, afin de garantir de bonnes solutions sous un temps de calcul raisonnable. Des résultats significatifs en termes de réduction du temps de préparation de commandes permettent de justifier l'intérêt de nos travaux.

Nous étudions ensuite un deuxième type de regroupement, où un ensemble de commandes est collecté simultanément mais, contrairement au cas précédent, à chaque fois qu'une commande est terminée, elle est immédiatement remplacée. Nous parlerons ici de « Rolling batching ». Cette approche est classiquement utilisée dans les « Drive ». Nous nous focalisons sur le cas de collecte effectuée sur des carousels, système le plus utilisé dans les « Drive ». Nous développons un algorithme permettant le calcul du temps d'attente de l'opérateur. Une résolution approchée couplant l'utilisation d'heuristiques et d'une métaheuristique est proposée afin de résoudre efficacement le séquençement des commandes. Nous notons que des gains significatifs sont obtenus par l'utilisation de la méthode proposée.

Mots-clés

Préparation de commandes, Optimisation, Métaheuristiques, Carousels, VLMs, Regroupement de commandes en lots.

Table des matières

Remerciements	i
Résumé	iii
Mots-clés	iv
Liste des Figures	vii
Liste des Tableaux	ix
Chapitre 1 Introduction	1
Chapitre 2 Revue de littérature	5
2.1 Positionnement de notre problème.....	5
2.1.1 Les entrepôts	5
2.1.2 Le degré d'automatisation.....	7
2.2 Les activités impactant la préparation de commandes.....	10
2.2.1 L'agencement et le zonage des entrepôts	11
2.2.2 L'affectation de produits aux emplacements	14
2.2.3 Le routing.....	17
2.2.4 Le regroupement de commandes en lots complets (Batching) et séquençement des commandes ...	20
2.3 Préparation de commandes et stockeurs automatisés	27
2.3.1 Automated Storage and Retrieval Systems (AS/RS).....	27
2.3.2 Autonomous Vehicle Storage and Retrieval System (AVS/RS).....	28
2.3.3 Shuttle Based Storage and Retrieval Systems (SBS/RS)	30
2.3.4 Le Vertical Lift Module (VLM)	31
2.3.5 Le Carousel	33
2.4 Résumé de la littérature et intérêt de notre travail	35
Chapitre 3 Regroupement de commandes en lots dans une zone de préparation équipée de VLMs	37
3.1 Procédé de préparation de commandes sur des stockeurs automatisés de type VLM avec regroupement en lots	37
3.1.1 Spécificité du Vertical Lift Module	38
3.1.2 Procédé de picking considérant un regroupement de commandes sur un ou plusieurs VLMs	39
3.2 Regroupement de commandes en lots dans un système de préparation de commandes utilisant 1 VLM	42

3.2.1	Notation et hypothèses	42
3.2.2	Evaluation du temps de préparation de commandes sur 1 VLM	44
3.2.3	Modèle d'optimisation et étude de cas	47
3.3	Regroupement de commandes dans un système de préparation de commandes utilisant plusieurs VLMs ..	56
3.3.1	Notations et hypothèses utilisées	56
3.3.2	Evaluation du temps de préparation de commandes sur plusieurs VLMs	57
3.3.3	Modèle d'optimisation et étude de cas	63
3.4	Résolution par métaheuristique	71
3.4.1	Structure de la solution et paramétrage des métaheuristiques testées	72
3.4.2	Test et comparaison des performances des métaheuristiques testées	75
Chapitre 4	Regroupement de commandes en lots dans une zone de préparation équipée de Carousels	79
4.1	Préparation de commandes avec des carousels	79
4.1.1	Spécificité du Carousel	79
4.1.2	Procédé de picking considérant un regroupement de commandes sur un ou plusieurs Carousels	81
4.2	Regroupement de commandes dans un système de préparation de commandes utilisant 1 Carousel	83
4.2.1	Notations et hypothèses	83
4.2.2	Evaluation du temps de préparation de commandes sur 1 Carousel	86
4.2.3	Modèle d'optimisation et étude de cas	87
4.3	Regroupement de commandes dans un système de préparation de commandes utilisant plusieurs Carousels	93
4.3.1	Notations et hypothèses utilisées	93
4.3.2	Evaluation du temps de préparation de commandes sur plusieurs Carousels	94
4.3.3	Modèle d'optimisation et étude de cas	98
4.4	Résolution métaheuristique	107
Chapitre 5	Séquencement des commandes dans un système Drive	109
5.1	Caractéristiques du système « Drive »	109
5.2	Définition de l'algorithme permettant le calcul du temps d'attente dans un système « Drive »	112
5.2.1	Hypothèses prises en comptes pour le calcul du temps d'attente d'un système « Drive »	112
5.2.2	Présentation de l'algorithme utilisé	113
5.3	Résolution approchée	116
5.3.1	Définitions d'heuristiques constructives	116
5.3.2	Définition de la Métaheuristique	120
5.4	Etudes de cas	122
Chapitre 6	Conclusion et perspectives de recherche	127
Bibliographie	131
Annexe	141

Liste des Figures

Figure 1 : Présentation d'un entrepôt type avec ses flux associés (tiré de http://www.cosmos-consultants.eu/h-log-partenaire).....	6
Figure 2 : Classification des systèmes de préparation de commandes (repris de De Koster, 2004)	8
Figure 3 : Illustration de la complexité du design d'un système de préparation de commandes (repris de Goetschalckx and Ashayeri 1989)	10
Figure 4 : Principales décisions devant être prises lors du design d'entrepôt (repris de De Koster et al., 2007)	12
Figure 5 : Le flying V et fishbone design présentés par Grue et Meller 2009	13
Figure 6 : Arbre de décision des zones selon Frazelle (2002)	14
Figure 7 : Méthode ABC inter et Intra allée De Koster et al, (2007)	16
Figure 8 : Illustration des emplacements d'une commande à collecter et sa représentation graphique.	18
Figure 9 : Différentes heuristiques et méthode optimale présentées par Roodbergen en 2001	18
Figure 10 : Modèle d'optimisation pour le regroupement de commandes en lots (proposé par Gademann & Van De Velde, 2005)	22
Figure 11 : Illustration d'un AS/RS basique	28
Figure 12 : Illustration d'un AVS/RS	29
Figure 13 : Illustration de SBS/RS	31
Figure 14 : Illustration VLM	32
Figure 15 : Illustration Carousel	33
Figure 16 : Succession des décisions à prendre en compte afin d'aborder la préparation de commandes (inspiré de Shouman et al (2007))	35
Figure 17 : Illustration d'un VLM (vue de côté).....	39
Figure 18 : Préparation de commandes avec regroupement en lots et utilisation de VLMs40	
Figure 19 : Procédure de préparation de commandes regroupées en lots sur V VLMs	41
Figure 20 : Illustration du temps de préparation de commandes lorsque le temps de collecte est considéré négligeable.....	45
Figure 21 : Illustration du temps de préparation de commandes lorsque le temps de picking sur un plateau est considéré proportionnel au nombre de lignes à collecter sur ce même plateau	46
Figure 22 : Gains en temps de préparation pour les trois scénarii étudiés.....	51
Figure 23 : Gains en temps de préparation pour les trois scénarii étudiés.....	52

Figure 24 : Illustration du temps de préparation de commandes sur 3 VLMs lorsque le temps de collecte est considéré négligeable	59
Figure 25 : Illustration du temps de préparation de commandes sur 3 VLMs lorsque le temps de collecte par plateau est considéré proportionnel au nombre de lignes de commandes à collecter sur ce plateau ($p = 0,2 * e$)	61
Figure 26 : Gains en temps de préparation pour les trois scénarii étudiés	66
Figure 27 : Gains en temps de préparation pour les trois scénarii étudiés	68
Figure 28 : Illustration d'un carrousel vertical.....	80
Figure 29 : Procédure de préparation de commandes regroupées en lots sur Q carousels	82
Figure 30 : Illustration du temps de préparation de commandes lorsque le temps de picking sur un plateau est considéré proportionnel au nombre de lignes à collecter sur ce même plateau	87
Figure 31 : Distance minimale dans un cercle.....	88
Figure 32 : Représentation graphiques des Gains Distances et Gains C pour les instances de l'Hopital Francais 2.....	92
Figure 33 : Illustration du temps de préparation de commandes lorsque le temps de picking sur un plateau est considéré proportionnel au nombre de lignes à collecter sur ce même plateau ($r = p$)	95
Figure 34 : Récapitulatif des gains pour l'hôpital Français 2.....	102
Figure 35 : Récapitulatif des gains pour l'Entreprise Suisse.....	103
Figure 36 : Illustration d'un système "Drive" (source : www.kardex-remstar.be).....	110
Figure 37 : Méthodologie utilisée pour définir l'ordre d'ajout des commandes dans un système « Drive ».....	121
Figure 38 : Gains en temps d'attente.....	124

Liste des Tableaux

Tableau 1 : Les performances de notre modèle en termes de gain plateau pour l'Hôpital français.....	50
Tableau 2 : Les performances de notre modèle en termes de gain C_i pour l'Hôpital français	50
Tableau 3 : Les performances de notre modèle en termes de gain plateau pour l'entreprise Suisse	52
Tableau 4 : Les performances de notre modèle en termes de gain C_i pour l'entreprise Suisse	52
Tableau 5 : Effet du nombre de commandes sur les performances de notre modèle	54
Tableau 6 : Effet du nombre de lignes de commandes sur les performances de notre modèle	55
Tableau 7 : Les performances de notre modèle en termes de gain plateaux pour l'Hôpital français.....	66
Tableau 8 : Les performances de notre modèle en termes de gain C_i pour l'Hôpital français	66
Tableau 9 : Les performances de notre modèle en termes de gain plateaux pour l'entreprise Suisse	67
Tableau 10 : Les performances de notre modèle en termes de gain C_i pour l'entreprise Suisse	67
Tableau 11 : Effet du nombre de commandes sur les performances	69
Tableau 12 : Effet du nombre de lignes de commandes par commande sur les performances	70
Tableau 13 : Résultats de l'influence des deux paramètres : Nombre de commandes et nombre moyen de lignes de commandes.....	70
Tableau 14 : Codage de la Solution.....	72
Tableau 15 : Opérateur SWAP	73
Tableau 16 : Opérateur de croisement à un point.....	74
Tableau 17 : Récapitulatif des résultats des métaheuristiques testés (1 minute de calcul) 76	
Tableau 18 : Récapitulatif des résultats des métaheuristiques testés (5 minutes de calcul)77	
Tableau 19 : Récapitulatif des résultats des métaheuristiques	77
Tableau 20 : Les performances de notre modèle pour l'Hôpital français 2	91
Tableau 21 : Les performances de notre modèle en termes de gain Distance pour l'Hôpital français 2.....	101
Tableau 22 : Les performances de notre modèle en termes de gain C pour l'Hôpital français 2	101

Tableau 23 : Les performances de notre modèle en termes de gain distance pour l'entreprise Suisse	102
Tableau 24 : Les performances de notre modèle en termes de gain C pour l'entreprise Suisse	102
Tableau 25 : Temps d'attente, de picking ainsi que de préparation de commandes en fonction du temps unitaire de picking pour notre méthode appliquée à l'Hôpital Français 2	104
Tableau 26 : Temps d'attente, de picking ainsi que de préparation de commandes en fonction du temps unitaire de picking pour la stratégie actuelle appliquée à l'Hôpital Français 2 .	105
Tableau 27 : Gains en temps d'attente et en temps de préparation de commandes pour L'hôpital Français 2	105
Tableau 28 : Temps d'attente, de picking ainsi que de préparation de commandes en fonction du temps unitaire de picking pour notre méthode appliquée à l'Entreprise Suisse.....	105
Tableau 29 : Temps d'attente, de picking ainsi que de préparation de commandes en fonction du temps unitaire de picking pour la stratégie actuelle appliquée à l'Entreprise Suisse ..	106
Tableau 30 : Gains en temps d'attente et en temps de préparation de commandes pour l'Entreprise Suisse	106
Tableau 31 : Récapitulatif des résultats des métaheuristiques testées (1 minute de calcul)	108
Tableau 32 : Encodage de la solution	120
Tableau 33 : Temps d'attente sous les différentes heuristiques de base	122
Tableau 34 : Temps d'attente des groupes d'heuristiques	123
Tableau 35 : Temps d'attente avec utilisation de la métaheuristique.....	123
Tableau 36 : Gain en temps d'attente et temps de préparation de commandes.....	124
Tableau 37 : Récapitulatif des critères utilisés ainsi que des gains associés lors du regroupement de commandes en lots.....	128

Chapitre 1 Introduction

Les dernières décades ont été le théâtre de tensions économiques sans précédent pour les entreprises, mais ont aussi permis des avancées technologiques très importantes. Dans cet environnement évolutif, les entreprises ont dû adapter leur Supply Chain pour assurer leur place dans un contexte fortement concurrentiel. Cette évolution a particulièrement été ressentie dans les entrepôts de distribution, levier important de performance pour les entreprises, de toutes tailles et de tous secteurs. La gestion de ces entrepôts induit des enjeux de productivité et de traçabilité, conditionnant la satisfaction du client final. De ce fait, cette gestion doit être à la fois orientée niveau de service, profit ainsi que pilotée de manière efficiente. Le niveau de service est la principale priorité des entrepôts et la préparation de commandes étant l'interface entrepôt client, cette activité doit être privilégiée. L'orientation profit se fait par une poursuite de productivité dans toutes les activités présentes au sein de l'entrepôt et enfin le pilotage efficient se fait grâce au contrôle des divers indicateurs jugés adéquats. Le Warehouse Management System (WMS) est un des leviers essentiels permettant l'aboutissement de cette gestion effectuée de manière efficiente et automatisée dans le cadre des activités d'entreposage. Cette catégorie de logiciel vise à gérer les opérations d'un entrepôt dans le but de les optimiser. L'usage du WMS est d'autant plus nécessaire lorsque les entreprises choisissent de passer le cap de l'automatisation.

La hausse du foncier dans l'Europe fut une des causes de nombreux changements dans la façon de stocker les produits. A l'heure actuelle, au sein d'un entrepôt moderne, le produit est de plus en plus souvent stocké dans des machines automatisées. Ainsi l'opérateur ne se déplace plus dans les allées comme c'est le cas lorsque l'entrepôt est composé d'allées de racks, mais le produit est désormais amené jusqu'à lui. Ces machines automatisées sont appelées des stockeurs automatisés de par leur vocation à stocker les produits. L'automatisation des zones de stockage dans les entrepôts, mis à part un gain de place non négligeable, permet un accroissement de la productivité ainsi que sécuriser les produits stockés à l'intérieur. Ce gain en productivité se calcule en observant le nombre de commandes collectées par unité de temps.

Dans un contexte où les entreprises recherchent une productivité accrue, due aux tensions économiques, et se tournent vers de nouvelles technologies, l'optimisation de la préparation de commandes devient cruciale dans les entrepôts de distribution. La préparation de commandes consiste à prélever et rassembler les articles dans la quantité spécifiée par la commande. La préparation de commande (ou order picking) est l'activité représentant la plus grande charge de travail ainsi que la plus grande part des coûts opérationnels pour un entrepôt (Tompkins et al, 2010). De plus, elle est l'activité absorbant le plus de main d'œuvre au sein d'un entrepôt ainsi que celle consommant le plus de ressource financière. En effet Tompkins et al. (2010), Drury (1988) et Henn (2013) quantifient entre 55% à 60% des coûts opérationnels le seul coût lié à la préparation de commandes dans un entrepôt manuel tandis que De Koster et al (2007) évoquent aussi le fait que dans le cas des entrepôts automatisés, le coût de cette activité est important. Préparer le plus efficacement les commandes permet d'assurer un taux de service plus élevé, ainsi qu'une augmentation du nombre de commandes pouvant être préparées par jour. Le traitement de commandes dans les entreprises a déjà subi d'énormes changements.

Là ou auparavant les commandes étaient collectées une à une, actuellement certaines entreprises choisissent de collecter plusieurs commandes simultanément. La pratique s'est répandue par l'utilisation de stockeurs automatisés et de convoyeurs. Plusieurs commandes sont disposées sur une table de préparation et la collecte de l'ensemble de ces commandes est faite sur un ou plusieurs stockeurs automatisés. Les bacs qui doivent contenir les commandes sont mis à disposition en face du stockeur automatisé. Dans un entrepôt manuel (avec présence de rack) on se rend compte que si deux commandes sont composées des mêmes produits, il est bénéfique de les collecter ensemble si cela est possible. Il en est de même dans les entrepôts automatisés comme nous en discutons dans ce document. Une commande est composée de lignes de commandes, chacune d'entre elles correspond à une quantité donnée d'une référence donnée. Récolter efficacement un portefeuille de commandes vise à récolter l'ensemble des lignes de commandes dans un temps minimal. Nous nous intéressons donc dans cette thèse à la problématique du regroupement de commandes. Plus précisément, nous considérons deux contextes industriels importants. Dans les deux cas, les commandes sont regroupées en lots, un lot étant composé d'un ensemble de commandes, et une commande ne pouvant appartenir qu'à un seul lot.

Le premier contexte sera appelé « batching » ou « regroupement de commandes en lots complets ». Pour cette première situation, lorsqu'un lot de commandes est mis en préparation, celui-ci ne quitte la zone de ramassage qu'une fois l'ensemble des commandes du lot collectées.

Ainsi les bacs quittent la zone de préparation lorsque toutes les lignes de commandes associées aux commandes incluses dans le lot sont collectées. Dans ce cas, l'attribution de chaque commande à un lot donné devient l'enjeu permettant une préparation optimisée. Nous abordons ce problème de batching, au cours de notre étude, de manière « statique » : l'ensemble des lots de commandes est constitué avant que la préparation ne démarre, par exemple le matin pour l'ensemble des commandes à préparer dans la journée. Une autre façon aurait été de créer dynamiquement les lots de commandes : par exemple créer un nouveau lot dès qu'un lot est entièrement collecté. Cependant, ce contexte n'apparaissant pas chez les clients de notre partenaire industriel, nous avons écarté cette piste d'étude.

Le second contexte industriel étudié est le « Rolling batching », inspiré du « Rolling picking » décrit dans William and Fosso (1995). Tout comme pour le batching, l'opérateur collecte les commandes simultanément. Cependant à chaque fois que l'ensemble des lignes d'une commande donnée sont collectées, le bac correspondant est enlevé de la zone de préparation et un nouveau bac désignant une nouvelle commande est rajouté. Cette pratique est fortement utilisée dans le secteur du e-commerce. A chaque fois qu'une commande se termine, il faut donc choisir quelle commande rajouter parmi celles en attente. Ainsi une commande venant d'être reçue peut-être directement mise en préparation. Dans ce cas la définition d'un séquençement des commandes sur la table de préparation devient l'enjeu critique. Nous pouvons aborder la problématique du choix de la commande à rajouter de plusieurs façons. Tout d'abord en « statique » lorsque à l'instant initial, nous connaissons l'ensemble des commandes à séquencer pour une période donnée (par exemple une journée) et nous décidons de l'ensemble du séquençement. Une manière plus dynamique d'aborder le problème est, même lorsque nous connaissons l'ensemble des commandes à séquencer à l'état initial, de décider au fil de l'eau de la commande à rajouter, c'est-à-dire en décidant, à chaque fois qu'une commande est collectée, celle à rajouter. Enfin la manière totalement dynamique d'aborder le problème est de considérer la réception des commandes au fil de l'eau et de décider de la commande à rajouter lorsqu'une commande est collectée en fonction de l'ensemble des commandes présentes. Nos travaux se focalisent essentiellement sur le Rolling Batching statique.

Cette thèse s'intéresse donc à l'optimisation de la préparation de commandes dans les entrepôts, et à la définition d'une méthode de résolution adaptée au fonctionnement du WMS dans un environnement automatisé. Les types stockeurs automatisés, ayant chacun leurs spécificités, se multiplient au fil du temps. Nous portons notre attention sur deux types de stockeurs automatisés : les VLMs et les Carousels. Ces deux types de machine, en plein développement dans

la pratique, se différencient par le temps pour passer d'un plateau donné, sur lequel sont rangés les produits, au plateau suivant à visiter. Ces stockeurs automatisés sont de plus en plus présents dans les entrepôts modernes choisissant de passer le cap de l'automatisation. Ainsi nous rechercherons une méthodologie pour définir les temps de préparation de commandes, nous permettant d'identifier un critère d'optimisation pertinent pour une minimisation du temps de préparation de commandes. Ce critère sera ensuite utilisé dans les modèles d'optimisation que nous élaborons et résolvons au cours de notre thèse.

Notre thèse est ancrée dans la pratique grâce à une synergie avec le secteur industriel, en cohérence avec le contexte d'un contrat CIFRE réalisé en partenariat avec l'entreprise KLS Logistic, éditeur du WMS Gildas. Ainsi cette thèse a été effectuée en gardant à l'esprit le point de vue pratique et notamment l'impérieuse nécessité d'avoir des temps de calcul adaptés.

Le reste de ce manuscrit est divisé comme suit :

- Le chapitre 2 de ce manuscrit est consacré à l'étude de la littérature concernant la préparation de commandes ainsi que les différents stockeurs automatisés présents dans les entrepôts.

Dans les chapitres 3 et 4, nous considérons le contexte dit de « batching », ou bien « de regroupement de commandes en lots complets ».

- Le chapitre 3 se focalise sur la méthodologie afin d'optimiser le regroupement de commandes en lots avec l'utilisation de stockeurs automatisés de type VLMs
- Le chapitre 4 est dédié à la méthodologie permettant l'optimisation du regroupement de commandes en lots avec l'utilisation de stockeurs automatisés de type Carousels.
- Le chapitre 5 de ce manuscrit se consacre au séquençement des commandes dans un système de préparation de commandes automatisé de type « drive », pour lequel on considère un contexte industriel de type « rolling batching ».
- Enfin ce manuscrit est conclu par une synthèse de l'apport de nos travaux et la proposition de perspectives de recherche.

Chapitre 2 Revue de littérature

Ce chapitre a pour objectif de faire une revue de littérature concernant l'activité de préparation de commandes.

Dans une première partie, nous présenterons l'environnement global dans lequel se positionne notre étude. En seconde partie, nous verrons la complexité de l'activité de préparation de commandes et les dimensions impactant cette activité. Nous nous concentrerons sur la dimension de regroupement de commandes en lot et de séquençement des commandes, décisions primordiales lors d'une préparation de commandes, et dimensions au cœur de la présente thèse. Ensuite nous détaillerons la littérature décrivant la préparation de commandes utilisant des stockeurs automatisés et conclurons ce chapitre par un résumé de la littérature ainsi que les motivations nous ayant amené à traiter ce sujet, montrant l'intérêt de notre travail.

2.1 Positionnement de notre problème

2.1.1 Les entrepôts

D'après Lambert et al. (1998a) plus de 750 000 entrepôts étaient présents à travers le monde à la fin des années 1990. Ces entrepôts deviennent de plus en plus présents mondialement, permettant un rapprochement des clients à servir mais sont aussi de plus en plus avancés technologiquement avec l'apparition de nouveaux systèmes de stockage (Automated Storage Retrieval Systems Production Section of the Material Handling Industry of America, 2005). Bien que ces entrepôts demandent de larges investissements impliquant de larges coûts opérationnels (coût du terrain, des équipements ainsi qu'une grande masse salariale), de nombreux intérêts justifient l'utilisation des entrepôts pour le stockage des références. Nous énumérons quelques-uns de ses intérêts ci-dessous (Lambert, Stock, & Ellram, 1998) :

- Surmonter les différences spatiales et temporelles entre client et fournisseur.
- Maintenir une source d'approvisionnement.
- Accomplir des économies lors des transports de marchandise en combinant les transports pour remplir au mieux les cargaisons sortantes.
- Assurer que la politique de service client de l'entreprise est respectée.

- Soutenir la logistique des retours en servant de zone de stockage temporaire avant consolidation pour les centres de recyclage

Les entrepôts sont vite devenus un enjeu pour les entreprises. De plus le coût lié à la gestion d'entrepôt au sein de la supply chain n'est pas négligeable. D'après ELA/AT Kearney (2004), 20% des coûts logistiques incombent aux activités liées à la gestion d'entrepôt.

Pour assurer les intérêts décrits précédemment, les entrepôts sont classiquement répartis en plusieurs zones connectées par des flux. La figure 1 montre ces interactions en décrivant les principales zones présentes dans un entrepôt et activités associées :

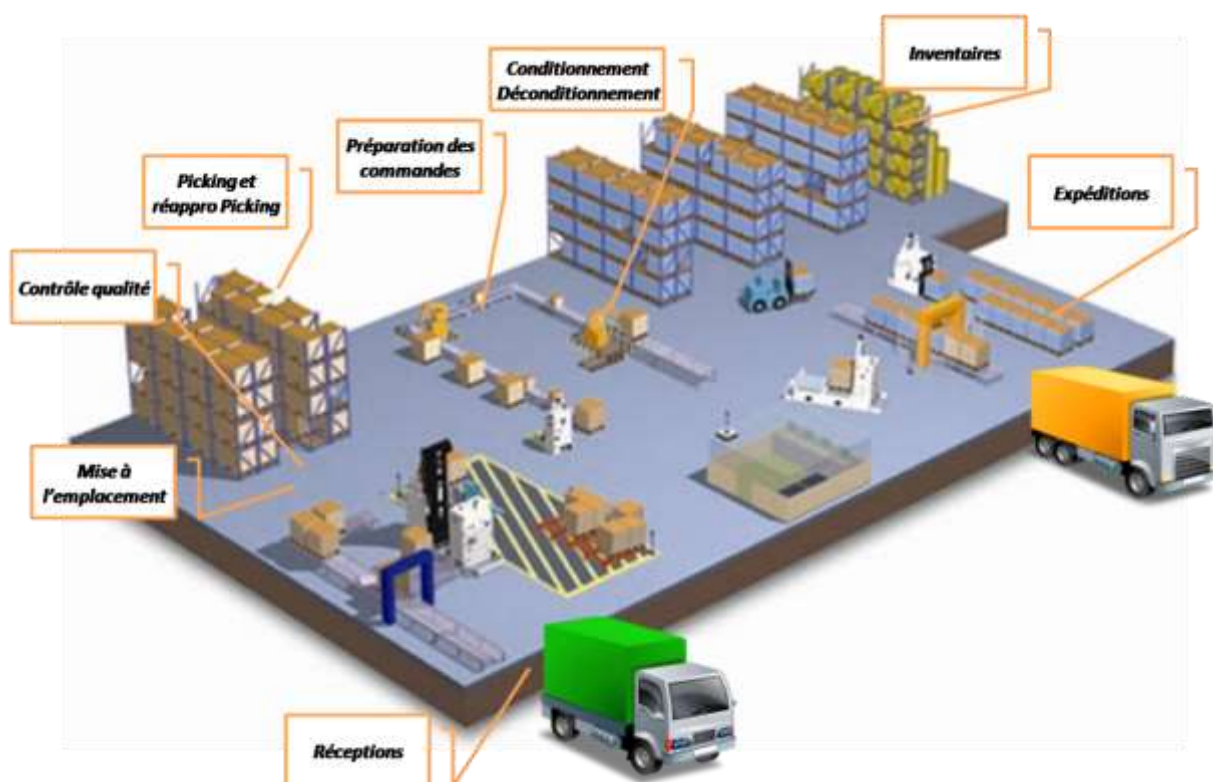


Figure 1 : Présentation d'un entrepôt type avec ses flux associés (tiré de <http://www.cosmos-consultants.eu/h-log-partenaire>)

Nous allons décrire les diverses activités liées à la gestion d'entrepôt ainsi que les principales zones décrites sur l'image ci-dessus. Les produits provenant de fournisseurs arrivent en zone de réception. L'activité de réception permet le déchargement des produits des camions. A cela s'ajoute l'inspection des marchandises pour trouver d'éventuelles casses ou pour signaler les produits ne correspondant pas à ceux présents sur le bon de réception. Une zone de contrôle qualité permet de voir la conformité des produits et les éventuelles défaillances ou casses liées au transport.

Le transfert de ces produits en zones de stockage inclut plusieurs activités dont les activités de déconditionnement (les produits en palette pourront ainsi être déconditionnés puis rangés dans des cartons ou autres formes de conditionnement si besoin). Plusieurs zones de stockage peuvent exister, généralement celles ayant les conditionnements les plus volumineux alimentant celles ayant les conditionnements les moins volumineux (par exemple la zone contenant les produits stockés dans des palettes alimentera la zone de stockage des produits conditionnés dans des cartons etc...). Ces zones de stockage permettent d'entreposer les références qui seront par la suite collectées en fonction des commandes clients.

La zone de préparation de commandes montrée ci-dessus est celle où les commandes seront collectées en fonction des quantités spécifiées par les commandes clients. Ces zones peuvent être manuelles ou automatisées suivant le degré d'automatisation de l'entrepôt (cf. section 2.1.2 Le degré d'automatisation). Ces commandes sont ensuite rassemblées si elles proviennent de plusieurs zones de stockage avant l'expédition chez le client.

Une vérification de l'intégralité de la commande avant expédition est faite afin de s'assurer de sa conformité avec le bon de livraison et le bon de commande. La zone d'expédition permet de charger les camions en fonction du bon d'expédition. Un flux direct entre la réception et l'expédition, appelé cross-docking, permet de faire transiter rapidement les produits sans les faire passer par une zone de stockage.

Notre travail se concentre sur l'activité de préparation de commandes permettant, en fonction des commandes clients, de collecter à partir du stock les produits requis dans les quantités spécifiées par la commande.

2.1.2 Le degré d'automatisation

Le degré d'automatisation est une variable impactant grandement les performances des entrepôts et leurs systèmes de préparation de commandes (Gu et al., 2010). De Koster et al. (2007) classifient les systèmes de préparation de commandes de la façon suivante :

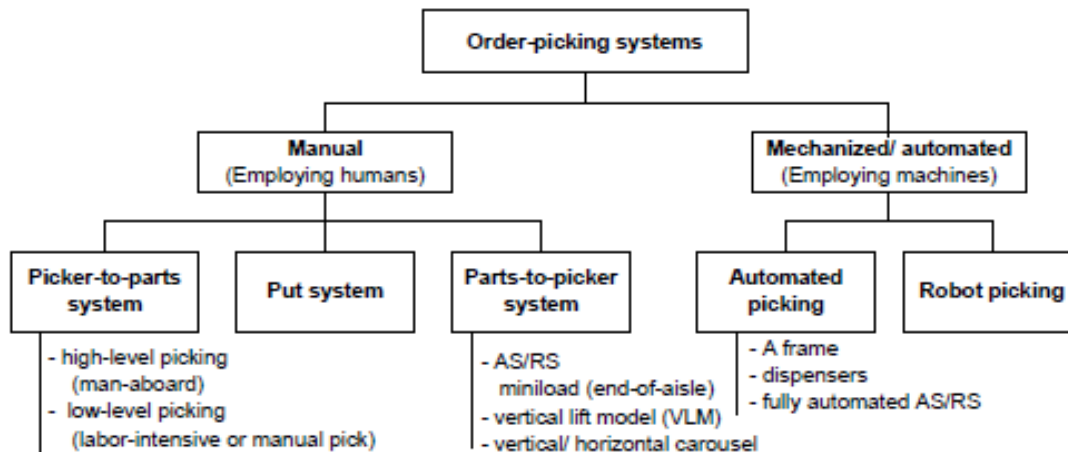


Figure 2 : Classification des systèmes de préparation de commandes (repris de De Koster, 2004)

Cette classification distingue tout d'abord les systèmes de préparations selon la présence ou non d'une personne pour effectuer la préparation de commandes. Le cas le moins fréquent est le système n'employant pas de préparateur, où la préparation de commandes est entièrement automatisée. Dans ce cas d'automatisation totale du système de préparation de commandes, deux cas se présentent, soit l'utilisation de robots, où des bras articulés prennent les produits du stock pour les placer directement dans les cartons/ palettes qui seront placées en zone d'expédition, soit par l'utilisation de machines automatisées de préparation de commandes. Dans ce second cas, un système utilisant une machine de préparation de commandes doublé de systèmes de convoyeurs sont utilisés. La machines place les produits sur le convoyeur qui les conduit jusqu'en zone d'expédition. Notons que ces systèmes restent encore très peu utilisés dans la majorité des entrepôts car demandent un investissement initial colossal, et donc un nombre de commandes à effectuer extrêmement grand pour rendre rentable le système automatisé.

Le second type de système de préparation de commandes est celui où il y a présence de préparateurs de commandes. Dans un tel cas, la façon dont le préparateur de commandes se déplace par rapport aux références à collecter permet de caractériser le système utilisé.

- Soit le préparateur de commandes se déplace jusqu'aux produits, ce système est nommé « Picker to Part ». Dans ce cas, l'opérateur chargé de la préparation de commandes se déplace dans les allées pour chercher les références à collecter. Il peut être à pied ou véhiculé.
- Soit ce sont les produits qui sont apportés au préparateur de commandes, ce système est nommé « Part to Picker ». Ce cas nécessite l'utilisation de stockeurs automatisés, appelés Automated Storage and Retrieval Systems (AS/RS) pouvant être, par exemple, de type

Vertical Lift Module (VLM) ou Carousel. Nous nous attarderons sur ces systèmes automatisés de préparation dans la Section 1.3.

- Nous pouvons noter qu'il existe aussi des systèmes mêlant le fait que les produits sont collectés de façon « Picker to Part » et/ou de façon « Part to Picker ». Ce système est appelé « Put system ». Ce dernier système décrit par De Koster (2007) est le moins présent en littérature ainsi qu'en pratique.

Il est à noter que d'après Burinskiene (2015), 75% des entrepôts fonctionnent avec un prélèvement manuel des références.

Nous nous focalisons, dans la suite de ce chapitre, sur les dimensions impactant la préparation de commandes, et plus précisément sur les activités de regroupement de commandes en lots (ou batching) et de séquençement des commandes. Ces activités permettent une réduction des coûts de préparation de commandes grâce à un plus grand débit de commandes préparées. De plus, nous positionnons notre étude sur les systèmes de préparation de commandes de type Part to Picker, utilisant les machines de type Carousel ou VLM. La section suivante décrit les activités impactant la préparation de commandes et la littérature correspondante.

2.2 Les activités impactant la préparation de commandes

Cette section est dédiée à l'étude de la littérature liée à l'activité de préparation de commandes et les dimensions l'impactant. Un système de préparation de commandes se révèle complexe par le nombre de facteurs impactant ce système. D'après Goetschalckx and Ashayeri (1989), ces facteurs se divisent entre les facteurs externes et facteurs internes.

- Les facteurs externes comprennent un grand nombre de paramètres, tel que les modèles de réapprovisionnement auprès des fournisseurs ainsi que les types de commandes des clients, l'objectif concernant les niveaux de stocks, les canaux de commercialisation, la demande globale pour les produits entreposés et enfin l'état de l'économie. Cependant ces facteurs (dont certains macroéconomiques) sont difficiles à modéliser et contextualiser, et donc en dehors du cadre de notre étude.
- Les facteurs internes incluent les caractéristiques organisationnelles, les politiques opérationnelles de l'entreprise ainsi que les caractéristiques du système de préparation que l'entreprise veut mettre en place, prenant en compte le degré d'automatisation, les caractéristiques physiques de l'entrepôt et la disponibilité des informations concernant les commandes. Les facteurs internes sont montrés dans la Figure 3 (basée sur les travaux de Goetschalckx and Ashayeri (1989)). Cette Figure représente aussi le degré de complexité d'un système de préparation de commandes, qui peut se mesurer par la distance sur chacun des axes par rapport à l'origine.

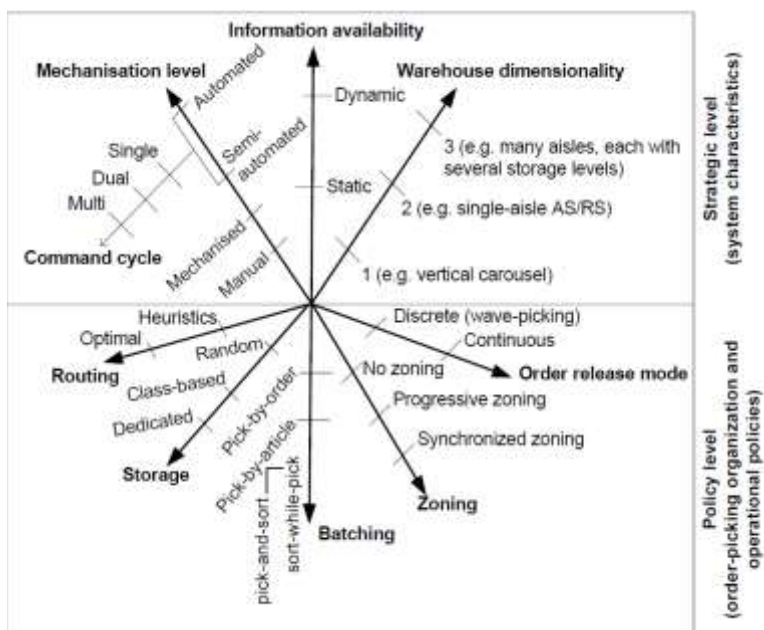


Figure 3 : Illustration de la complexité du design d'un système de préparation de commandes (repris de Goetschalckx and Ashayeri 1989)

De Koster et al. (2007) résume cette classification en divisant la littérature existante en plusieurs dimensions affectant la préparation de commandes. En prenant ces dimensions en compte, nous avons extrait 4 dimensions principales affectant la préparation des commandes :

- **L'agencement et le Zonage de l'entrepôt**
- **L'affectation des produits à des emplacements de stockage donnés**
- **Le parcours de cueillette**
- **Le regroupement des commandes en lot (ou batching) et le séquençement des commandes**

Dans cette section, nous allons introduire ces différentes dimensions puis étudier la littérature correspondante, en gardant à l'esprit que cette thèse se focalise sur les activités de regroupement de commandes en lots ainsi que le séquençement des commandes (dernière des dimensions précédemment citées). Nous décrivons donc rapidement pour les trois premières dimensions la littérature pour comprendre les travaux effectués, avant de nous attarder sur la dimension de regroupement de commandes en lot et de séquençement des commandes. Nous notons, par ailleurs, que l'essentiel de cette section concerne les systèmes « Picker to Part ».

2.2.1 L'agencement et le zonage des entrepôts

Ces deux activités, l'agencement de l'entrepôt et le zonage, sont généralement dissociées, bien que très corrélées en réalité. La dimension concernant l'agencement définit comment équiper l'entrepôt en matière de zones de stockage avec le positionnement des différentes zones (notamment celles de réception et d'expédition par rapport à la zone de stockage). Un dimensionnement doit être décidé concernant le nombre d'allées ainsi que la position relative de ces allées par rapport aux zones. Le zonage permet de diviser l'entrepôt en zones, puis les zones en sous-zones afin de répartir la charge de travail auprès des différents préparateurs de commandes assignés aux différentes zones.

D'après Caron et al. (2000), l'agencement ainsi que le zonage de l'entrepôt ont un impact considérable sur la distance parcourue et plus de 60% de la distance totale parcourue provient de cet agencement. Ces deux aspects sont intrinsèquement liés. En effet la constitution de zones au sein d'un entrepôt doit se faire en tenant compte de l'agencement et vice versa. Si la préparation doit se faire sur une zone froide ou une zone pour matériaux dangereux, il faut l'anticiper lors de l'agencement (pour prévoir le nombre de blocks, de racks et allées, etc...) pour que le préparateur affecté à cette zone n'ait pas à traverser diverses sous-zones pour la préparation de commandes.

2.2.1.1 L'agencement d'entrepôt

L'agencement d'entrepôt est divisé dans la littérature en deux sous-activités, la première décrivant les caractéristiques des zones les unes par rapport aux autres, la seconde se focalisant sur la zone de préparation en décrivant l'agencement au sein même d'une zone de stockage.

La première sous-activité est faite de manière relative (les zones sont agencées les unes par rapport aux autres), appelée « facility layout problem » ou problème « d'agencement de l'établissement ». Ce premier point est peu étudié dans la littérature. Meller et Gau (1996a) puis Tompkins et al (2010) décrivent plusieurs agencements d'entrepôt dits « efficaces » et des procédures permettant d'agencer les diverses zones les unes par rapport aux autres. Meller et Gau (1996b) donnent une description exhaustive de la littérature concernant ce premier problème. Plus récemment Heragu et al. (2005) proposent des modèles et des heuristiques permettant de quantifier ainsi que d'allouer les produits relatifs aux diverses zones.

La seconde sous-activité est l'étude du nombre de blocks par zone, d'allées par block, de racks par allées ainsi que leur dimensionnement, communément appelé « Aisle configuration problem » ou bien « problème de configuration des allées ». Ce second problème d'agencement est directement lié à la préparation de commandes. Nous nous attarderons dessus en étudiant la littérature concernant les différentes configurations de la zone de préparation de commandes. De Koster et al. (2007) affirment que les facteurs qui doivent être considérés sont : le nombre de blocks, les dimensions des allées, leurs nombres ainsi que leurs formes, le nombre de racks ainsi que les emplacements de points d'entrée/ sortie (« I/O » sur les Figures suivantes). Ces facteurs sont résumés par la figure 4 ci-dessous :

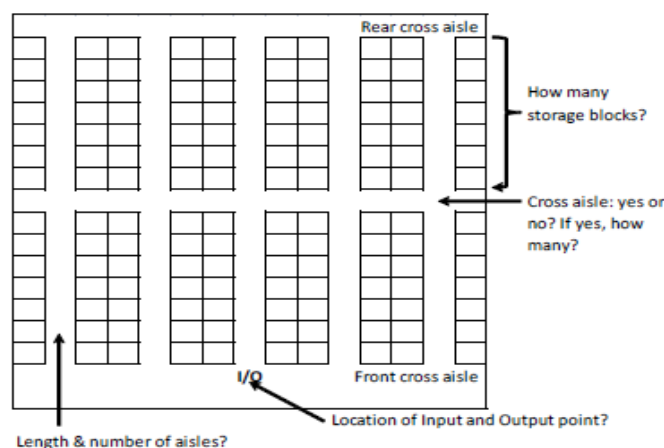


Figure 4 : Principales décisions devant être prises lors du design d'entrepôt (repris de De Koster et al., 2007)

De Koster et Roodbergen (2001) considèrent le système décrit dans la figure précédente (Figure 4) et justifient leurs choix par une amélioration de la distance moyenne de picking. De plus cet agencement est utilisé communément en industrie et dans le secteur de la grande distribution. Gue et Meller (2009), décrivent deux autres systèmes montrés ci-dessous étant moins traditionnels (Figure 5) mais qui offriraient une réduction de la distance de picking de 10% pour le Flying V et 20 % pour le Fishbone par rapport aux systèmes présentés précédemment :

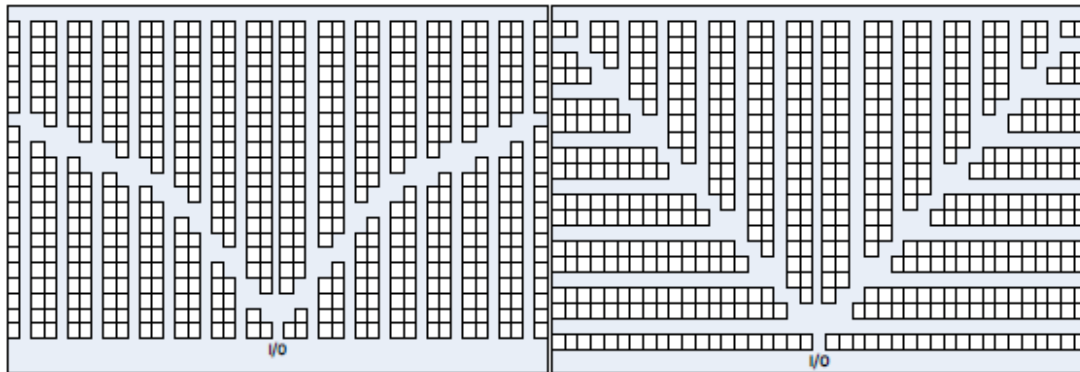


Figure 5 : Le flying V et fishbone design présentés par Grue et Meller 2009

Les principaux objectifs de l'agencement de l'entrepôt sont de pouvoir minimiser le temps de picking ou la distance moyenne parcourue (De Koster et al., 2010), cependant la maximisation de l'utilisation d'équipement ou la minimisation de l'espace requis peuvent aussi être pris en considération (De Koster et al., 2007). Certains algorithmes définissant le nombre de zones sont présentés par Corley and Roberts (1972) ; une heuristique utilisant la théorie des graphes est proposée par Kim and Kim (1995). Ces algorithmes proposent de minimiser la distance parcourue par l'opérateur pour un nombre de stockage et/ou de préparation plutôt que le coût. Une autre solution est d'utiliser un algorithme « Cut tree » car « facile et puissant afin d'aider les concepteurs d'entrepôts à générer des mises en place performantes » (Montreuil et al., 1989). Dans cet article, une autre solution hybride combinant programmation linéaire et théorie des graphes est utilisée pour conceptualiser l'agencement qui inclut les allées.

2.2.1.2 Le zonage

L'avantage principal de subdiviser l'entrepôt en zones est la réduction du temps de déplacement (car des zones plus petites impliquent que les préparateurs ont une plus grande connaissance de leurs zones) ainsi qu'une congestion plus faible. Cependant un plus grand nombre de zones implique aussi une plus grande répartition des commandes qui doivent être réunies avant la livraison aux clients et donc plus de personnel (De Koster et al., 2007). Une façon de classer les produits par zone est faite en fonction de leur besoin de conditionnement, comme une restriction

sur la température par exemple. Frazelle (2002) propose une classification permettant de décider des emplacements pour les produits (Cf. Figure 6) :

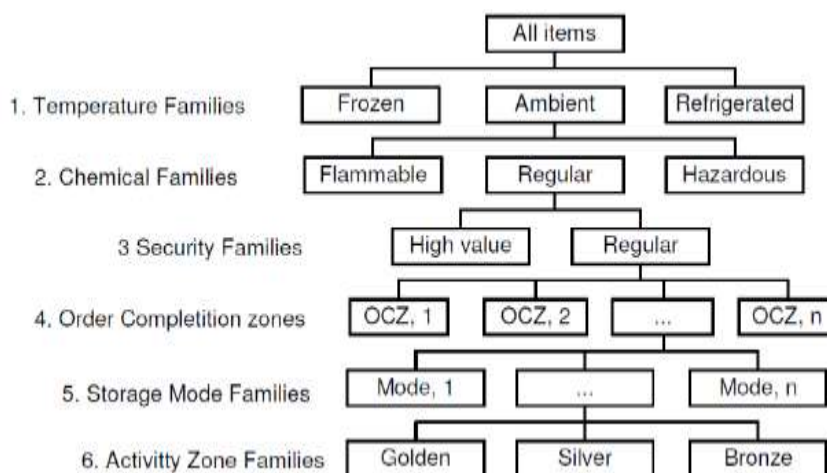


Figure 6 : Arbre de décision des zones selon Frazelle (2002)

Cette classification permet seulement de créer et affecter des zones spécifiques par type de produit. D'autres méthodes d'affectation ainsi que de zonage à l'intérieur de ces zones répertoriées sont utilisées afin de pouvoir générer une subdivision adéquate et performante de l'entrepôt. La question du nombre optimal de zones a été traitée par De Koster et al. (2010) : dans un premier temps affecter les produits aux emplacements (cf. 2.2.2) présent dans la zone de stockage puis définir un parcours de cueillette pour la collecte des produits (cf. 2.2.3). Le chemin le plus court permettant la collecte donne la distance à parcourir grâce à un modèle d'optimisation. Ce modèle est utilisé pour divers nombres et taille de zones afin de trouver la combinaison donnant la plus faible distance à parcourir.

Dans cette section nous avons vu les différents impacts de l'agencement de l'entrepôt et du zonage sur la préparation de commandes. La prochaine section est consacrée à l'affectation des produits au sein du stock.

2.2.2 L'affectation de produits aux emplacements

L'affectation des produits à un emplacement de stockage est l'activité permettant, dans l'idéal, d'assigner un produit au « meilleur emplacement ». Un « meilleur emplacement » fournit un accès plus rapide et plus ergonomique aux produits stockés (Shouman et al. 2005). Il faut choisir un meilleur emplacement de par les coûts engendrés par l'activité de stockage que l'entreprise cherche à rentabiliser. Ainsi optimiser la place disponible est crucial et donc un choix des emplacements pour les produits est nécessaire.

D'après Chan et Chan (2011), il y a plusieurs facteurs affectant la politique d'affectation d'un emplacement à un produit : le plan de cueillette, l'équipement utilisé ainsi que le zonage de l'entrepôt. A cela s'ajoute les caractéristiques intrinsèques aux produits : ses dimensions physiques et conditions de stockage.

Il y a deux principales stratégies d'affectation des produits aux stocks (Bartholdi and Hackman, 2011), le stockage à emplacements partagés et le stockage à emplacement dédiés.

Le Stockage à emplacements partagés implique de ne pas affecter d'emplacement fixe à un produit. Ainsi lorsqu'un emplacement devient vide, il est disponible pour tout type de produit. Ceci évite d'attendre que le produit soit réapprovisionné afin de remplir l'espace (voir Stockage à emplacement Dédié, paragraphe ci-après). Le principal désavantage de l'utilisation d'une telle pratique est le fait que les préparateurs de commandes ne peuvent se familiariser aux emplacements de stockage, impliquant donc un temps de recherche des références plus important. La littérature discerne généralement deux types de stockages partagés :

- Le stockage dit « aléatoire » qui implique que chaque nouveau produit est assigné avec probabilité égale à n'importe quel emplacement. De ce stockage résulte une grande utilisation de l'espace au prix d'une augmentation de la distance parcourue (Choe and Sharp, 1991).
- Une extension à ce modèle est le stockage dans « l'emplacement le plus proche », qui, comme son nom l'indique, choisit pour stocker un produit l'emplacement le plus proche disponible, si deux emplacements ont la même proximité, le choix se fera aléatoirement entre les deux emplacements désignés.

Le Stockage à emplacement Dédié alloue un emplacement prédéfini aux produits. L'avantage est de savoir où le produit est stocké et donc familiarise le préparateur de commandes avec les produits. Ainsi le temps de recherche du produit devient plus faible. Cependant la place n'est pas utilisée de manière optimale de par le fait que si un article subit une rupture, l'emplacement restera vide. Plus généralement, lorsque l'emplacement défini à un article se vide par collectes successives, il ne pourra être rempli que lorsque le même produit sera commandé et livré, ceci créant une non utilisation de l'espace entre la dernière collecte d'un produit et sa prochaine mise en stock.

- Parmi les plus utilisés et cités dans la littérature, le stockage basé sur le COI (Cube per Order Index) : ce mode de stockage a été établi par Heskett en 1963 et consiste à ranger les

produits en fonction d'un indice, le COI, qui est défini par : $COI_i = \frac{\text{Volume occupé par produit } i}{\text{Demande du produit } i}$.

Les produits ayant l'indice COI le plus faible sont rangés le plus proche de la zone de sortie. Plus d'information concernant l'application et les avantages du COI sont disponibles dans Kallina et Lynn (1976), Malmborg et Bhaskaran (1987, 1989 et 1990).

- D'autres politiques de stockage sont proposées par la littérature, par exemple celle plaçant les produits ayant des interactions fortes à côté, ce que fait la « Order oriented slotting methods » proposée par Mantel et al. en 2007. Les produits étant communément collectés ensemble seront positionnés côte à côte.
- Une autre politique incontournable de stockage est celle basée sur la classe. D'après Zhao et al. (2012) cette politique implique de ranger les produits par une caractéristique commune. Cette méthode mêle les deux types d'affectation aux stocks : des classes sont créées dépendant d'un ou plusieurs attributs (turnover, demande annuelle, dimension du produit, etc..). Les produits possédant les mêmes attributs seront rangés proches les uns des autres. Un rangement aléatoire est par la suite effectué sur ses « sous zones » créées en fonction d'un attribut commun. Une manière classique de pouvoir les placer est la méthode de Pareto. Cette méthode, généralement appelée **méthode ABC** dans les entrepôts, définit ainsi 3 zones qui peuvent être inter ou intra allées comme défini par la figure ci-dessous (De Koster et al., 2006) :

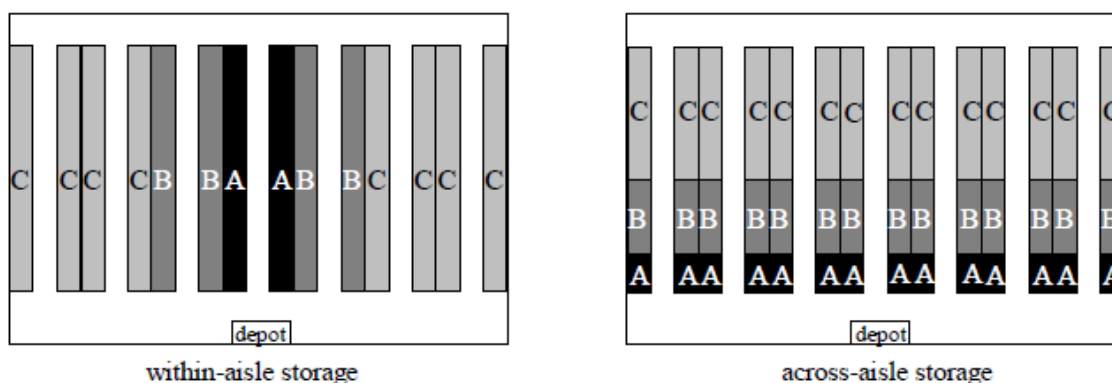


Figure 7 : Méthode ABC inter et Intra allée De Koster et al, (2007)

Le nombre de classes et leurs dimensionnements ont longtemps été débattus dans la littérature : par exemple une seule classe par allée pour Jarvis and Mcdowell (1991), donnant lieu à une réduction de la distance parcourue de 10 à 20 % indépendamment du parcours de cueillette et un maximum de 4 classes pour Petersen et al. (2004). Ce dernier article préconise un maximum du nombre de classes de 2 à 4 en fonction des caractéristiques de l'entrepôt ainsi que des différences entre les produits. Le-Duc et De Koster (2005a), Le Duc et De Koster (2005b) et Le Duc (2005)

traitent de l'optimisation du nombre de zones pour un plan de cueillette particulier ainsi que de l'estimation de la distance parcourue pour collecter une série de commandes en utilisant la méthode des classes ABC. Enfin le nombre maximum de classes de produits dans le cas de stockeurs automatisés est 6 d'après les études de Yang (1988) et Van den Berg et Gademan (2000).

Le défaut des politiques de stockage basées sur les classes ou la politique de stockage COI est l'encombrement généré par les opérateurs devant les zones à fortes affluences (Le-Duc, 2005). Ces politiques de stockage créent des embouteillages car certaines parties de la zone de picking sont beaucoup plus demandées par l'ensemble des préparateurs que d'autres zones.

D'autres politiques d'affectation d'emplacement ont été créées par des études spécifiques et ne peuvent pas être répertoriées dans les catégories énoncées ci-dessus. Nous citons par exemple Marvel et al (2001) qui développent une heuristique qui intègre d'une part l'expertise humaine d'entrepôt dans le secteur alimentaire et d'autre part un système d'optimisation afin de sélectionner les zones d'entrepôt adéquates. Examiner et quantifier les décisions actuelles qui déterminent l'affectation d'un produit à un stock donné sont à la base de la création de l'heuristique.

Les objectifs utilisés sont divers pour le choix d'affectation aux emplacements, le choix se porte sur minimiser la distance parcourue par les préparateurs dans l'essentiel des cas (COI, stockage basé sur la classe) ou alors l'espace utilisé (stockage partagé). L'aspect financier est parfois préféré par exemple dans Malmberg et al (1986) où la minimisation du coût de stock et du coût de picking, suivant la politique de gestion des stocks, est faite. Nous voyons donc que le choix d'une politique de stockage peut avoir de grands effets sur la préparation de commandes. Maintenant que nous avons vu les différentes façons de définir les emplacements des produits, nous allons étudier comment les produits sont prélevés du stock par la dimension « routing ».

2.2.3 Le routing

La définition d'un parcours de cueillette ou routing a pour objectif de fixer la séquence optimale d'emplacements à visiter par le préparateur afin de pouvoir collecter les produits présents dans sa liste de picking de la manière la plus efficace possible. Ce problème peut être vu comme un cas particulier du problème du voyageur de commerce, largement discuté au sein de la littérature (Lawler et al., 1985). Pour définir une préparation de commandes optimale, nous devons imposer une séquence de collecte des produits. Cette séquence va définir les emplacements à visiter pour

la préparation de commandes et définir un parcours de cueillette. La littérature discutant de ce problème se focalise essentiellement sur les systèmes « Picker to Part ».

Une manière de représenter ce parcours de cueillette est de transposer l'ensemble des produits à collecter sur un graphe présentant les allées comme étant les arcs et si un produit doit être collecté au sein d'une allée, un nœud le représentant est à visiter (De Koster et al. ,2007) :

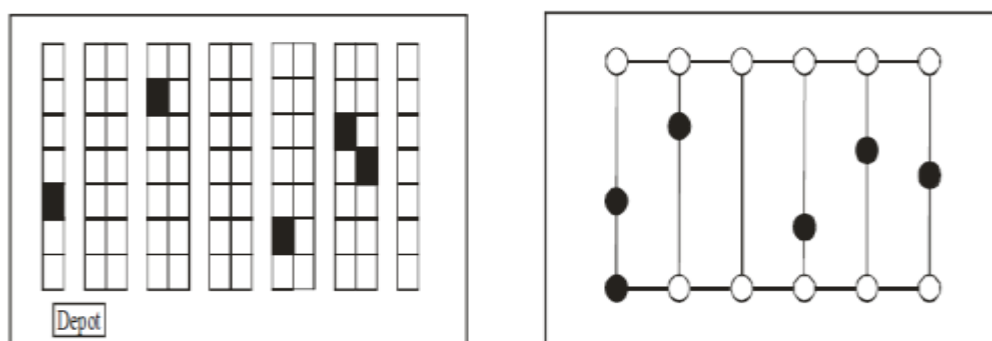


Figure 8 : Illustration des emplacements d'une commande à collecter et sa représentation graphique.

L'importance apportée à ce domaine s'est significativement accrue lors des dernières décennies par l'arrivée des nouveaux concepts de Just In Time (JIT) ou Cycle Time Reduction par exemple. Dans la pratique le parcours de cueillette est fourni par des heuristiques fournissant un temps de calcul raisonnable. De plus les routes optimales ne sont pas forcément suivies par les préparateurs qui dévient de la route donnée (Gademann et Van de Velde en 2005).

Selon De Koster en 2007, les algorithmes de résolution exacts, ne prennent pas en compte les embouteillages jusqu'à présent car accroissent le temps de calcul tandis que les heuristiques peuvent les prendre en compte plus facilement.

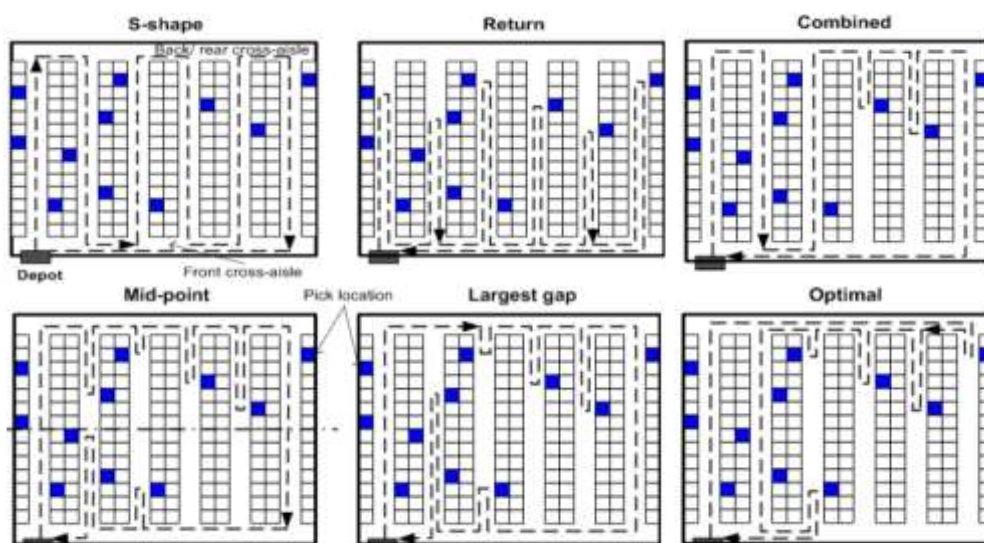


Figure 9 : Différentes heuristiques et méthode optimale présentées par Roodbergen en 2001

Le parcours le plus simple est dénommé le « S Shape », ainsi chaque allée contenant un produit à collecter est visitée dans toute sa longueur par le préparateur et les allées démunies de produit à collecter ne sont pas visitées. Après la dernière allée contenant un produit à collecter, le préparateur rentre au dépôt. L'heuristique « Return » fonctionne de la manière suivante : pour chaque allée où un produit à collecter est présent, le préparateur va rentrer et repartir par l'entrée de l'allée indépendamment de la distance entre l'entrée de l'allée et l'emplacement du produit. L'heuristique « Midpoint » est un cas particulier de « Return » et divise les allées en deux de la manière montrée sur la figure 9. L'heuristique « Largest Gap » est similaire au « Midpoint » mais l'opérateur revient sur ses pas si, pour chacune des allées, la distance entre le prochain emplacement de collecte et l'actuel est plus grand que la distance pour retourner à l'entrée de l'allée. Hall (1993) prouve que l'heuristique « largest gap » fournit toujours de meilleurs résultats que « Midpoint ». Enfin l'heuristique « Combined » combine l'ensemble de ces méthodes grâce à l'utilisation de la programmation dynamique (Roodbergen, 2001). La figure 9 récapitule les différentes heuristiques citées précédemment.

Afin de comparer et choisir entre ces heuristiques, Petersen (1997) conclut qu'entre la solution optimale et la meilleure méthode heuristique, un écart d'environ 5% en termes de distance parcourue existe. Il est donc possible d'atteindre une solution quasi-optimale par l'utilisation d'heuristiques très simples. De Koster et Poort (1998) montre qu'une augmentation d'environ 10% de la distance à parcourir est obtenue par l'utilisation du S-Shape par rapport à l'optimal mais le temps de calcul est divisé par 15.

Roodbergen et De Koster (2001) fournissent une approche énoncée par De Koster (2010) comme étant « probablement la meilleure heuristique combinée faite jusqu'alors ». En ne se cloisonnant pas à un seul block mais en considérant des nombres d'allées et croisements d'allées différents, elle améliore encore l'heuristique « combined ». Cette heuristique ainsi créée combine deux méthodes basiques : soit traverser entièrement une allée devant être visitée, soit entrer et sortir par le même côté (ce qui peut être vu comme un choix entre S Shape ou return pour chaque allée), le choix se faisant par un programme dynamique. Nous observons donc que la façon dont les produits seront collectés a un impact sur les performances de la préparation de commandes.

La prochaine section est dédiée à la littérature concernant le thème central de la présente thèse, l'activité de regroupement de commandes en lots ainsi que le séquençement des commandes

2.2.4 Le regroupement de commandes en lots complets (Batching) et séquençement des commandes

Le problème de regroupement des commandes en lots complets ou encore Order Batching Problem (OBP), consiste à déterminer quelles commandes seront collectées ensemble par l'opérateur. Le séquençement des commandes ou bien l'ordonnancement des commandes ou encore « order sequencing » consiste à choisir dans quel ordre les commandes seront mises en préparation dans le système de préparation de commandes.

Ces deux activités se révèlent cruciales dans un but d'optimisation des performances de l'activité de préparation de commandes (Pazour and Meller, 2013). Le regroupement de commandes en lots ainsi que le séquençement des commandes au sein d'une zone de picking sont des sujets apparus dès les années 80 dans la littérature scientifique (Armstrong et al, 1979 ; Bartholdi & Platzman, 1986).

Le regroupement de commandes en lots est un sujet très longuement abordé dans la littérature considérant un système Picker to part. Nous notons aussi que le séquençement des commandes est beaucoup moins traité que le regroupement de commandes en lots au sein de la littérature. De plus, pour les papiers étudiant le séquençement, nous observons que ce sujet est plus traité avec les systèmes automatisés (systèmes de type « Part to Picker »).

2.2.4.1 Le regroupement de commandes en lots complets ou batching

Il est souvent plus efficace de pouvoir affecter plusieurs commandes à la fois à un préparateur. Ce regroupement peut se faire si les commandes sont suffisamment petites ou si le préparateur détient une capacité de cargaison suffisante, ou alors si le système de préparation est automatisé grâce à un système de convoyeur. Cette dimension est définie dans la littérature comme « **comment combiner différentes commandes clients au sein d'une même tournée** » (Roodbergen et Vis, 2009). De plus, le regroupement de commandes en lots est considéré comme la dimension pouvant amener les plus grands gains, et en particulier lorsque les commandes sont constituées de peu de produits (Petersen et Aase, 2004).

Le problème de regroupement de commandes est défini comme étant un problème NP-Difficile comme le prouvent Gademann et al. (2001) dans le cas Picker to part, et Gademann & Van de Velde (2005) prouvent qu'avec l'objectif de minimisation du temps de trajet total pour collecter une séquence de commandes données, l'OBP est toujours un problème NP-difficile lorsque le

nombre de commandes par lots est plus grand que 2. Ainsi les algorithmes pour la résolution exacte de ces problèmes ont un temps d'exécution exponentiel en la taille des données d'entrée.

Il y a deux types de regroupement de commandes en lots complets dans la littérature : le regroupement statique et le regroupement dynamique. Le regroupement de commandes en lots complets dit « statique » implique une connaissance de toutes les commandes lors du début d'une période (Henn, 2012). Connaissant cela, un regroupement est effectué pour chacune des périodes considérées (tous les jours, demi-journées, etc...). Cette méthode est prépondérante dans la littérature car le système dynamique est récent avec, notamment, l'arrivée du e-commerce. Donc la majorité des articles traitant le regroupement des commandes aborde le problème statique (Henn et al., 2012). L'information sur la commande, dans le cas du regroupement dynamique, est transmise au fil de l'eau, impliquant qu'une commande venant d'être reçue peut-être incluse dans le prochain lot de commandes à être collecté.

Nous nous intéressons tout d'abord au regroupement de commandes en lots complets statique ou batching statique. Comme expliqué dans le paragraphe précédent, cette pratique vise à effectuer un regroupement de commandes en lots toutes les périodes données. Ceci implique la connaissance de toutes les commandes à regrouper dès le début de la période. En pratique, à partir d'un portefeuille de commandes, un regroupement est fait en ayant un objectif et certaines contraintes tous les matins ou plusieurs fois par jour. Ainsi la liste de picking créée est fixe entre deux périodes.

Pour un système Picker to part, l'objectif prédominant est la minimisation de la distance parcourue par les préparateurs. Un second objectif peut être utilisé : minimiser le temps de collecte d'un ensemble de lots de commandes. Le modèle linéaire ci-dessous est établi par Gademann & Van de Velde (2005). Celui-ci prend en compte la capacité du préparateur, la contrainte (2-1) imposant que la somme des commandes dans un même lot n'excède pas la capacité maximum du préparateur. La distance parcourue est l'objectif à minimiser (2-2). La contrainte (2-3) impose que les commandes soient dans un seul lot. L'article traite d'un entrepôt avec 400 emplacements et 30 clients différents, et jusqu'à 10 commandes pouvant être combinées comme contrainte de capacité.

J	Ensemble des commandes clients où $J = \{1, \dots, N\}$
C	Capacité du système de picking
c_j	Capacité requise par la commande j ($j \in J$)
I	Ensemble des lots faisables
d_i	Distance requise pour collecter toutes les commandes du lot i ($i \in I$)
Soit $a_i = (a_{i1}, \dots, a_{in})$ le vecteur des paramètres binaires a_{ij} , indiquant si la commande j ($j \in J$) est incluse dans le lot i ($a_{ij} = 1$) ou non ($a_{ij} = 0$). Ainsi l'ensemble des lots faisables est caractérisé par le fait que la contrainte de capacité n'est pas violée :	
$\sum_{j \in J} c_j * a_{ij} \leq C \quad (2 - 1)$	
De plus la variable de décision utilisée est :	
x_i	Variable binaire, $x_i = 1$, si le lot i est choisi, ou $x_i = 0$, sinon.
Le modèle d'optimisation peut être formulé par :	
$\text{Min } \sum_{i \in I} d_i * x_i \quad (2 - 2)$	
Les contraintes du modèle sont :	
$\sum_{j \in J} x_i * a_{ij} = 1, \forall j \in J \quad (2 - 3)$	
$x_i \in \{0,1\}, \forall i \in I \quad (2 - 4)$	

Figure 10 : Modèle d'optimisation pour le regroupement de commandes en lots (proposé par Gademann & Van De Velde, 2005)

Nous pouvons bien imaginer que plus le nombre de commandes et le nombre de produits devient grand, plus la recherche de la solution se complexifie. Il est important de garder en tête que le nombre de lots possible grandit exponentiellement avec le nombre de commandes. C'est pourquoi l'utilisation d'heuristiques est utile afin de fournir une solution dans un temps acceptable qui soit adapté au mieux aux configurations d'entrepôt réel (en termes de nombre de commandes notamment). Aussi de nombreuses heuristiques et méta-heuristiques ont été étudiées afin de résoudre ce problème en un temps de résolution raisonnable.

Hsu et al (2005) utilisent un algorithme génétique qui montre un temps de résolution acceptable pour les expériences allant jusqu'à 300 commandes et 200 produits différents.

Henn et Wäscher en 2012 comparent deux méta-heuristiques afin de regrouper les commandes en lots. De plus, plusieurs heuristiques de parcours de cueillette sont étudiées afin de montrer que les solutions apportées par les deux méta-heuristiques peuvent être adaptées à différents routings. La méta-heuristique utilisée est de type « recherche tabou ». L'application numérique est plus adaptée à notre cadre d'étude en prenant en compte jusqu'à 900 emplacements de

stockage, et jusqu'à 100 commandes comprenant au maximum 25 références différentes à collecter par commande. Pan et al (2012) utilisent une heuristique pour associer les commandes en fonction de la similarité des parcours de cueillette afin de créer des lots. La capacité est prise comme contrainte et 3 différents parcours de cueillette sont utilisés pour créer des similarités entre les produits. Les commandes sont donc regroupées en lots en fonction des similitudes des articles présents au sein des différentes commandes. Cependant l'application numérique n'est pas adaptée car elle ne prend en compte qu'un très faible nombre de commandes à regrouper (maximum 10 commandes). D'autres articles ne conviennent pas à notre étude en raison de la faible étendue des études de cas, mais nous montre l'utilisation de différentes méthodes de résolution comme par exemple l'utilisation d'un algorithme génétique par Koch and Wäscher (2011) avec un nombre maximum de 60 commandes.

Jusqu'à présent la seule contrainte étudiée était la capacité du préparateur. Cependant en pratique, la date de livraison doit aussi être respectée. La prise en compte de ce paramètre comme contrainte donne lieu à des changements d'objectif. Ainsi dans la littérature, une minimisation non plus de la distance parcourue mais des retards a été étudiée par Henn et Schmid (2011). Une priorité plus grande est donnée aux commandes dont la date de livraison est la plus proche. La contrainte de capacité est toujours présente. La méthode de résolution utilisée est la métaheuristique de type « recherche locale itérée ». L'application numérique, montre une vitesse de calcul respectable pour 80 commandes et jusqu'à 25 articles par commande. Chen et al (2015) utilisent le même objectif et y incluent aussi le routing grâce à l'utilisation conjointe de deux métaheuristicques de type « algorithme génétique » et « colonie de fourmis ». La première métaheuristique crée les lots en minimisant les retards totaux et la seconde minimise le chemin pour un lot donné.

Azadnia et al (2013) étudient l'optimisation des retards moyens des commandes, cependant l'étude est restreinte à un seul préparateur et un stockage dédié est choisi comme politique d'affectation des produits à un emplacement. Des poids sont donnés à chaque commande dépendant de sa date de livraison et une métaheuristique de type « algorithme génétique » permet une création et séquençement des lots.

Nous avons jusqu'alors uniquement décrit le système statique de regroupement des commandes en lots. Cependant avec l'arrivée de l'e-commerce, le système statique montre ses limites et un système dynamique va progressivement se développer pour certaines applications. Le regroupement en lots de manière dynamique permet à un préparateur d'effectuer un

regroupement des commandes considérant les commandes en attente de préparation et donc de reformer des lots en fonction des commandes qui arrivent en temps réel.

Ainsi les temps d'arrivées des commandes et de livraisons des commandes peuvent être considérés. Iraj et al. (2014) décrivent un modèle minimisant les retards totaux des commandes en considérant les commandes arrivant durant une période donnée. C'est un modèle sur le problème de « l'online batching » (le regroupement de commandes en lots complets dynamique). Ainsi « à chaque temps t , une décision doit être prise concernant toutes les commandes non encore collectées ». Cet article nous fournit une application numérique intéressante : jusqu'à 400 commandes pouvant contenir entre 5 et 25 lignes de commandes, une capacité de picking au maximum de 75 unités, la date de livraison pouvant être aléatoire ou suivre une distribution uniforme. De plus, une politique d'affectation au stock basée sur la classe est utilisée (52% des produits demandés appartient à la classe A, 36% à la classe B et 12% à la classe C). Cependant rien n'est dit sur les temps de calcul pour obtenir les solutions.

Bukchin et al (2012) utilisent les chaînes de Markov afin de traiter le problème du regroupement de commandes en lots complets en approche dynamique. Une minimisation des pénalités impliquées par les retards est élaborée en ayant pour contrainte la capacité. Le procédé d'arrivée des demandes est modélisé par une distribution de poisson. Malheureusement une restriction est faite puisque le temps de cueillette dépend seulement du nombre de produits et non de leurs emplacements au sein de l'entrepôt. Cette contrainte simplifie grandement l'étude et ne permet pas de faire une cartographie des emplacements pour effectuer une collecte réaliste.

2.2.4.2 Le séquençement des commandes

Définir l'ordre dans lequel les commandes seront collectées est un sujet peu traité dans la littérature. Dans un système Picker to Part, la littérature considère des regroupements de commandes en lots complets. Henn et Schmidt (2013) créent des lots de commandes puis les séquent. Ainsi un séquençement des lots est réalisé avec l'objectif de minimisation des retards. Cet article très intéressant est focalisé sur les entrepôts manuels ayant une zone de préparation de commandes de type « Picker to Part ». Hong et al. (2012) présentent un modèle joignant le regroupement en lots et le séquençement des commandes permettant d'optimiser le temps de collecte dans un entrepôt traditionnel, tout en considérant plusieurs opérateurs et la congestion. Un recuit simulé est utilisé afin de trouver une séquence des lots à collecter par les divers opérateurs.

Il ne faut pas confondre la séquence selon laquelle les lignes de commandes d'une commande donnée seront collectées, référant au parcours de cueillette, et le séquençement des commandes qui implique l'ordre à laquelle les commandes sont mises en préparation.

Boysen et al (2017a) et Boysen et al (2017b) présentent un modèle de séquençement des commandes avec la présence d'un système d'armoires mobiles. Une heuristique permettant de séquencer les commandes fonctionnant de telle sorte que la dernière armoire visitée pour collecter une commande soit la première armoire à visiter pour la commande suivante à collecter si possible. De plus le choix de la séquence de collecte au sein des armoires mobiles à visiter est aussi une variable de décision, ainsi il y a à la fois séquençement des commandes et le séquençement des lignes de commandes à collecter. Des heuristiques ainsi qu'une résolution exacte sont présentées. Une métaheuristique de type « recuit simulé » est présentée pour le séquençement des commandes puis un algorithme « greedy » est utilisé pour le séquençement des lignes de commandes.

Certains articles traitent le séquençement des commandes utilisant des AS/RS de type Carousel (cf. section suivante). Lee and Kuo (2008), par exemple, font l'hypothèse de collecte d'une seule commande à la fois. L'objectif est de séquencer les commandes ainsi que les lignes de commandes permettant de minimiser la distance parcourue par la machine.

Williams et Fosso (1995) considèrent le séquençement des commandes avec l'utilisation des stockeurs automatisés (ici de type de carousel). Une simulation ainsi qu'une comparaison de systèmes de préparation de commandes, le premier considérant un regroupement de commandes en lots, et le second considérant le « Rolling picking ». Nous nous attardons sur le « rolling picking » car nous considérons un système de préparation de commandes similaire dans la Chapitre dédiée au Rolling batching (Chapitre 5 de ce manuscrit). Ce système est caractérisé par plusieurs commandes simultanément en cours de collecte sur plusieurs stockeurs automatisés. Jusqu'ici, ce système est similaire au regroupement de commandes en lots. La différence provient du fait que lorsqu'une commande se termine, elle est remplacée par une nouvelle sur la table de préparation tandis que dans le cas de batching, il faut attendre que toutes les commandes soient collectées avant que celles-ci ne quittent la zone de préparation. Ce court article ne rentre pas dans les détails et compare les deux systèmes sans vraiment exposer les conditions de chacun des systèmes, ni la modélisation mathématique permettant d'évaluer les performances de chacun d'eux.

Nous remarquons l'absence d'articles se focalisant sur le système de préparation telle que nous l'étudions dans la Section 5 de notre manuscrit. Nous nous inspirons du terme employé par Williams et Fosso (1995) en utilisant le terme « rolling batching » lorsque nous nous référons à un tel système de préparation de commandes.

Au cours de cette section, nous avons défini puis décrit les différentes dimensions impactant sur l'activité de préparation de commandes, ainsi que la littérature abordant ces dimensions. Nous avons focalisé notre attention sur l'activité de regroupement de commandes en lots ainsi que de séquençement des commandes. L'essentiel des travaux décrits était dans un contexte de système « Picker to Part ». La prochaine section est dédiée à l'étude de la littérature concernant la préparation de commandes lorsque la collecte s'effectue sur des stockeurs automatisés.

2.3 Préparation de commandes et stockeurs automatisés

Les entrepôts peuvent avoir des degrés d'automatisation très hétérogènes, autant par la technologie employée (types de machines est utilisées) que par l'étendue que prend cette automatisation au sein de l'entrepôt (le nombre d'articles stockés dans les machines et le nombre de mouvements engendrés par ces machines). Bien que l'entrepôt manuel, dit entrepôt « traditionnel », soit majoritairement présent dans l'environnement industriel, l'automatisation prend de plus en plus d'ampleur ces dernières décennies avec l'apparition de machines dédiées au stockage des références et facilitant la préparation des commandes clients. Cette partie est consacrée à l'étude de la littérature concernant les principaux systèmes de préparation de commandes automatisés. Nous allons recenser et détailler la littérature sur ces différents systèmes pour ensuite nous focaliser sur les systèmes qui nous intéressent, les Carousels et les VLMs. Pour une étude exhaustive de la littérature sur les AS/RS, nous renvoyons le lecteur sur les travaux de Roodbergen et Vis (2009) et Manzini et al. (2005).

2.3.1 Automated Storage and Retrieval Systems (AS/RS)

Les systèmes appelés Automated Storage and Retrieval Systems (AS/RS) (transstockeur en français) ont été largement étudiés depuis leur introduction dans les années 60 et ont grandement évolué (Carlo et Vis (2012)). De par l'utilisation du terme AS/RS plus fréquent dans la littérature, nous continuerons d'employer ce terme tout au long de ce manuscrit. Leur implémentation permet des gains en charges salariales, un gain de place au sol ainsi qu'un taux d'erreur de prélèvement réduit par rapport aux systèmes traditionnels (Zollinger, 2014). Les versions les plus basiques d'AS/RS comprennent deux rangées de racks où sont stockées les références, chacune des références est directement accessible par l'AS/RS. Une illustration graphique est disponible en figure 11 ci-dessous. Les AS/RS se déclinent sous beaucoup de formes et de types en fonction des besoins et dimensionnement des plateformes. Deux revues de littérature particulièrement pertinentes concernant les AS/RS sont données par Roodbergen et Vis (2009) et Tompkins et al. (2010).



Figure 11 : Illustration d'un AS/RS basique

Les prochaines sections sont dédiées à l'étude de la littérature concernant la préparation de commande utilisant des machines automatisées dérivées des AS/RS. Nous nous intéresserons à deux sous classes des AS/RS, nommées Autonomous Vehicle Storage and Retrieval System (AVS/RS) et Shuttle Based Storage and Retrieval System (SBS/RS), puis nous nous focaliserons sur l'étude de la littérature concernant les VLMs et Carousels, principaux sujets d'étude de la présente thèse.

2.3.2 Autonomous Vehicle Storage and Retrieval System (AVS/RS)

De nouveaux concepts d'AS/RS ont vu le jour prenant en compte les spécificités client tel qu'un débit accru ou des contraintes de délai. Les Autonomous Vehicle Storage and Retrieval Systems (AVS/RS) sont constitués d'un véhicule autonome permettant le mouvement horizontal dans les allées sur chaque niveau tandis qu'une plateforme élévatrice permet le mouvement vertical à chaque extrémité des allées (Ekren and Heragu, 2011). Il peut y avoir plusieurs allées permettant un plus faible temps d'attente pour l'opérateur (Cf Figure 12 ci-dessous) (Ekren, 2011). Le véhicule se déplace entre les emplacements de stockage pour récupérer le contenant et la plateforme élévatrice permet de récupérer un contenant en se positionnant à la bonne hauteur et de l'amener à la hauteur du préparateur.

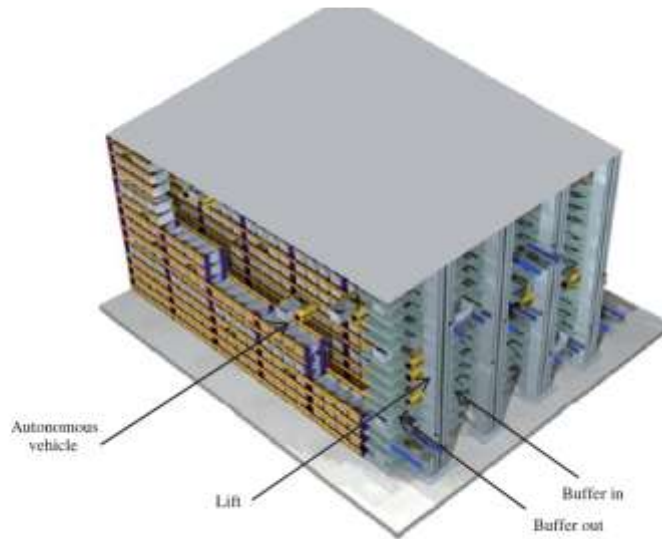


Figure 12 : Illustration d'un AVS/RS

De nombreux travaux ont été consacrés à ces machines et l'étude de leurs performances. Malmborg (2002) développe des modèles analytiques permettant d'évaluer les temps de cycle et les utilisations des véhicules pour stocker ou collecter une liste de références. Kuo et al. (2007) proposent un modèle afin d'estimer l'utilisation des ressources et les coûts d'implémentations d'AVS/RS. 12 scénarii sont évalués en fonction du nombre de véhicules, de plateformes, d'allées et de la distribution de probabilité pour le temps moyen de cycle des véhicules. Kuo et al. (2008) proposent une extension de ce modèle en considérant diverses politiques de stockage basées sur les classes. Marchet et al. (2012) se focalisent sur les temps de cycle ainsi que les temps d'attente. Ils utilisent des réseaux de files d'attente ouverts pour la modélisation et la simulation afin de valider leur approche.

La littérature concernant les AVS/RS se focalise aussi sur la conception de ces machines. Ainsi Fukurani et Malmborg (2008) continuent et étendent le travail de Malmborg (2002). Ils simulent 15 scénarii avec différentes caractéristiques d'AVS/RS. Les configurations optimales d'AVS/RS tendent à avoir des allées plus courtes afin d'optimiser le mouvement des véhicules, d'avoir des racks plus bas pour avoir un mouvement vertical plus rentable et moins de déplacements à vide. Ekren et Heragu (2011) font varier le nombre de plateformes élévatoires et de véhicules pour étudier les performances du système en termes de temps de cycle en utilisant une analyse basée sur la régression afin d'identifier les différents facteurs permettant d'impacter les performances des AVS/RS. Marchet et al. (2013) s'intéressent à un compromis entre les coûts de mise en place et les performances pour de nouveaux modèles d'AVS/RS. Ils considèrent différents scénarii et

simulent le coût de ces installations. Pour chacun de ses scénarii, les performances en termes de temps de cycle et de temps d'attente sont aussi étudiées.

2.3.3 Shuttle Based Storage and Retrieval Systems (SBS/RS)

Les SBS/RS sont relativement nouveaux et sont généralement utilisés pour le stockage de bacs (Malmberg, 2002). Ce type d'AS/RS est très proche des AVS/RS. La différence provient de l'équipement utilisé pour le mouvement horizontal. Ce sont des nacelles mobiles qui l'effectuent (Lehrer et al., 2016) alors que pour les AVS/RS, ce sont des véhicules autonomes. Les plateformes élévatrices sont utilisées, tout comme pour les AVS/RS (Cf. Figure 13). Lehrer et al. (2014) et Lehrer (2013) étudient la consommation énergétique ainsi que les réductions d'émission de CO₂ de ce système suivant divers scénarii dans le cas de Cycle à Commande Double (CCD). Le CCD implique que la machine peut effectuer à la fois un stockage ainsi qu'une collecte par cycle. Il se différencie du Cycle à Commande Simple (CCS) où simplement un stockage ou bien une collecte peut être fait par cycle. Un cycle se définit par le fait que la machine, en partant du point de départ, effectue une action dans le cas CCS ou deux actions dans le cas CCD avant de revenir à ce point de départ (le point de départ généralement au niveau de l'opérateur chargé de la préparation de commande). Carlos et Vis (2012) se focalisent sur les aspects d'ordonnancement et évaluent le débit (i.e. nombre de commandes collectées ou le nombre de stockages effectués par unité de temps). Dans ce papier, un type de SBS/RS utilisant deux plateformes est utilisé. Lehrer et al (2015) et Lehrer (2016) définissent des modèles analytiques afin de calculer les temps de cycle. Ainsi ces temps sont divisés entre les divers temps d'attente de chacun des composants plus le temps de collecte et de mise en stock. Le temps moyen pour un fonctionnement en CCS ainsi qu'en CCD est défini en fonction des caractéristiques de la machine (nombre d'allées, longueur des espaces de stockage et nombre de nacelles). Lehrer et al (2016) évaluent le débit dans les deux différents cas (CCS et CCD), cependant, aucune étude numérique ne permet de valider leur modèle.

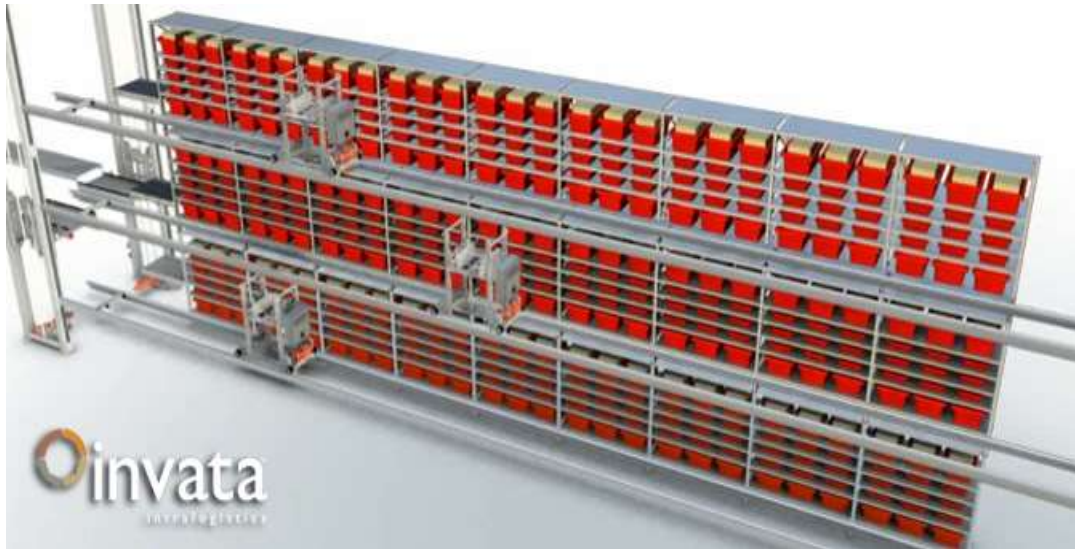


Figure 13 : Illustration de SBS/RS

2.3.4 Le Vertical Lift Module (VLM)

Le Vertical Lift Module (VLM) ayant pour traduction « Module d'Élévation Verticale » est une variante d'AS/RS plus économique ainsi que moins encombrante que les solutions présentées ci-dessus. Dans le cadre de notre étude nous utiliserons le terme VLM pour se référer à ce système. Les produits sont stockés sur des plateaux positionnés à l'avant ou à l'arrière de la machine (cf. Figure 14 ci-après). Une fois un plateau appelé, une plateforme élévatrice se déplace verticalement entre les deux rangées verticales des plateaux. Une fois la hauteur où est situé le plateau atteint, la plateforme extrait le plateau correspondant pour ensuite revenir au point d'ouverture de la machine et le plateau est alors disponible pour que le préparateur effectue la collecte. La vitesse verticale de la plateforme est bien plus grande que la vitesse de déplacement horizontale. Ainsi le temps pour extraire un plateau ne change pas significativement d'un plateau à l'autre.

Les VLMs s'utilisent généralement en groupe de plusieurs machines permettant d'effectuer des entrées ou sorties d'un plateau d'un VLM donné pendant la collecte d'articles sur un autre VLM. Donc le temps de recherche de plateau est réduit car il est fait en temps dit « masqué ». Nous notons aussi que les plateaux présents au sein d'un VLM n'ont pas d'emplacement précis au sein du stockeur. Ainsi le plateau n°1 qui serait situé à une certaine hauteur du stockeur avant son extraction, pourrait être placé à une autre hauteur au sein du stockeur lors de son rangement. En effet, chaque fois qu'un plateau doit être rangé, celui-ci a la hauteur de la référence la plus haute calculée, donnant la hauteur maximale prise par ce plateau. Un système interne au VLM place le plateau automatiquement afin de pouvoir optimiser la place disponible au sein du VLM en

fonction de la hauteur des produits présents sur le plateau. Le numéro du plateau ne définira pas son emplacement au sein de la machine. C'est une des raisons de la forte utilisation de ce type de machine en pratique par les entrepôts dont l'objectif principal est de gagner de la place au sein de l'entrepôt.

Bien que ce type de machine soit de plus en plus utilisé dans les entrepôts faisant le choix de s'automatiser, la littérature concernant ce type de machine reste rare. Nous pouvons cependant noter les travaux de Meller et Klote (2004) qui analysent les temps de cycle avec l'utilisation d'un ou plusieurs VLMs. Une définition analytique des temps de cycle est faite en utilisant la théorie des files d'attente. Afin de déterminer les temps de cycle ainsi que le taux d'utilisation de l'opérateur, le temps de picking ou temps pour collecter une ligne de commandes est considéré déterministe ou ayant une distribution exponentielle. Dukic et al. (2015) étendent le travail de Meller et Klote (2004) en étudiant un type de VLM très particulier. Ce VLM dit « à double ouverture » permet d'avoir, au sein d'une même machine, deux plateformes élévatoires permettant à un plateau d'être à disposition du préparateur pendant qu'un second plateau est en attente et donc disponible aussitôt que la collecte est faite sur le premier. Cette technologie s'éloigne de notre sujet d'étude mais nous pouvons noter que la méthodologie de calcul est la même que celle utilisée par Meller et Klote (2004). Plus récemment Rosi et al. (2016) étudient les divers avantages de l'utilisation des VLMs en termes de temps de cycle moyen ainsi que de débit. Plusieurs scénarii prenant en compte diverses configurations de VLMs sont examinés avec des temps de picking déterministes.

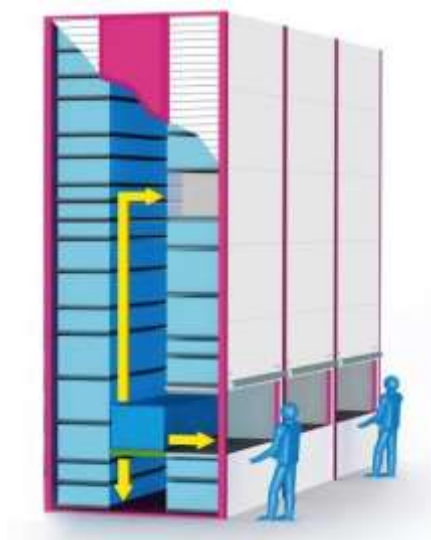


Figure 14 : Illustration VLM

2.3.5 Le Carousel

Cette section se consacre à la littérature concernant les systèmes de préparation de commandes de type carousel. La différence entre un système de type VLM et un système de type carousel est la manière de présenter un plateau à un préparateur de commandes situé en face de ce type de système. Les plateaux à l'intérieur du carousel sont reliés par une chaîne effectuant des rotations autour d'un des axes. Ainsi le temps de présentation d'un plateau va dépendre de la position relative de ce plateau par rapport au plateau actuellement présenté au préparateur de commandes. Dans le cas des carousels, la séquence selon laquelle les plateaux sont visités est donc extrêmement importante, en comparaison avec le VLM où les temps pour chercher les plateaux sont quasiment constants quel que soit le plateau visité. Il existe deux types de carousel en fonction de l'axe autour duquel s'effectue la rotation de la chaîne. Le carousel vertical a ses plateaux tournant autour de l'axe horizontal tandis que le carousel horizontal a ses plateaux tournant autour de l'axe vertical. Une illustration du carousel vertical est donnée dans la figure 15.



Figure 15 : Illustration Carousel

Une revue exhaustive de la littérature a été effectuée dans Litvak et Vlasiou (2010) recensant les papiers traitant la préparation de commandes associés au système des carousels. Au vu de l'abondance des papiers traitant les carousels, nous discuterons des traitant la préparation de commandes. Nous excluons les papiers traitant le stockage des références au sein des carousels ainsi que ceux traitant la conception des carousels ou des zones comprenant les carousels. Pour ces papiers, on peut se référer à la revue de littérature de Litvak et Vlasiou (2010). Tout d'abord, différentes stratégies de picking des références sont considérées dans la littérature. Ces stratégies

visent à définir l'ordre de collecte des produits au sein du carousel en définissant la séquence de plateaux à visiter. Dans le cas de collecte d'une seule commande, Litvak et Adan (2001) développent la stratégie « nearest item ». Cette heuristique vise à se déplacer jusqu'au plateau le plus proche du plateau actuel ayant un produit à collecter.

Litvak et Adan (2002) utilisent une autre heuristique appelée « m-step strategy ». Suivant cette stratégie, le carousel prend la route la plus courte entre celles changeant de direction après au maximum « m » produits collectés. Ainsi il y aura un seul changement de sens et $2*m+2$ distances seront calculées. Si $m=0$, 2 distances seront donc calculées, l'une collectant tous les produits en effectuant la rotation dans un sens, l'autre distance calculée si le carousel effectue sa rotation dans l'autre sens. Le résultat de la 0-step stratégie est la plus faible distance entre les deux choix. La 1-step strategy va calculer deux distances supplémentaires, une pour chaque sens de rotation et prendre le meilleur résultat de la 0-step strategy et des deux nouveaux résultats. La nouvelle distance sera celle où le carousel tourne dans un sens et change de sens de rotation après avoir collecté m produits, soit un produit dans le cas 1-step. La même procédure est calculée pour tout entier m. D'après l'auteur, cette dernière stratégie semble permettre un temps de préparation de commandes meilleur que l'heuristique présenté par Litvak et Adan (2001).

Lorsque plusieurs commandes doivent être collectées, Lee and Kuo (2008) proposent un algorithme pour trouver la séquence de collecte des références pour chaque commande et la séquence des commandes minimisant le coût global de picking. Dans ce cas, le coût de picking dépend de la distance que le carousel doit parcourir plus le coût du préparateur de commandes qui doit parcourir la distance entre le carousel et les bacs. Le temps de picking est supposé constant par ligne de commandes.

Le seul article que nous ayons trouvé sur la préparation de commandes avec des carousels et un regroupement dit en lots est celui de Bartholdi and Platzman (1986). Dans cet article, les commandes sont regroupées en lots ; mais à l'intérieur d'un lot donné les articles au sein d'une même commande doivent être collectés consécutivement et donc les commandes sont collectées une par une. Le séquençement des lots y est abordé et plusieurs heuristiques sont développées pour permettre ce séquençement. Mais à nouveau le fait que les commandes d'un lot donné doivent être collectées les unes après les autres différencie donc grandement cet article de notre travail

2.4 Résumé de la littérature et intérêt de notre travail

Au cours de ce chapitre, nous avons abordé les diverses thématiques permettant de mieux comprendre les facteurs impactant notre sujet et de le positionner. Ainsi la préparation de commandes est impactée par une grande diversité de « dimensions ». Shouman et al (2007) résumant la succession de décisions à prendre afin d'étudier la performance d'un système de préparation de commandes :



Figure 16 : Succession des décisions à prendre en compte afin d'aborder la préparation de commandes (inspiré de Shouman et al (2007))

Les différentes dimensions impactant la préparation de commandes, se traduisent, dans le cas de stockeurs automatisés, comme suit. L'agencement de l'entrepôt fera référence au nombre de plateaux dans les machines ainsi qu'au nombre de machines. Le zonage déterminera les produits présents au sein des stockeurs. La politique d'affectation des produits aux emplacements définit un placement des produits sur les différents plateaux. Le parcours de cueillette décrit dans quel ordre les plateaux seront visités pour collecter les références.

Notre objectif est un regroupement ou un séquençement des commandes (en fonction du type de préparation de commandes souhaité par la plateforme) permettant la minimisation du temps de préparation. Nous cherchons donc, à travers cette thèse, à déterminer les lots de commandes (i.e. attribuer à chacune des commandes un numéro de lot) si un regroupement des commandes en lots complets est nécessaire ou un séquençement de ces commandes si la préparation de commandes est en Rolling batching. Nous prendrons les autres dimensions comme des contraintes. Ainsi le nombre de machines et nombre de plateaux par machines, ainsi que la disposition de ces dernières au sein des machines seront des données d'entrée venant des entrepôts clients de notre partenaire industriel KLS Logistic. Notre objectif est donc de pouvoir, en fonction de ces données, avoir un temps de collecte optimal grâce à un regroupement des commandes en lots ou bien un séquençement des commandes.

Notre travail est motivé par l'absence de papiers dans la littérature concernant les modèles sur le regroupement en lots des commandes dans le cas de systèmes « Part to Picker » utilisant les stockeurs automatisés aussi répandus que les VLMs et Carousels. Ainsi le Chapitre 3 est dédié au

batching ou regroupement de commandes en lots complets lorsque le système de préparation de commandes est équipé de machine de type VLM tandis que le chapitre 4 se focalise sur le même problème lorsque le système de préparation de commandes est équipé de Carousels. De plus nous avons évoqué l'absence de littérature permettant un séquençement des commandes dans un système de type « rolling batching » mis à part l'article de Williams et Fosso (1995) (cf. section 2.2.4.2). Ce type de système permet un grand débit de commandes collectées ainsi qu'une organisation adaptée à l'e-commerce. Le chapitre 5 se focalise sur le séquençement des commandes dans un système adapté au e-commerce, que nous nommons « système drive ».

Enfin, de par l'étude de la littérature et l'étude de divers entrepôts de distribution, nous observons que la plupart des entreprises utilisent des procédures basiques afin de réaliser un séquençement ou un regroupement en lots des commandes (en utilisant, par exemple le FIFO comme procédure). Ces procédures, bien que rapides d'implémentation, donnent cependant des résultats souvent loin du résultat optimal. Pouvoir créer des modèles d'optimisation permettant un regroupement de commandes ainsi qu'un séquençement des commandes optimisés et compréhensible par les entreprises fut une motivation supplémentaire pendant la réalisation de cette thèse.

Chapitre 3 Regroupement de commandes en lots dans une zone de préparation équipée de VLMs

Dans ce chapitre, nous allons nous focaliser sur l'activité de regroupement de commandes en lots avec l'utilisation de stockeurs du type VLM. Ainsi, la première section décrira le fonctionnement du procédé de préparation de commandes lorsque celles-ci sont regroupées en lots et que la collecte doit s'effectuer sur un ou plusieurs stockeurs de type VLM. La section 2 sera dédiée à un cas spécifique, lorsque la préparation se fait sur une seule machine, tandis que la section 3 se focalisera sur le cas d'une préparation s'effectuant sur plusieurs VLMs. Pour chacune de ces deux sections, nous définirons les temps de préparation de commandes, nous permettant d'établir un modèle d'optimisation que nous testerons sur des études de cas issues de données réelles d'entreprise. Enfin, la section 4 présentera une métaheuristique pour fournir une solution dans le cas où la résolution exacte ne peut pas prouver l'optimalité des résultats en un temps de calcul raisonnable. Les travaux étudiés dans ce chapitre ont été présentés dans Lenoble et al. (2016), Laajili et al. (2017) et Lenoble et al. (2017b).

3.1 Procédé de préparation de commandes sur des stockeurs automatisés de type VLM avec regroupement en lots

Tout au long de cette partie, nous nous focalisons sur l'activité de regroupement de commandes en lots que nous cherchons à optimiser. En d'autres termes, quelles sont les commandes que l'opérateur va collecter simultanément pour avoir un temps de préparation de commandes minimal. Ainsi nous décrivons comment se fait la préparation de commandes dans un tel cas, en expliquant les spécificités du stockeur automatisé de type VLM.

3.1.1 Spécificité du Vertical Lift Module

Le Vertical Lift Module ou VLM est une machine composée de plateaux où sont rangés les références stockées (cf. figure 17 ci-après). Une fois un produit demandé pour la préparation de commandes, la machine va chercher le plateau sur lequel est stocké ce produit. L'extraction du plateau se déroule en plusieurs phases, tout d'abord la plateforme élévatrice se met à hauteur du plateau. Cette opération verticale est très rapide. Une fois la plateforme à bonne hauteur, un système de préhension composé de crochets agrippe le plateau et le tire horizontalement jusqu'à ce que le plateau entier soit sur la plateforme. S'en suit une nouvelle opération verticale afin d'amener le plateau au niveau de l'opérateur en le positionnant face à l'ouverture de la plateforme. Enfin la plateforme élévatrice pousse horizontalement le plateau afin que celui-ci permette à l'opérateur d'effectuer la collecte. Une fois la collecte finie sur le plateau, il est rangé et le plateau suivant est extrait en utilisant le même procédé.

Afin que les produits stockés au sein du plateau ne subissent aucun dommage ou ne soient pas renversés du plateau, les opérations horizontales sont bien plus lentes que celles verticales. En effet les chiffres donnés par un constructeur de VLM, l'entreprise Kardex Remstar, sont les suivants : Vitesse Verticale : jusqu'à 2 m.s^{-1} ; Vitesse Horizontale : jusqu'à un maximum de 0.7 m.s^{-1} . De plus, l'accélération horizontale est plus faible que l'accélération verticale, afin d'assurer la sécurité des produits. Ainsi donc, même si la distance horizontale à parcourir par les plateaux est faible, sachant que l'accélération horizontale l'est aussi, le temps pour atteindre la vitesse maximale sera long. Le temps de déplacement horizontal est donc bien plus élevé que le temps de déplacement vertical. Enfin, nous confirmons cette hypothèse par les dires de notre partenaire industriel nous affirmant que les différences de temps d'extraction ne sont pas notables entre les plateaux. Nous avons pu confirmer cette affirmation dans la pratique en chronométrant les différents temps d'extraction de plateaux au sein d'un entrepôt, et nous n'avons pas noté de différences significatives entre les temps pour extraire différents plateaux.

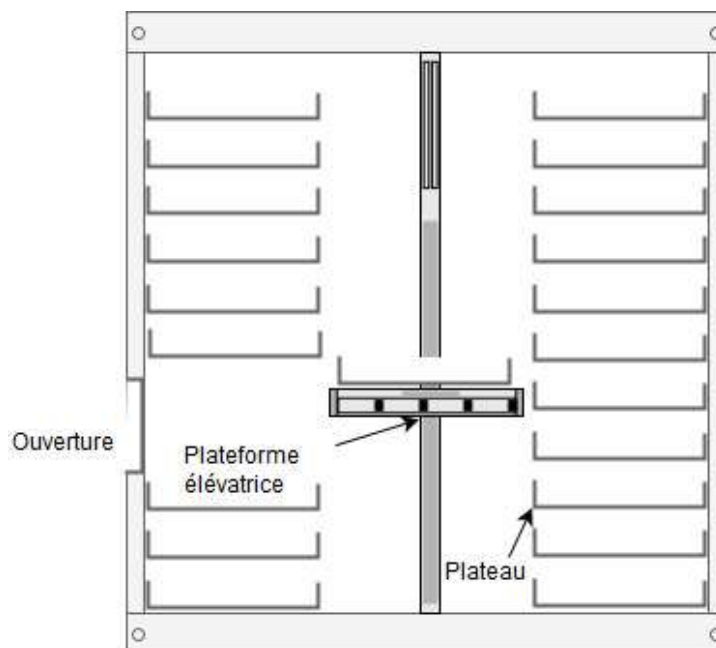


Figure 17 : Illustration d'un VLM (vue de côté)

3.1.2 Procédé de picking considérant un regroupement de commandes sur un ou plusieurs VLMs

Nous considérons donc un ou plusieurs VLMs sur lesquels doivent s'effectuer l'opération de préparation de commandes. De plus, un seul préparateur de commandes est devant un ou plusieurs VLMs pour effectuer la collecte.

Nous rappelons qu'une commande est composée de lignes de commandes, chacune correspondant à une quantité à collecter d'une référence donnée. Chacune des références est stockée sur un plateau du VLM. Lorsque plusieurs commandes clients sont reçues et doivent être préparées, la première décision est de regrouper ces commandes en lots. Ainsi à partir du portefeuille de commandes comprenant l'ensemble des commandes à regrouper, plusieurs lots de commandes sont constitués, chaque lot étant un sous-ensemble du portefeuille. Un lot est donc composé d'un ensemble de lignes de commandes à collecter. En pratique nous notons qu'une commande n'est pas scindée sur plusieurs lots. En d'autres termes, toutes les lignes constituant une commande sont dans un même lot. Une fois la décision faite concernant l'attribution des commandes à chacun des lots, ces différents lots de commandes doivent être collectés. Le procédé de collecte sur les VLMs se déroule de la façon suivante :

Pour chacun des lots, le préparateur de commandes reçoit un certain nombre de bacs. Ce nombre est égal au nombre de commandes dans le lot à collecter. Dans la pratique, une commande est

placée dans un seul bac (voir Figure 18 ci-dessous). Chaque fois que le préparateur collecte une ligne de commande sur un plateau (nous rappelons qu'une ligne est une quantité donnée d'une référence donnée), il place cette quantité dans le bac associé à la commande.



Figure 18 : Préparation de commandes avec regroupement en lots et utilisation de VLMs

Le préparateur de commandes commence donc par la collecte des lignes de commandes positionnées sur le premier plateau du premier VLM devant être visité. La collecte consiste à retirer toutes les différentes lignes de commandes de ce plateau et à les mettre dans les bacs correspondants. Une fois cela fait, le plateau de ce VLM est rangé et le VLM extrait le plateau suivant devant être visité. Pendant ce temps, le préparateur marche jusqu'au VLM suivant et récolte les lignes à collecter sur le premier plateau à visiter par le second VLM. L'opérateur chargé de la préparation de commandes continue jusqu'à ce que les lignes de commandes soient collectées sur le premier plateau devant être visité sur chacun des VLMs, en d'autres termes, jusqu'à ce que tous les VLMs contenant des références devant être collectées soient visités une fois. Le préparateur de commandes revient ensuite au premier VLM pour collecter les références sur le second plateau du premier VLM. Il répète le procédé décrit ci-dessus pour la seconde tournée de collecte. Ce procédé continue jusqu'à ce que tous les produits soient collectés. Ces bacs quittent ensuite la zone de picking et de nouveaux bacs correspondant au prochain lot arrivent. Le préparateur de commandes répète le procédé décrit ci-dessus jusqu'à ce que tous les lots soient collectés.

Ainsi le temps de préparation de commandes est divisé en deux temps, le temps de collecte des produits auquel s'ajoute d'éventuels temps d'attente devant les machines. Nous négligeons le temps de marche du préparateur pour aller d'un VLM à un autre car celui-ci est, en pratique,

extrêmement faible au vu de la proximité des machines. Le procédé de préparation de commandes peut être décrit comme ci-dessous (Fig. 19) avec V VLMs présents dans la zone de picking et P_l tournées de plateaux à visiter sur le lot l . Nous notons que l'étape de collecte peut être nulle sur un plateau d'un VLM, si la p -ième tournée de plateaux actuellement visité excède le nombre de plateaux devant être vus par ce VLM pour un lot donné.

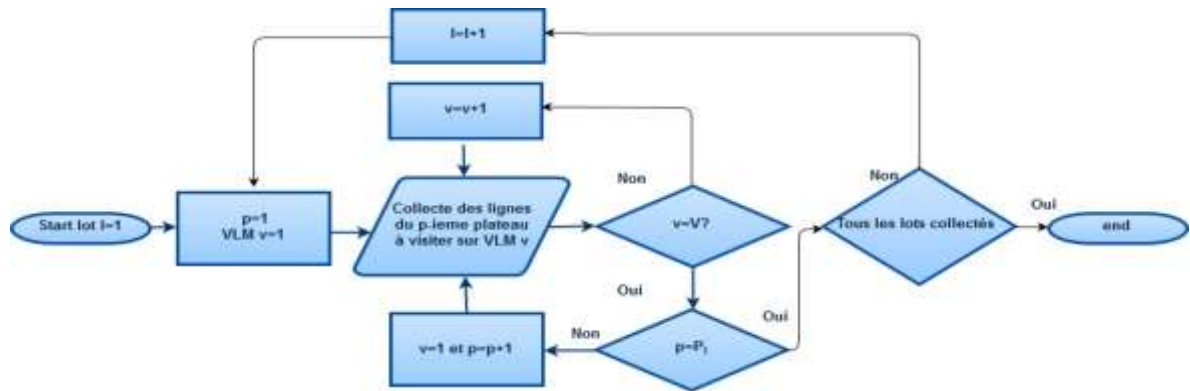


Figure 19 : Procédure de préparation de commandes regroupées en lots sur V VLMs

3.2 Regroupement de commandes en lots dans un système de préparation de commandes utilisant 1 VLM

Dans cette section nous considérons un préparateur devant effectuer la préparation sur 1 VLM. Notre objectif est la création de lots en minimisant le temps de préparation de commandes, temps pris par le préparateur pour collecter toutes les commandes du portefeuille. Selon nos partenaires industriels, minimiser le temps de préparation de commandes est l'un des principaux objectifs des entreprises utilisant des VLMs. Dans le reste de cette section, nous allons tout d'abord modéliser le temps de préparation de commandes. Ainsi nous présentons les notations et les hypothèses adoptées. Puis, nous évaluons le temps de préparation de commandes pour un portefeuille donné. Par la suite nous développons notre modèle d'optimisation permettant un regroupement de commandes en lots. En se basant sur des données réelles provenant de deux sociétés, nous avons testé notre modèle et comparé ses performances avec les méthodes de regroupement en lots utilisées dans les entreprises considérées.

3.2.1 Notation et hypothèses

Nous considérons les notations suivantes :

- N : le nombre de commandes clients.
- $\mathbb{N} = \{1, \dots, N\}$: l'ensemble des indices des commandes clients à regrouper, indexé par n .
- S : la taille maximale des lots (i.e. le nombre maximum de commandes pouvant être attribué à un lot). Dans la pratique ce nombre est constant et dépend de la place disponible pour accueillir les bacs en face du préparateur de commandes.
- B : Nombre de lots de commandes à créer. En considérant le nombre de commandes et la taille de lots, nous calculons $B = \left\lceil \frac{N}{S} \right\rceil$ avec $\lceil \cdot \rceil$ la fonction borne supérieure.
- $\mathcal{B} = \{1, \dots, B\}$: ensemble des indices des lots de commandes indexé par b .
- K : Nombre total de plateaux présents au sein du VLM.
- $\mathcal{K} = \{1, \dots, K\}$: Ensemble des numéros de plateaux du VLM.
- K^b : Nombre de plateaux devant être visités au cours du lot b .
- W : Temps total d'attente du préparateur (Somme de tous les temps où le préparateur est disponible pour la collecte mais le plateau sur lequel il doit collecter ne l'est pas).

- P : Temps total de collecte, correspondant au temps durant lequel le préparateur collecte les références sur les plateaux et les pose dans les bacs.
- C : Temps total de préparation de commandes, $C = W + P$.
- W^b : Temps d'attente associé au lot b .
- P^b : Temps de collecte associé au lot b .
- C^b : Temps de préparation de commandes associé au lot b : $C^b = W^b + P^b$.
- W_l^b : Temps d'attente de l'opérateur devant le l -ième plateau devant être visité pour le lot b
- P_l^b : Temps de collecte des références situées sur le l -ième plateau à visiter du lot b
- C_l^b : Temps de préparation de commandes pour le l -ième plateau à visiter du lot b , $C_l^b = W_l^b + P_l^b$.
- L_l^b : Nombre de lignes de commandes à collecter sur le l -ième plateau à visiter par le lot b
- p : Temps de collecte d'une ligne de commandes sur un plateau quand celui-ci est disponible. Ce temps comprend la collecte sur le plateau ainsi que la dépose de la quantité demandée d'une référence dans un bac. Ce temps de collecte est aussi appelé temps de picking. Nous faisons l'hypothèse que ce temps de collecte d'une ligne de commande est constant, c'est-à-dire que quelle que soit la référence à prélever, le temps de collecte sera le même. Dans la pratique, ce paramètre dépend de nombreux facteurs tels que la quantité demandée de la référence ainsi que des opérations additionnelles éventuelles (par exemple le scan, la pesée ou la mise sous sachet de la référence). Ces facteurs intrinsèques à chaque entrepôt sont difficiles à estimer dans la pratique. D'après notre partenaire industriel KLS Logistic, la variation de la valeur de p entre différentes lignes de commandes au sein d'un même entrepôt n'est pas significative pour la plupart de leurs clients équipés de VLMs, motivant notre choix de considérer p constant.
- e : Temps requis pour que le VLM range le plateau actuel et extraie le plateau suivant sur lequel la collecte doit se poursuivre. Nous considérons que la valeur de e est la même pour chacun des plateaux.

Afin de faciliter la présentation de ce problème, nous allons tout d'abord étudier le cas où le temps de collecte d'une ligne de commande est négligeable par rapport au temps de rangement d'un plateau et d'extraction du plateau suivant. Nous étudierons par la suite le cas réaliste où le temps de collecte sur le l -ième plateau à visiter dépend du nombre de lignes de commandes à collecter sur ce plateau, donnant la relation suivante : $P_l^b = L_l^b * p \forall b \in \mathcal{B}, \forall l \in \{1, \dots, K^b\}$.

Nous faisons l'hypothèse, tout au long de cette partie, que la collecte d'un lot commence par un temps d'attente, afin que le VLM range le plateau actuellement disponible et aille chercher le premier plateau devant être visité.

Comme nous l'avons dit précédemment, le temps de préparation d'un portefeuille de commandes donné est la somme des temps d'attente où le préparateur patiente jusqu'à ce que le VLM devienne disponible pour la collecte, et des temps de collecte où l'opérateur récupère les références des plateaux pour les mettre dans les bacs.

Nous avons les relations suivantes :

$$C = \sum_{b=1}^B C^b = \sum_{b=1}^B \sum_{l=1}^{K^b} C_l^b \quad (3-1)$$

$$W = \sum_{b=1}^B W^b = \sum_{b=1}^B \sum_{l=1}^{K^b} W_l^b \quad (3-2)$$

$$P = \sum_{b=1}^B P^b = \sum_{b=1}^B \sum_{l=1}^{K^b} P_l^b \quad (3-3)$$

En rassemblant (3-1), (3-2) et (3-3) ainsi que le fait que le temps de préparation de commandes est constitué de temps d'attente et de picking, nous obtenons la relation suivante :

$$C = W + P = \sum_{b=1}^B W^b + P^b = \sum_{b=1}^B \sum_{l=1}^{K^b} (P_l^b + W_l^b) \quad (3-4)$$

3.2.2 Evaluation du temps de préparation de commandes sur 1 VLM

Nous utilisons les notations ainsi que les hypothèses faites dans la section précédente afin d'évaluer le temps de préparation de commandes. Dans un but pédagogique permettant au lecteur de se familiariser avec les notations et notre méthodologie de calcul, nous nous intéresserons tout d'abord au temps de préparation en considérant le temps de picking négligeable par rapport au temps que le VLM met pour changer de plateau. Par la suite, nous étudions le cas plus complexe et plus réaliste où le temps de collecte sur un plateau est proportionnel au nombre de lignes à collecter sur ce même plateau. Nous notons que la situation avec temps de picking négligeable est un cas particulier de la situation où le temps de picking est proportionnel au nombre de lignes à collecter.

Nous allons évaluer le temps de préparation d'un portefeuille de commandes regroupées en lots, ce qui nous permettra de définir un critère d'optimisation pour notre problème de regroupement de commandes. Pour chacun des cas concernant le temps de collecte, nous présentons un lemme suivi de sa preuve avant d'illustrer le cas par un exemple.

Lemme 1 : Dans le cas du temps de temps de collecte négligeable, le temps de préparation de commandes est donné par : $C = e * \sum_{b=1}^B K^b$.

Nous présentons un exemple avant de démontrer le lemme 1.

Exemple 1. Temps de préparation de commandes considérant un temps de picking négligeable

Nous considérons dans l'exemple présenté dans la figure 20 ci-après un lot b visitant 3 plateaux sur le VLM. Sur cette figure et à travers le prochain paragraphe, nous expliquons comment le temps de préparation de commandes est calculé.

Le préparateur de commandes commence par attendre en face du VLM jusqu'à ce que le premier plateau soit disponible prenant e unités de temps. Lorsque ce plateau devient disponible, le préparateur collecte les références, mais ce temps est considéré ici négligeable. Ensuite, le premier plateau est rangé et l'opérateur doit attendre jusqu'à ce que le second plateau soit disponible pour la collecte, prenant à nouveau e unités de temps. S'en suit une activité de collecte, dont nous négligeons le temps, avant que le VLM range ce second plateau à visiter et extraie le troisième plateau. L'opérateur attend de nouveau e unités de temps devant le VLM puis effectue la collecte des références, dont le temps est négligé ici. Le temps de préparation de commandes pour ce lot b est donc : $C^b = W^b = \sum_{l=1}^{K^b} W_l^b = 3 * e$.

e	e	e
W_1^b	W_2^b	W_3^b
C_1^b	C_2^b	C_3^b
C^b		

Figure 20 : Illustration du temps de préparation de commandes lorsque le temps de collecte est considéré négligeable

*Preuve du lemme 1 : le cas de temps de collecte étant supposé négligeable, $P_l^b = 0 \forall l$ et, par conséquent, $P^b = 0$. Ainsi, $C^b = W^b = \sum_{l=1}^{K^b} W_l^b = K^b * e$ car le temps de recherche d'un nouveau plateau est le même quel que soit le plateau. Le temps de préparation de commandes est donc donné par $C = \sum_{b=1}^B K^b * e$, donnant la relation suivante $C = e * \sum_{b=1}^B K^b$. ■*

Lemme 2 : Dans le cas du temps de collecte sur un plateau proportionnel au nombre de lignes de commande à collecter sur ce plateau, le temps de préparation de commandes est donné par :

$$C = e * \sum_{b=1}^B K^b + p * \sum_{b=1}^B \sum_{l=1}^{K^b} L_l^b.$$

Exemple 2. Temps de préparation de commandes considérant un temps de collecte sur un plateau proportionnel au nombre de lignes à collecter sur ce plateau

Nous considérons un cas plus réaliste où le temps de collecte sur un plateau est maintenant proportionnel au nombre de lignes (i.e. nombre de références) ayant besoin d'être collectées sur ce même plateau. Nous considérons un lot b ayant 3 plateaux à visiter. Le nombre de lignes de commande à collecter est : $L_1^b = 2$, $L_2^b = 1$, et $L_3^b = 3$. Nous rappelons que L_l^b est le nombre de lignes de commandes ayant besoin d'être collectées sur le l -ième plateau à visiter pour le lot b . La figure 21 ci-dessous illustre les temps pris en compte dans le calcul du temps de préparation de commandes que nous décrivons dans le paragraphe suivant.

Le préparateur commence par attendre en face du VLM jusqu'à ce que le premier plateau soit disponible, prenant e unités de temps. Une fois ce plateau disponible, il collecte les deux lignes de commandes prenant $P_1^b = 2 * p$ unités de temps. Après cela, le premier plateau est rangé et le nouveau est présenté pour la collecte, durant e unités de temps. L'opérateur y collecte la ligne de commande (prenant $P_2^b = p$). L'opérateur patiente jusqu'à ce que le dernier plateau soit présenté devant lui (prenant à nouveau e unités de temps) avant de collecter les 3 lignes de commandes sur ce plateau (prenant $P_3^b = 3 * p$). Le temps de préparation de commandes de ce lot b est donc $C^b = W^b + P^b = \sum_{l=1}^{K^b} (P_l^b + W_l^b) = e + 2 * p + e + p + e + 3 * p = 3 * e + 6 * p$.

e	p	p	e	p	e	p	p	p	
W_1^b		P_1^b		W_2^b		P_2^b		W_3^b	
C_1^b		C_2^b		C_3^b		C^b			

Figure 21 : Illustration du temps de préparation de commandes lorsque le temps de picking sur un plateau est considéré proportionnel au nombre de lignes à collecter sur ce même plateau

Preuve du lemme 2 : Nous rappelons tout d'abord que le temps de préparation de commandes est défini par $C = \sum_{b=1}^B C^b$ et $C^b = W^b + P^b$.

*Dans le cas du temps de collecte par plateau proportionnel au nombre de lignes de commandes à collecter sur ce même plateau, nous avons $C^b = W^b + P^b = \sum_{l=1}^{K^b} (P_l^b + W_l^b) = K^b * e + \sum_{l=1}^{K^b} L_l^b * p$.*

*Ainsi le temps de préparation pour l'ensemble des lots est donné par $C = \sum_{b=1}^B (K^b * e + \sum_{l=1}^{K^b} L_l^b * p) = e * \sum_{b=1}^B K^b + p * \sum_{b=1}^B \sum_{l=1}^{K^b} L_l^b$ sachant que e et p sont constants. ■*

Proposition 1 : Dans un système de préparation de commandes équipé d'un VLM, quel que soit l'hypothèse prise en compte concernant le temps de collecte (négligeable par rapport au temps de changement de plateaux, ou proportionnel au nombre de lignes à collecter), minimiser le temps de préparation d'un ensemble de commandes est équivalent à minimiser le nombre de plateaux visités pour collecter les différents lots.

Preuve de la proposition 1

Lorsque le temps de collecte est considéré négligeable, nous utilisons la formule énoncée dans le lemme 1 : $C = e * \sum_{b=1}^B K^b$. Nous observons donc que minimiser le temps de préparation de commandes C est équivalent à minimiser le nombre de plateaux visités sur l'ensemble des lots à collecter ($\sum_{b=1}^B K^b$).

Lorsque le temps de collecte sur un plateau est proportionnel au nombre de ligne de commandes à collecter sur ce plateau, nous utilisons la formule énoncée dans le lemme 2 : $C = e * \sum_{b=1}^B K^b + p * \sum_{b=1}^B \sum_{l=1}^{K^b} L_l^b$. C'est important de noter que le nombre total de lignes de commandes à collecter sur l'ensemble des lots, $\sum_{b=1}^B \sum_{l=1}^{K^b} L_l^b$, est une valeur constante ne dépendant pas du regroupement en lots choisi. Ainsi dans le cas du temps de collecte par plateau proportionnel au nombre de lignes de commandes à collecter, minimiser le temps de préparation de commandes est équivalent à minimiser le nombre de plateaux visités sur l'ensemble des lots ($\sum_{b=1}^B K^b$). ■

3.2.3 Modèle d'optimisation et étude de cas

Dans cette section, nous développons le modèle d'optimisation permettant un regroupement de commandes en lots lorsque la collecte s'effectue sur un VLM. Nous présentons par la suite et discutons les résultats des études de cas, permettant de valider notre modèle et d'évaluer ses performances.

3.2.3.1 Modèle d'optimisation

La principale décision de notre modèle est la construction de différents lots à partir d'un portefeuille de commandes donné. Les variables de décision sont :

- $x_{n b} = 1$ si la commande n est incluse dans le lot b ; 0 sinon.
- $T_{k b} = 1$ si le plateau k est visité durant le lot b ; 0 sinon.

Nous utilisons aussi la variable U_{kn} , déduite des données du problème ; U_{kn} est égale à 1 quand la commande n contient une référence située sur le plateau k ; 0 sinon.

Nous avons démontré que minimiser le temps de préparation de commandes revient à minimiser le nombre total de plateaux visités sur l'ensemble des lots ($\sum_{b=1}^B K^b$). Il est important de noter que $K^b = \sum_{k=1}^K T_{kb}$

Ainsi, la fonction objectif de notre modèle d'optimisation prenant en compte 1 VLM est donnée par

$$\text{Minimiser } \sum_{b=1}^B \sum_{k=1}^K T_{kb} \quad (3-5)$$

Les contraintes de notre modèle sont :

$$\sum_{n=1}^N x_{nb} \leq S, \forall b \in \mathcal{B} \quad (3-6)$$

$$\sum_{b=1}^B x_{nb} = 1, \forall n \in \mathbb{N} \quad (3-7)$$

$$T_{kb} = \sum_{n=1}^N x_{nb} * U_{kn}, \forall b \in \mathcal{B}; \forall k \in \mathcal{K} \quad (3-8)$$

$$x_{nb} \in \{0; 1\}, \forall n \in \mathbb{N}; \forall b \in \mathcal{B}.$$

$$T_{kb} \in \{0; 1\}, \forall k \in \mathcal{K}; \forall b \in \mathcal{B}.$$

La contrainte (3-6) impose que le nombre maximal de commandes par lots n'excède pas la taille maximale des lots donné par S . La contrainte (3-7) garantit que chaque commande client est incluse dans un et un seul lot. La contrainte (3-8) permet le calcul de T_{kb} . En effet un lot doit visiter un plateau si au moins une des commandes incluses dans ce lot doit visiter ce plateau.

3.2.3.2 Etude de cas avec des données réelles

Nous avons résolu ce problème d'optimisation décrit ci-dessus avec CPLEX version 12.6.3.0 sur un ordinateur possédant un processeur Intel Core i3 dual 2.3 GHz avec 3Go de mémoire vive et fonctionnant sur l'environnement Windows 7 Professionnel 32 bits.

Notre collaboration avec KLS Logistic nous a permis d'utiliser les bases de données de leurs clients, nous permettant de faire de multiples expériences. Nous montrons, dans cette partie, les résultats des données extraites de deux différentes entreprises de deux secteurs très différents : un Hôpital en France et une entreprise en Suisse. Par souci de confidentialité des données, nous ne donnerons pas leurs noms et les appellerons : Hôpital Français et Entreprise Suisse. La grande

différence entre les deux jeux de données est le nombre de commandes ainsi que le nombre moyen de lignes de commandes par commande de chacun des jours présentés. Il y a plus de commandes ainsi que plus de lignes de commandes par jour dans le cas de l'entreprise Suisse que dans le cas de l'Hôpital Français.

Pour chacun de ces établissements, nous prenons pour stratégie actuelle de regroupement de commandes la politique consistant à incorporer une commande dans un lot en fonction de la date à laquelle cette commande a été effectivement préparée dans l'entrepôt. Ainsi l'ajout des commandes à un lot est fait jusqu'à ce que le lot soit rempli et ce procédé continue jusqu'à ce qu'il n'y ait plus de commandes à regrouper.

Pour les exemples traités, les commandes sont préparées systématiquement pour une livraison le lendemain ou alors préparées le matin pour une livraison l'après-midi. Nous sommes donc dans une situation où les contraintes de dates de livraison n'ont en fait pas besoin d'être incorporées (elles peuvent être considérées identiques).

Nous comparons les résultats de notre méthode d'optimisation à ceux des stratégies actuelles des entreprises. Cette comparaison est faite par rapport à deux critères : le nombre total de plateaux visités et le temps de préparation de commandes associé. Rappelons, que dans le cas 1 VLM, nous avons démontré que ces deux critères étaient équivalents.

Nous notons que le nombre de plateaux visités est un critère intéressant car il impacte la durée de vie des VLMs. Un plus grand nombre de plateaux sortis implique une plus faible durée de vie de la machine. Ainsi nous utiliserons le critère de comparaison suivant :

$$\text{Gain plateau} = \frac{\text{Nb plateaux visités avec stratégie actuelle} - \text{Nb plateaux visités avec notre méthode}}{\text{Nb plateaux visités avec stratégie actuelle}}$$

Dans notre comparaison concernant le second critère, le temps de préparation de commande, nous supposons que le temps requis au VLM pour changer de plateaux est de $e=1$. Pour le temps de collecte, nous allons considérer 3 cas. D'après KLS Logistic, le temps de collecte d'une ligne de commandes dans les entrepôts représente entre 10% et 30% du temps de changement de plateaux selon les cas. Soit C_1 le temps de préparation de commandes lorsque le temps de collecte est supposé négligeable. En supposant le temps de picking sur un plateau proportionnel au nombre de lignes de commandes à collecter, nous étudions deux différentes configurations : C_2 et C_3 . Dans le cas de C_2 , nous considérons le cas extrême où le temps de collecte d'une ligne de commandes est le même que le temps que le VLM met pour changer de plateau $p=1=e$. Dans le

cas C_3 , nous considérons un temps réaliste de collecte d'une ligne de commandes en supposant $p=0,2$. La comparaison des C_i a été faite en utilisant la formule suivante :

$$\text{Gain } C_i = \frac{C_i \text{ avec stratégie actuelle} - C_i \text{ avec notre méthode}}{C_i \text{ avec stratégie actuelle}}$$

Etude de cas n° 1 : Hôpital Français

Le tableau 1 ci-dessous permet la comparaison entre la stratégie actuelle de regroupement de commandes en lots de l'hôpital Français et notre modèle en termes de nombre de plateaux visités sur l'ensemble des lots. Conformément au mode opératoire de l'Hôpital Français, le nombre maximal de commandes pouvant être traitées simultanément est de 6 ($S = 6$).

Chacune des instances décrites ci-dessous correspond à une journée et représente les commandes clients à préparer durant cette journée. Nous indiquons également le nombre de commandes pour chacune des instances ainsi que le nombre de lignes de commandes. Ce tableau récapitule les valeurs du critère utilisé pour la stratégie actuelle « Nombre de plateaux visités stratégie actuelle » ainsi que pour notre approche « Nombre de plateaux visités notre méthode ». Enfin, ce tableau indique le critère « Gain plateau » comme défini précédemment et le temps de résolution mis par CPLEX pour trouver la solution optimale pour notre approche.

Tableau 1 : Les performances de notre modèle en termes de gain plateau pour l'Hôpital français

Instance	NB Commandes	NB lignes de commandes	NB plateaux visités stratégie actuelle	NB plateaux visités notre approche	Gain plateau	Temps de résolution (en secondes)
Jour 1	19	53	40	29	28%	1,52
Jour 2	20	54	44	33	25%	1,63
Jour 3	20	44	36	27	25%	1,57
Jour 4	20	57	47	34	28%	1,02
Jour 5	35	79	66	43	35%	3,52
Jour 6	43	81	66	38	42%	7,31
Jour 7	63	130	121	58	52%	109,03

Le tableau 2 fournit la comparaison des résultats entre la stratégie actuelle de regroupement des commandes et notre modèle en termes de temps de préparation de commandes. Nous rappelons que C_1 , C_2 et C_3 correspondent respectivement à p négligeable, $p=1$ et $p=0,2$. C_1 et C_2 correspondent à 2 cas extrêmes et C_3 à un cas réaliste. Enfin une représentation graphique de ces gains est montrée dans la figure 22 ci-dessous.

Tableau 2 : Les performances de notre modèle en termes de gain C_i pour l'Hôpital français

Instance	C_1 Stratégie actuelle	C_1 Notre approche	Gain C_1	C_2 Stratégie actuelle	C_2 Notre approche	Gain C_2	C_3 Stratégie actuelle	C_3 Notre approche	Gain C_3
Jour 1	40	29	28%	93	82	12%	50,6	39,6	22%
Jour 2	44	33	25%	98	87	11%	54,8	43,8	20%
Jour 3	36	27	25%	80	71	11%	44,8	35,8	20%
Jour 4	47	34	28%	104	91	13%	58,4	45,4	22%
Jour 5	66	43	35%	145	122	16%	81,8	58,8	28%
Jour 6	66	38	42%	147	119	19%	82,2	54,2	34%
Jour 7	121	58	52%	251	188	25%	147,0	84,0	43%

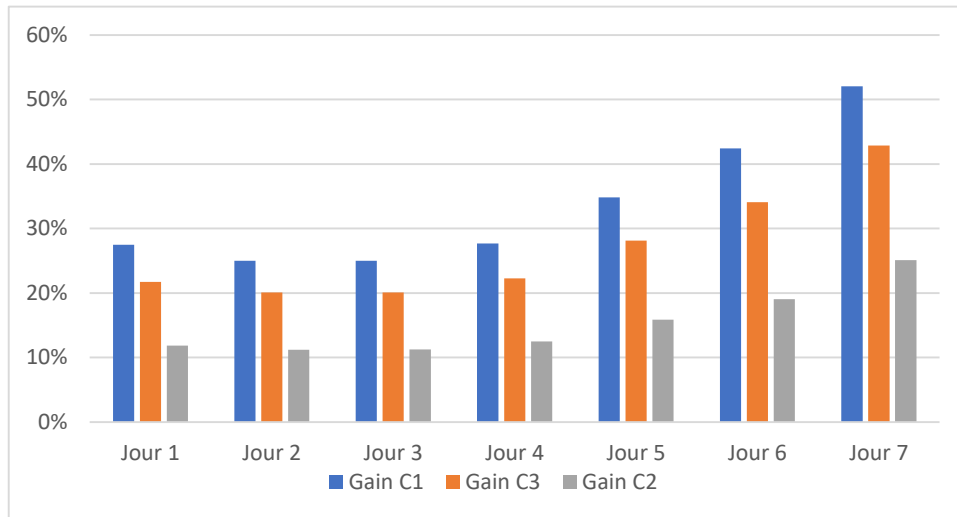


Figure 22 : Gains en temps de préparation pour les trois scénari étudiés

Etude de cas n° 2 : Entreprise Suisse

Nous étudions maintenant le cas de l'entreprise Suisse, spécialisée dans la fabrication d'instruments de mesure de très haute précision. Comme dans le cas de l'Hôpital Français, nous considérons 7 instances différentes correspondant à une journée de commandes. Dans le cas de l'entreprise Suisse, 8 commandes peuvent être traitées simultanément ($S=8$).

Le tableau 3 résume les résultats de manière similaire au cas de l'Hôpital français. Cependant, nous avons stoppé CPLEX après 1 heure de calcul si le résultat optimal n'est pas prouvé avant et nous donnons alors le meilleur résultat obtenu par CPLEX. Dans le tableau 3, nous faisons référence aux instances ne pouvant pas être résolues par CPLEX au bout d'une heure par une Astérisque « 3600* ».

Tableau 3 : Les performances de notre modèle en termes de gain plateau pour l'entreprise Suisse

Instance	NB Commandes	NB lignes de commandes	NB plateaux visités stratégie actuelle	NB plateaux visités notre approche	Gain plateau	Temps de résolution (en secondes)
Jour 1	31	269	170	127	25%	3,40
Jour 2	50	224	169	125	26%	78
Jour 3	60	307	217	157	28%	1190
Jour 4	81	352	275	209	24%	3600*
Jour 5	103	329	248	166	33%	3600*
Jour 6	116	490	327	250	24%	3600*
Jour 7	145	364	281	187	33%	3600*

Le tableau 4 ci-dessous montre les résultats en termes de gain en temps de préparation de commandes pour le cas de l'entreprise Suisse. Une illustration de ces gains est disponible dans figure 23 ci-après.

Tableau 4 : Les performances de notre modèle en termes de gain C_i pour l'entreprise Suisse

Instance	C_1 Stratégie actuelle	C_1 Notre approche	Gain C_1	C_2 Stratégie actuelle	C_2 Notre approche	Gain C_2	C_3 Stratégie actuelle	C_3 Notre approche	Gain C_3
Jour 1	170	127	25%	439	396	10%	223,8	180,8	19%
Jour 2	169	125	26%	393	349	11%	213,8	169,8	21%
Jour 3	217	157	28%	524	464	11%	278,4	218,4	22%
Jour 4	275	209	24%	627	563	10%	345,4	281,4	19%
Jour 5	248	166	33%	577	495	14%	313,8	231,8	26%
Jour 6	327	250	24%	817	740	9%	425,0	348,0	18%
Jour 7	281	187	33%	645	551	15%	353,8	259,8	27%

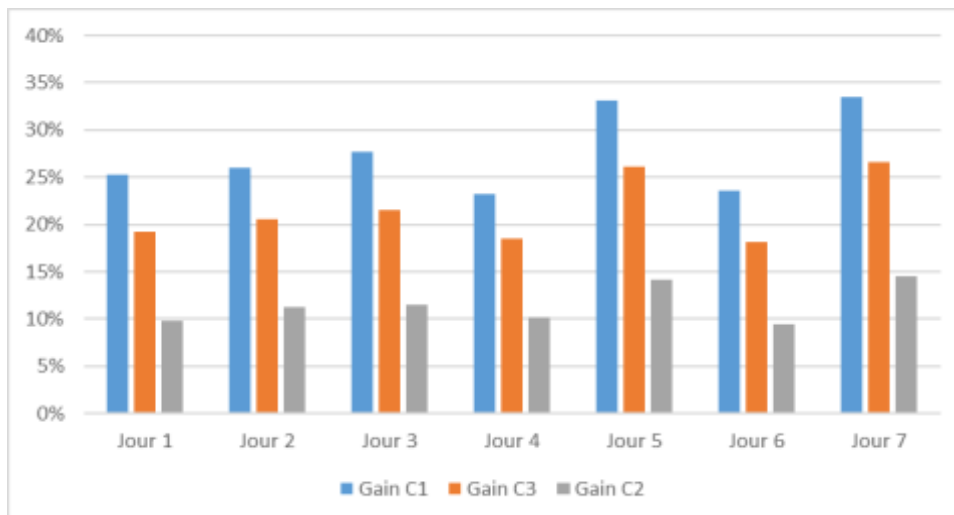


Figure 23 : Gains en temps de préparation pour les trois scénarii étudiés

Discussion générale des résultats

Ces résultats nous ont permis de comparer les gains en termes de nombre de plateaux sortis (critère d'optimisation) et de temps de préparation de commandes.

Nous observons dans les tableaux 1 et 3 (pour respectivement l'hôpital Français et l'entreprise Suisse) que le critère de comparaison « gain plateau » s'étend de 25% à 52% pour l'hôpital Français et de 24% à 33% pour l'entreprise Suisse. Les tableaux 2 et 4 résument les gains en termes de temps de préparation, critère primordial pour les managers au sein des entrepôts. Concernant l'Hôpital Français, nous observons des gains C_1 allant de 25% à 52%, la moyenne étant de 34% de gains (ce sont les mêmes gains que le « gain plateau », puisque le temps de collecte est supposé nul). Dans le cas considérant le temps de picking proportionnel au nombre de lignes de commandes à collecter avec $p=1$, les gains C_2 vont de 11% à 25% avec une moyenne égale à 15%. Enfin dans le cas réaliste $p=0,2$, nous observons des gains C_3 s'étendant de 20% à 43% avec une moyenne de 27%. Concernant l'entreprise suisse, nous observons les gains allant de 24% à 33% avec une moyenne de 28% pour le cas temps de picking négligeable. Les gains concernant C_2 sont entre 9% et 15% avec une moyenne à 11%. Enfin les gains concernant C_3 s'étalent de 18% à 27% avec une moyenne à 22%.

Nous notons une baisse des gains significative lorsque le paramètre p augmente. En effet le temps total de collecte ($p * \sum_{b=1}^B \sum_{l=1}^{K^b} L_l^b$) est le même quel que soit la solution et donc ne peut pas être minimisé. Naturellement, plus p est élevé plus le temps de collecte est important par rapport au temps d'attente, ainsi donc moins le temps de préparation de commande C peut être amélioré.

Dans le cas de l'Hôpital Français, les temps de résolution n'excèdent pas une minute de calcul pour la plupart des instances, seule l'instance ayant 63 commandes a un temps de résolution supérieur à 1 minute avec 109 secondes requises pour que CPLEX confirme l'optimalité du résultat. Pour l'entreprise Suisse, nous observons que CPLEX ne confirme pas l'optimalité du résultat trouvé dans toutes les instances. En effet lorsque le nombre de commandes excèdent 60, nous observons que CPLEX n'arrive pas à confirmer l'optimalité en moins de 1 heure de calcul. Dans ce cas, la meilleure solution trouvée par CPLEX est retenue.

Effet du nombre de commandes

Afin d'approfondir notre analyse des résultats, nous étudions l'effet du nombre de commandes sur les performances de notre approche. Ainsi nous créons des sous-ensembles d'instances en fonction du nombre de commandes à préparer pour chacune des entreprises, et analysons les résultats associés à ces sous-ensembles.

Nous créons deux sous-ensembles en fonction de la valeur médiane de chacun des ensembles d'instances. Ainsi pour chacune des entreprises, nous avons un premier sous ensemble qui

rassemble la moitié des jours ayant un nombre de commandes inférieur à la médiane, l'autre moitié des instances est placée dans le sous-ensemble ayant un nombre de commandes supérieur à la médiane. Lorsqu'il y a un nombre impair d'instances, le résultat médian est égal à une des valeurs, ainsi nous devons choisir dans lequel des deux sous-ensembles positionner cette instance. Nous choisissons la règle suivante. Si la moyenne de l'ensemble des données est supérieure à la valeur médiane, alors l'instance intermédiaire sera placée dans le premier sous-ensemble comprenant les valeurs inférieures à la médiane. Par exemple pour l'Hôpital Français, au regard du nombre de commandes par instance, nous avons la médiane égale à 20 et un nombre impair d'instances, la moyenne étant égale à 31. Nous aurons donc les deux sous-ensembles suivants : Le premier sous-ensemble comprend les instances ayant un nombre de commandes par jour inférieur ou égal à 20, et le second sous-ensemble les instances ayant plus de 20 commandes par jour à regrouper. Nous faisons de même avec l'Entreprise Suisse : la médiane est 81 et la moyenne 83, nous avons un premier sous-ensemble comprenant les jours ayant 81 commandes ou moins, tandis que le second sous-ensemble comprend les instances ayant un nombre de commandes à regrouper supérieur à 81.

Nous analysons le « gain plateau » moyen pour chacun des sous-ensembles et le gain C_2 , celui-ci faisant référence au cas réaliste prescrit par KLS Logistic. Les résultats des différents sous-ensembles sont exprimés dans le tableau suivant.

Tableau 5 : Effet du nombre de commandes sur les performances de notre modèle

Entreprise	Sous ensemble d'instances	Gain plateau moyen	Gain C_3 moyen
Hôpital Français	Instances ayant 20 commandes ou moins	26%	21%
	Instances ayant plus de 20 commandes	43%	35%
Entreprise Suisse	Instance ayant 81 commandes ou moins	26%	20%
	Instances ayant plus de 81 commandes	30%	24%

Ces résultats montrent que les instances ayant un plus grand nombre de commandes donnent des gains plus importants. Nous expliquons cela par le fait qu'avoir plus de commandes implique une plus grande chance d'avoir des commandes « similaires » en termes des plateaux vus par celles-ci et donc de pouvoir créer des lots ayant des commandes « similaires ». Donc avoir des instances avec un grand nombre de commandes accroît l'intérêt de notre modèle en pratique.

Effet du nombre de lignes de commandes

Nous nous focalisons maintenant sur le nombre de lignes de commandes moyen pour chacune des instances. Nous créons les sous-ensembles de la même façon que dans la partie précédente. Pour l'Hôpital Français, la médiane est de 2.26 lignes de commandes par commandes pour les

instances étudiés. De plus la moyenne est de 2.39. Ainsi le premier sous-ensemble inclut les instances ayant une moyenne de 2.26 lignes de commandes par commande ou moins, tandis que le second sous ensemble inclut les instances ayant une moyenne de lignes de commandes par commande supérieure à 2.26. Concernant l'Entreprise Suisse, nous observons une valeur médiane à 4.34 et une valeur moyenne à 4.64. Les sous-ensembles d'instance et leurs gains sont fournis dans le tableau suivant :

Tableau 6 : Effet du nombre de lignes de commandes sur les performances de notre modèle

Entreprise	Sous ensemble d'instances	Gain plateau moyen	Gain C_3 moyen
Hôpital Français	Instances ayant une moyenne de lignes de commandes inférieure ou égal à 2.26	39%	31%
	Instances ayant une moyenne de lignes de commandes supérieure à 2.26	27%	21%
Entreprise Suisse	Instances ayant une moyenne de lignes de commandes inférieure ou égal à 4.34	28%	22%
	Instances ayant une moyenne de lignes de commandes supérieure à 4.34	26%	20%

Nous observons qu'une augmentation du nombre de lignes de commandes par commande détériore les performances de notre modèle. Nous expliquons ce résultat surprenant de la manière suivante. Si nous avons moins de lignes de commandes par commande, la probabilité d'avoir plusieurs commandes visitant exactement les mêmes plateaux est plus grande que lorsque les commandes ont un grand nombre de lignes de commandes. Ainsi il devient plus probable de trouver des commandes similaires (i.e. visitant exactement les mêmes plateaux).

Nous voyons donc que les deux paramètres précédemment étudiés, i.e. le nombre de commandes ainsi que le nombre de lignes de commandes par commandes, ont un impact sur les performances de notre modèle de regroupement de commandes.

3.3 Regroupement de commandes dans un système de préparation de commandes utilisant plusieurs VLMs

Nous considérons dans cette section le cas d'une zone de préparation constituée de plusieurs VLMs. La différence fondamentale entre le cas où la préparation s'effectue sur un VLM et celle où la préparation s'effectue sur plusieurs VLMs est que le changement de plateaux peut se faire en temps masqué. En effet, lorsque le préparateur effectue la collecte sur un VLM, les autres VLMs effectuent un changement de plateaux, permettant ainsi à l'opérateur d'avoir des temps d'attente moins importants devant les VLMs.

Notre objectif principal reste inchangé, développer un modèle d'optimisation pouvant être utilisé en pratique par les managers afin d'avoir un regroupement de commandes minimisant le temps de préparation de commandes. Tout comme lors de la partie précédente, nous évaluons le temps de préparation de commandes à travers deux situations (temps de picking négligeable ou proportionnel au nombre de lignes à collecter par plateaux). À partir de cette évaluation, nous déterminons un objectif d'optimisation et le modèle d'optimisation associé. Enfin nous présentons les études de cas suivies d'une discussion des résultats obtenus.

3.3.1 Notations et hypothèses utilisées

Nous gardons et adaptons les hypothèses utilisées dans la partie précédente au cas où la préparation s'effectue sur plusieurs VLMs. Cette section énonce les ajouts effectués :

- V : Nombre de VLMs considérés.
- $\mathcal{V} = \{1, \dots, V\}$: ensemble des indices des VLMs, indexé par v .
- K_v : Nombre total de plateaux présents au sein du VLM v .
- $\mathcal{K}_v = \{1, \dots, K_v\}$: Ensemble des indices des plateaux du VLM v , indexé par k .
- K_v^b : Nombre de plateaux visités au cours du lot b sur le VLM v
- W_{lv}^b : Temps d'attente devant le l -ième plateau visité sur le VLM v pour le lot b . Ce paramètre représente le temps où le préparateur est en face du VLM v en attendant que celui-ci soit disponible pour la collecte au cours du lot b . Nous avons la relation suivante : $W_l^b = \sum_{v=1}^V W_{lv}^b$.

- P_{lv}^b : Temps de collecte des références sur le l -ième plateau du VLM v pour le lot b . Nous avons la relation suivante, $P_l^b = \sum_{v=1}^V P_{lv}^b$.

- L_{lv}^b : Nombre de lignes de commandes sur le l -ième plateau devant être visité par le lot b sur le VLM v .

Dans ce paragraphe nous décrivons les hypothèses que nous ajoutons dans le cas où la collecte s'effectue sur plusieurs VLMs. Nous imposons tout d'abord la séquence de visite des VLMs pour le préparateur. Ainsi pour un lot b donné, le préparateur commence par le VLM ayant le plus grand nombre de plateaux à visiter pour ce lot b . Le second VLM visité sera celui ayant le second plus grand nombre de plateaux à visiter pour le lot b , et ce jusqu'au dernier VLM qui est celui ayant le plus faible nombre de plateaux à visiter pour le lot b . Une fois la première tournée de VLMs terminée (i.e. le premier plateau de chacun des VLMs est visité), le préparateur revient au premier VLM visité et garde la même séquence de visite de VLMs, jusqu'à ce que toutes les références associées aux commandes incluses dans le lot b soient collectées. De plus, nous faisons l'hypothèse que le temps de marche entre la zone où les bacs sont disposés et les divers VLMs est négligeable, hypothèse réaliste au vu de la faible distance à parcourir entre les VLMs et les bacs.

Pour un lot b donné, nous renommons les différents VLMs tel que le VLM1 soit celui visité en premier et donc ayant le plus grand nombre de plateaux à visiter pour le lot b . Le VLM2 est celui ayant le second plus grand nombre de plateaux à visiter. Ainsi de suite jusqu'au VLM V qui a le plus faible nombre de plateaux à visiter pour le lot b . Conformément à cette notation, nous avons : $K_1^b \geq K_2^b \geq \dots \geq K_V^b \forall b \in \mathcal{B}$. Il est cependant important de noter que le VLM 1 d'un lot ne sera pas obligatoirement le VLM 1 du lot suivant.

3.3.2 Evaluation du temps de préparation de commandes sur plusieurs VLMs

Le but de cette section est de pouvoir évaluer le temps de préparation de commandes C associé à la collecte sur les V VLMs de B lots de commandes. Nous considérons les deux situations suivantes :

- Temps de collecte considéré négligeable, impliquant $P_{lv}^b = 0 \forall v \in \vartheta; \forall b \in \mathcal{B}$ et $\forall l \in \{1, \dots, K_1^b\}$
- Temps de collecte sur un plateau proportionnel au nombre de lignes à collecter sur ce plateau, impliquant $P_{lv}^b = L_{lv}^b * p \forall v \in \vartheta; \forall b \in \mathcal{B}$ et $\forall l \in \{1, \dots, K_1^b\}$

3.3.2.1 Temps de préparation de commandes sur V VLMs considérant un temps de collecte négligeable.

Proposition 2. Dans un système de préparation de commandes équipé de V VLMs, dans le cas du temps de collecte négligeable, minimiser le temps de préparation de commandes est équivalent à minimiser la somme sur les différents lots du maximum de plateaux à visiter pour chacun des lots. Le temps de préparation de commandes est donné par : $C = e * \sum_{b=1}^B \max_v K_v^b$

Remarque : Suite à notre façon de numérotter les VLMs en fonction du nombre de plateaux à visiter pour chacun des lots, nous pouvons écrire $C = e * \sum_{b=1}^B K_1^b$.

Nous présentons un exemple avant de démontrer la proposition 2.

Exemple 3 : Temps de préparation de commandes considérant un temps de collecte négligeable sur V VLMs

Nous considérons l'exemple suivant : 3 VLMs sont présents pour la collecte de références pour un lot b donné, et le nombre de plateaux à visiter pour ce lot est $K_1^b = 4$; $K_2^b = 3$ et $K_3^b = 2$. La figure 24 ci-après illustre les différents temps et montre l'utilisation du temps masqué.

Le préparateur de commandes commence par la visite du premier VLM (celui ayant le nombre de plateaux à visiter le plus élevé pour le lot b) et doit attendre que le premier plateau de ce VLM soit extrait. Cette opération prend e unités de temps. Grâce aux différentes machines opérant simultanément afin de favoriser la recherche de plateaux en temps masqué, l'ensemble des VLMs présents vont extraire le premier plateau sur lequel la collecte doit être faite. Le VLM 2 et 3 ayant déjà extrait leur plateau lorsque le préparateur termine la collecte sur le VLM 1, et le temps de déplacement de l'opérateur étant négligeable, la collecte se fait instantanément sur les VLM 2 et 3. Ainsi le temps de la tournée de visite du premier plateau pour l'ensemble VLMs est égal à e .

Le préparateur continue la collecte des références de ce lot en revenant au premier VLM. Il doit attendre que ce VLM range le premier plateau visité et extrait le second. Cette opération prend e unités de temps. Nous rappelons que cette opération se fait simultanément sur tous les VLMs. Donc, tout comme pour le premier plateau visité par les VLMs, le préparateur collecte sur l'ensemble des VLMs (instantanément comme le temps de collecte est négligeable). Ainsi le second plateau du VLM 2 et du VLM 3 seront visités par le préparateur.

Une fois cela fait, il revient au premier VLM pour la collecte du troisième plateau. Ici seuls les VLM 1 et VLM 2 changent de plateaux comme seuls ces deux-là ont encore des plateaux à visiter

(puisque $K_3^b = 2$). La collecte du troisième plateau est de nouveau instantanée, la tournée de visite du troisième plateau prend donc e unités de temps.

Enfin, il ne reste plus que la collecte sur le dernier plateau à visiter par le VLM 1 sur le lot b . Suivant le même procédé décrit précédemment, le dernier plateau devant être visité fait attendre l'opérateur e unités de temps avant que la collecte ne soit faite instantanément sur ce plateau, finissant ainsi la collecte des références du lot b .

Ainsi, le temps total de préparation de commandes est composé de la somme des temps d'attente devant le premier VLM, égale à $4 * e$, soit $K_1^b * e$.

VLM 1	e	e	e	e
VLM 2	e	e	e	
VLM 3	e	e		
	W_1^b	W_2^b	W_3^b	W_4^b
	C_1^b	C_2^b	C_3^b	C_4^b
	C^b			

Figure 24 : Illustration du temps de préparation de commandes sur 3 VLMs lorsque le temps de collecte est considéré négligeable

Preuve de la Proposition 2

Dans le cas de temps de collecte négligeable (i.e., $P_l^b \cong 0$ et donc $P^b \cong 0$), nous avons pour un lot de commandes à collecter la relation suivante : $C^b = W^b = \sum_{v \in \mathcal{D}} \sum_{l=1}^{K_v^b} W_{lv}^b = \sum_{l=1}^{K_1^b} W_{l1}^b$

Nous rappelons que le temps de changement de plateaux sur chacun des VLMs est le même et égal à e et que les temps d'extraction des plateaux sur les VLMs 2 à V sont masqués par le temps d'attente passé sur le VLM 1.

Ainsi le temps de préparation sur un lot b donné est : $C^b = \sum_{l=1}^{K_1^b} W_{l1}^b = \sum_{l=1}^{K_1^b} e = K_1^b * e$. Et donc le temps de préparation de commandes sur l'ensemble des lots est, $C = \sum_{b=1}^B C_b = \sum_{b=1}^B (K_1^b * e) = e * \sum_{b=1}^B K_1^b$. ■

3.3.2.2 Temps de préparation de commandes sur V VLMs considérant un temps de collecte par plateau proportionnel au nombre de lignes.

Nous évaluons le temps de préparation de commandes sur V VLMs lorsque le temps de collecte par plateau est proportionnel au nombre de lignes à collecter sur ce même plateau.

Proposition 3. Dans un système de préparation de commandes équipé de V VLMs, dans le cas du temps de collecte par plateau proportionnel au nombre de lignes à collecter sur ce plateau, le temps de préparation de commandes est donné par :

$$C = \sum_{b=1}^B \sum_{l=1}^{K_1^b} \sum_{v=1}^V (P_{lv}^b + W_{lv}^b), \text{ Où :}$$

$$W_{lv}^b = \max(e - (\sum_{i=1}^{v-1} (L_{li}^b * p + W_{li}^b) + \sum_{i=v+1}^V (L_{l-1i}^b * p + W_{l-1i}^b)); 0) \quad \forall 1 < v < V,$$

$$W_{l1}^b = \max(e - \sum_{i=2}^V (L_{l-1i}^b * p + W_{l-1i}^b); 0),$$

$$W_{lV}^b = \max(e - \sum_{i=1}^{V-1} (L_{li}^b * p + W_{li}^b); 0),$$

$$\text{Et } P_{lv}^b = L_{lv}^b * p.$$

Nous présentons un exemple avant de démontrer la proposition 3.

Exemple 4 : Temps de préparation de commandes sur V VLMs, considérant un temps de collecte par plateau proportionnel au nombre de lignes à collecter

Dans cet exemple, nous considérons 3 VLMs et les caractéristiques suivantes concernant la collecte du lot b :

$$K_1^b = 4; L_{11}^b=1; L_{21}^b=1; L_{31}^b=1; L_{41}^b=1.$$

$$K_2^b = 3; L_{12}^b=2; L_{22}^b=1; L_{32}^b=2.$$

$$K_3^b = 2; L_{13}^b=2; L_{23}^b=2.$$

La préparation commence par un temps d'attente devant le premier VLM, prenant e unités de temps. Après ce temps d'attente le préparateur collecte la ligne sur ce premier plateau du VLM 1, prenant $P_{11}^b = p$ unités de temps (car $L_{11}^b = 1$). Le préparateur se déplace sur le second VLM, celui-ci ayant déjà le premier plateau disponible pour la collecte. Deux lignes de commandes sont collectées sur ce plateau ($L_{12}^b=2$) et $P_{12}^b = 2 * p$. S'en suit la collecte sur le VLM 3 ayant son premier plateau disponible, prenant $P_{13}^b = 2 * p$ unités de temps sachant que $L_{13}^b = 2$.

Après cette première tournée où le préparateur a visité le plateau de chacun des VLMs présent, il commence la seconde tournée par le VLM 1. Le temps d'attente W_{21}^b , si celui-ci existe, est la différence entre le temps requis par le premier VLM pour changer de plateau (e unités de temps) et le temps passé entre le moment où le préparateur a fini la collecte sur ce premier plateau du VLM 1 et le moment où le préparateur revient au VLM 1 pour visiter le second plateau. Ce second

temps est celui pendant lequel l'opérateur attend sur les autres VLMs augmenté du temps de collecte sur les VLMs autres que le VLM1, étant égal à $W_{12}^b + P_{12}^b + W_{13}^b + P_{13}^b = 0 + 2 * p + 0 + 2 * p$. Si cette différence entre les deux temps énoncés est négative, alors il n'y aura pas de temps d'attente car le plateau sera disponible pour la collecte avant que le préparateur ne revienne. Ainsi $W_{21}^b = \max(e - W_{12}^b - P_{12}^b - W_{13}^b - P_{13}^b; 0) = \max(e - 4 * p; 0)$. Après ce possible temps d'attente, le préparateur collecte une ligne de commandes, sachant que $L_{21}^b = 1$, prenant donc $P_{21}^b = p$ unités de temps, avant de se déplacer sur le VLM2. Le temps d'attente en face du VLM 2 se calcule comme pour celui décrit ci-dessus. Ce temps d'attente existera si le préparateur passe plus de temps (d'attente et de collecte) sur les VLM 3 et VLM1, depuis qu'il a quitté le VLM 2, que le temps que ce VLM met pour changer de plateau. Ainsi $W_{22}^b = \max(e - W_{13}^b - P_{13}^b - W_{21}^b - P_{21}^b; 0) = \max(e - \max(e - 4 * p; 0) - 3 * p; 0)$. Après ce possible temps d'attente, le préparateur collecte une ligne de commandes $P_{22}^b = p$. Le temps d'attente en face du VLM 3 est : $W_{23}^b = \max(e - W_{21}^b - P_{21}^b - W_{22}^b - P_{22}^b; 0)$ et le temps de collecte sur ce troisième VLM est $P_{23}^b = 2 * p$. Cela finit la visite du second plateau à visiter par les VLMs.

Le préparateur commence la troisième tournée visitant les plateaux des VLMs 1 et 2 (le troisième n'ayant plus de plateaux à visiter sur le lot b ($K_3^b = 2$)) par un possible temps d'attente. Ainsi il devra attendre $W_{31}^b = \max(e - W_{22}^b - P_{22}^b - W_{23}^b - P_{23}^b; 0)$ et collectera sur le troisième plateau une ligne de commande prenant $P_{31}^b = p$. De la même façon, le temps d'attente sur le second VLM pour le troisième plateau à visiter est $W_{32}^b = \max(e - W_{23}^b - P_{23}^b - W_{31}^b - P_{31}^b; 0)$ et le temps de collecte est $P_{32}^b = 2 * p$.

Enfin le VLM1 a encore un plateau à visiter et le possible temps d'attente devant celui-ci est de $W_{41}^b = \max(e - W_{32}^b - P_{32}^b; 0)$. Finalement, l'opérateur collecte la ligne de commande prenant $P_{41}^b = p$. La figure 25 ci-dessous illustre les différents temps nous permettant de calculer C. Sur cette illustration, nous considérons $p = 0,2 * e$.

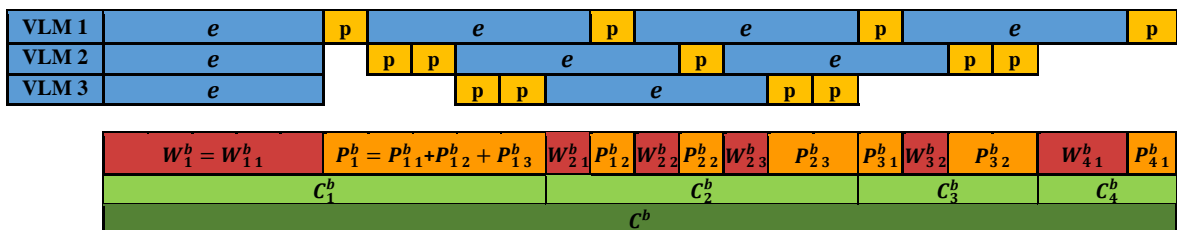


Figure 25 : Illustration du temps de préparation de commandes sur 3 VLMs lorsque le temps de collecte par plateau est considéré proportionnel au nombre de lignes de commandes à collecter sur ce plateau ($p = 0,2 * e$)

Preuve de la Proposition 3.

Par définition, nous avons $C = \sum_{b=1}^B \sum_{l=1}^{K_1^b} \sum_{v=1}^V (P_{lv}^b + W_{lv}^b)$. Sachant que le temps de préparation de commandes sur un plateau est proportionnel au nombre de lignes de commandes ayant besoin d'être collectées sur ce même plateau, le temps de collecte sur le l -ième plateau du VLM v pour lot b est $P_{lv}^b = L_{lv}^b * p$.

Le temps d'attente en face du l -ième plateau du VLM v pour lot b , W_{lv}^b , si celui-ci existe, est la différence entre le temps requis par le VLM v pour ranger le $(l - 1)$ ième plateau et extraire le l -ième, et le temps passé entre le moment où le préparateur finit de collecter sur le plateau $(l - 1)$ du VLM v et le moment où il revient sur ce VLM v pour la collecte sur le l -ième plateau. Si il existe un temps d'attente du l -ième plateau du VLM v pour le lot l , alors W_{lv}^b est donnée par la formule (1) – (2) – (3) où :

(1) Le temps requis par le VLM pour stocker le $(l - 1)$ -ième plateau et extraire le l -ième. Ce temps est constant pour chacun des VLMs et est égal à e .

(2) Le temps passé par l'opérateur pour collecter les références du $(v+1)$ ième au V ième VLMs sur le $(l - 1)$ ième plateau à visiter et du 1 au $(v-1)$ ième VLMs sur le l -ième plateau. Ce temps est égal à $\sum_{i=v+1}^V L_{l-1 i}^b * p + \sum_{i=1}^{v-1} L_{li}^b * p$,

(3) Les temps d'attente en face du $(v+1)$ ième au V ième VLM du $(l - 1)$ ième plateau à visiter et du 1 au $(v-1)$ ième VLM du l -ième plateau à visiter. Ces temps d'attente sont égaux à $\sum_{i=v+1}^V W_{l-1 i}^b + \sum_{i=1}^{v-1} W_{li}^b$.

Si la Formule (1) – (2) – (3) donne une valeur négative, alors il n'y aura pas de temps d'attente et dans ce cas $W_{lv}^b = 0$. Le temps d'attente en face du l -ième plateau du VLM v pour lot b est donc donné par

$$W_{lv}^b = \max(e - (\sum_{i=1}^{v-1} (L_{li}^b * p + W_{li}^b) + \sum_{i=v+1}^V (L_{l-1 i}^b * p + W_{l-1 i}^b)); 0) \text{ pour } 1 < v < V.$$

De la même façon,

$$W_{l1}^b = \max(e - \sum_{i=2}^V (L_{l-1 i}^b * p + W_{l-1 i}^b); 0),$$

$$W_{lV}^b = \max(e - \sum_{i=1}^{V-1} (L_{li}^b * p + W_{li}^b); 0) \blacksquare$$

3.3.3 Modèle d'optimisation et étude de cas

Au cours de cette section, nous développons le modèle d'optimisation permettant un regroupement de commandes en lots dans le cas de plusieurs VLMs. Puis nous présentons et discutons les résultats des études de cas permettant de valider notre modèle et d'évaluer ses performances.

Nous observons que la formulation du temps de préparation de commandes dans le cas multiples VLMs est extrêmement complexe. Inclure directement le temps de préparation de commandes exprimé en proposition 3 comme fonction objectif de notre modèle d'optimisation donnerait un modèle difficilement solvable par des méthodes exactes.

Ainsi nous décidons d'utiliser une fonction objectif intermédiaire, qui est la somme du nombre de plateaux devant être visité par le « premier VLM » de chacun des lots. Nous rappelons que le premier VLM est celui devant visiter le plus grand nombre de plateaux pour chacun des lots ; ce n'est pas nécessairement le même VLM d'un lot à l'autre. Ce choix est motivé par plusieurs arguments :

Tout d'abord, nous avons démontré que, dans le cas multiple VLMs avec temps de collecte négligeable, la minimisation du temps de préparation de commandes est équivalente à la minimisation de ce critère. Deuxièmement nous observons que le temps de préparation sera d'autant plus faible que les changements de plateaux se font en temps masqué. Or le critère choisi favorise effectivement le travail en temps masqué en réduisant le nombre maximal de plateaux vu sur les stockeurs, essayant donc d'équilibrer les nombres de plateaux visités sur les différents VLMs, et par conséquent, de ne pas faire la collecte sur un seul stockeur. Enfin, nos études de cas montrent une corrélation positive entre la minimisation de ce critère et la minimisation du temps de préparation de commandes montré dans la Proposition 3, comme nous le verrons en 3.3.3.2.

Nous utilisons donc le critère *Minimisation* $\sum_{b=1}^B K_1^b$ comme fonction objectif de notre modèle d'optimisation. Nous rappelons une nouvelle fois que K_1^b est le résultat de notre stratégie de regroupement de commandes dans notre modèle et donc doit être exprimé comme fonction des variables de notre modèle.

3.3.3.1 Modèle d'optimisation

La principale décision de notre modèle est la construction de différents lots à partir d'un portefeuille de commandes donné. Les variables de décision sont :

- $x_{nb}=1$ si la commande n est incluse dans le lot b ; 0 sinon.

- $T_{kvb}=1$ si le plateau k du VLM v est visité durant le lot b ; 0 sinon.

Nous utilisons aussi la variable d'entrée U_{kvn} , déduite des données du problème, U_{kvn} égal à 1 quand la commande n contient une référence positionnée sur le plateau k du VLM v ; 0 sinon.

Le nombre de plateaux visités par le premier VLM pour un lot b donné est $K_1^b = \max_{v \in \mathcal{V}} (\sum_{k=1}^K T_{kvb})$.

Ainsi notre fonction objectif est :

$$\text{Minimiser } \sum_{b=1}^B \max_{v \in \mathcal{V}} (\sum_{k=1}^K T_{kvb}) \quad (3-11)$$

Les contraintes de notre modèle sont :

$$\sum_{n=1}^N x_{nb} \leq S, \quad \forall b \in \mathcal{B} \quad (3-12)$$

$$\sum_{b=1}^B x_{nb} = 1, \quad \forall n \in \mathcal{N} \quad (3-13)$$

$$T_{kvb} = \sum_{n=1}^N x_{nb} * U_{kvn}, \quad \forall b \in \mathcal{B}, \quad \forall k \in \mathcal{K}_v, \quad \forall v \in \mathcal{V} \quad (3-14)$$

$$x_{nb} \in \{0; 1\}, \quad \forall n \in \mathcal{N}; \quad \forall b \in \mathcal{B}$$

$$T_{kvb} \in \{0; 1\}, \quad \forall k \in \mathcal{K}_v; \quad \forall v \in \mathcal{V}; \quad \forall b \in \mathcal{B}.$$

La contrainte (3-12) impose que le nombre maximal de commandes par lots n'excède pas S . La contrainte (3-13) garantit que chaque commande client doit être incluse dans un lot et un seul. La contrainte (3-14) permet le calcul de T_{kvb} . En effet un lot doit visiter un plateau k d'un VLM v si au moins une des commandes incluses dans ce lot doit visiter ce plateau de ce VLM.

3.3.3.2 Etude de cas utilisant des données réelles

Nous comparons donc les performances de notre stratégie de regroupement avec celles de la stratégie utilisée par l'Hôpital Français (disposant d'une zone comprenant 2 VLMs) et de l'entreprise Suisse (disposant d'une zone comprenant 4 VLMs) en fonction de deux critères : le critère d'optimisation proposé et le temps total de préparation de commandes donné en Proposition 3.

$$\text{Gain plateau} = \frac{\text{Nb plateaux visités VLM1 avec la stratégie actuelle} - \text{Nb plateaux visités VLM1 avec notre méthode}}{\text{Nb plateaux visités VLM1 avec la stratégie actuelle}}$$

$$\text{Gain } C_i = \frac{C_i \text{ avec la stratégie actuelle} - C_i \text{ avec notre méthode}}{C_i \text{ avec la stratégie actuelle}}$$

Le critère « Gain plateau » reflète la comparaison en termes du critère choisi pour notre optimisation tandis que le second critère (Gain C_i) reflète la différence des temps de préparation de commandes sous les 3 hypothèses de temps de collecte.

Tout comme dans le cas 1 VLM, nous supposons que le temps nécessaire au VLM pour changer de plateaux est de $e=1$. Nous utilisons les précédentes notations pour le temps de préparation de commandes en fonction de l'hypothèse utilisée : C_1 est le temps de préparation de commandes lorsque le temps de collecte est supposé négligeable. Lorsque le temps de collecte sur un plateau est proportionnel au nombre de lignes de commandes à collecter, nous étudions deux différentes configurations : C_2 et C_3 . Pour C_2 , nous considérons le cas extrême où le temps de collecte d'une ligne de commandes égale le temps qu'un VLM met pour changer de plateau $p=1=e$. Dans le cas C_3 , nous considérons le cas réaliste en supposant $p=0,2$. Nous rappelons que d'après KLS Logistic, ce temps est compris entre 0.1 et 0.3.

Nous avons codé puis résolu le problème d'optimisation décrit dans la section 2.3.1 avec CPLEX version 12.6.3.0 sur un ordinateur possédant un processeur Intel Core i3 dual 2.3 GHz avec 3Go de mémoire vive et fonctionnant sur l'environnement Windows 7 Professionnel 32 bits. De plus nous stoppons CPLEX après une heure de calcul si la solution optimale n'a pas été prouvée. Nous faisons référence à de telles instances en notant « 3600* » pour le temps de résolution. Dans ce cas, nous prenons la dernière solution fournie par CPLEX.

Etude de cas n° 1 : Hôpital Français

La zone de préparation de l'Hôpital Français prise en compte dans cette étude de cas est constitué de 2 VLMs disposant de 35 plateaux chacun. Tout comme le cas 1 VLM, le nombre de commandes par lot est de 6 maximum. Ainsi nous réglons les paramètres $V = 2$, $K_1 = K_2 = 35$ et $S = 6$.

Nous considérons 10 instances représentant chacune une journée de commandes à préparer. Les tableaux 7 et 8 ci-dessous récapitulent les résultats de la comparaison de notre méthode par rapport à la stratégie actuelle de regroupement de commandes. La figure 10 ci-après illustre les résultats concernant les Gains C_i présentés sur le tableau 26 ci-dessous.

Tableau 7 : Les performances de notre modèle en termes de gain plateaux pour l'Hôpital français

Instance	NB Commandes	NB lignes de commandes	NB plateaux visités stratégie actuelle	NB plateaux visités notre approche	Gain plateau	Temps de résolution (en seconds)
Jour 1	17	32	17	11	35%	2,14
Jour 2	19	36	19	14	26%	2,01
Jour 3	36	83	45	30	33%	2,09
Jour 4	46	65	37	22	41%	14
Jour 5	47	71	46	29	37%	18
Jour 6	50	73	48	27	44%	21
Jour 7	53	119	71	41	42%	650
Jour 8	60	84	56	31	45%	64
Jour 9	63	105	69	33	52%	1320
Jour 10	68	116	82	45	45%	3600*

 Tableau 8 : Les performances de notre modèle en termes de gain C_i pour l'Hôpital français

Instance	C_1 Stratégie actuelle	C_1 Notre approche	Gain C_1	C_2 Stratégie actuelle	C_2 Notre approche	Gain C_2	C_3 Stratégie actuelle	C_3 Notre approche	Gain C_3
Jour 1	17	11	35%	42	35	17%	21,8	15,6	28%
Jour 2	19	14	26%	44	40	9%	23,8	19,2	19%
Jour 3	45	30	33%	107	96	10%	57,0	42,2	26%
Jour 4	37	22	41%	85	77	9%	46,2	32,8	29%
Jour 5	46	29	37%	109	89	18%	58,4	41,0	30%
Jour 6	48	27	44%	105	87	17%	58,8	38,4	35%
Jour 7	71	41	42%	183	142	22%	93,4	61,2	34%
Jour 8	56	31	45%	127	104	18%	70,2	45,6	35%
Jour 9	69	33	52%	167	116	31%	88,2	48,6	45%
Jour 10	82	45	45%	190	151	21%	103,6	66,0	36%

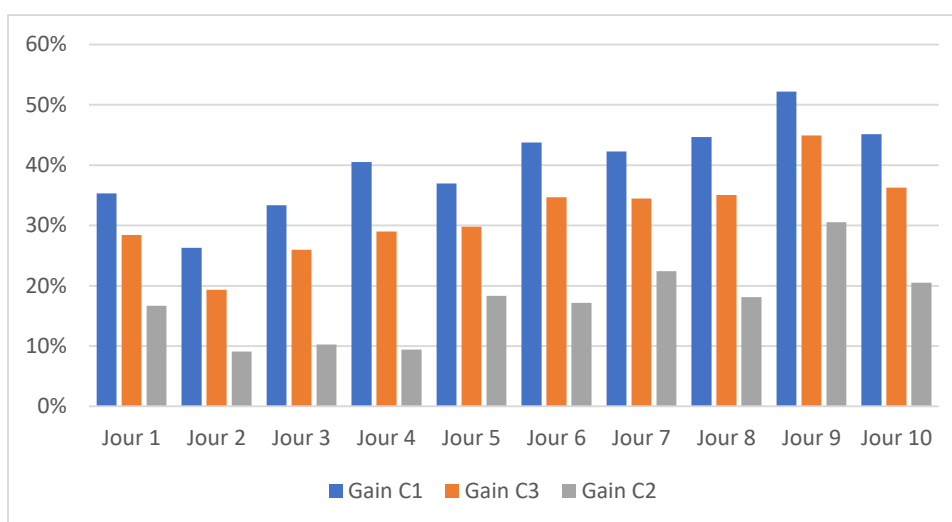


Figure 26 : Gains en temps de préparation pour les trois scénarii étudiés

Etude de cas n° 2 : Entreprise Suisse

Pour l'entreprise Suisse, la zone de préparation de commandes est composée de 4 VLMs ayant 130 plateaux par VLM. Tout comme le cas 1 VLM, le nombre maximal de commandes simultanément présentes en face des VLMs est de 8 commandes. Ainsi nous réglons les paramètres : $V = 4$, $K_1 = K_2 = K_3 = K_4 = 130$ et $S = 8$.

Les tableaux 9 et 10 ci-dessous présentent les résultats obtenus et la Figure 27 illustre les gains en temps de préparation de commandes pour les 3 hypothèses étudiées.

Tableau 9 : Les performances de notre modèle en termes de gain plateaux pour l'entreprise Suisse

Instance	NB Commandes	NB lignes de commandes	NB plateaux visités stratégie actuelle	NB plateaux visités notre approche	Gain	Temps de résolution (en seconds)
Jour 1	31	269	81	62	23%	9
Jour 2	45	148	53	39	26%	11
Jour 3	50	224	76	49	36%	30,59
Jour 4	60	307	100	73	27%	65,61
Jour 5	81	352	123	79	36%	3600*
Jour 6	103	329	115	67	42%	3600*
Jour 7	116	490	143	93	35%	3600*
Jour 8	145	364	137	74	46%	3600*
Jour 9	160	780	264	186	30%	3600*
Jour 10	169	502	187	124	34%	3600*
Jour 11	192	1134	204	148	27%	3600*

Tableau 10 : Les performances de notre modèle en termes de gain Ci pour l'entreprise Suisse

Instance	C_1 Stratégie actuelle	C_1 Notre approche	Gain C_1	C_2 Stratégie actuelle	C_2 Notre approche	Gain C_2	C_3 Stratégie actuelle	C_3 Notre approche	Gain C_3
Jour 1	81	62	23%	293	281	4%	104	87	16%
Jour 2	53	39	26%	162	154	5%	66,2	52,8	20%
Jour 3	76	49	35%	244	231	5%	97	71	27%
Jour 4	100	73	27%	332	315	5%	128	101	21%
Jour 5	123	79	36%	388	363	6%	154	112	27%
Jour 6	115	67	42%	368	342	7%	145	100	31%
Jour 7	143	93	35%	525	506	4%	188	143	24%
Jour 8	137	74	46%	424	383	10%	176	116	34%
Jour 9	264	186	30%	848	804	5%	335	262	22%
Jour 10	187	124	34%	553	524	5%	233	173	26%
Jour 11	204	148	27%	685	657	4%	264	213	19%

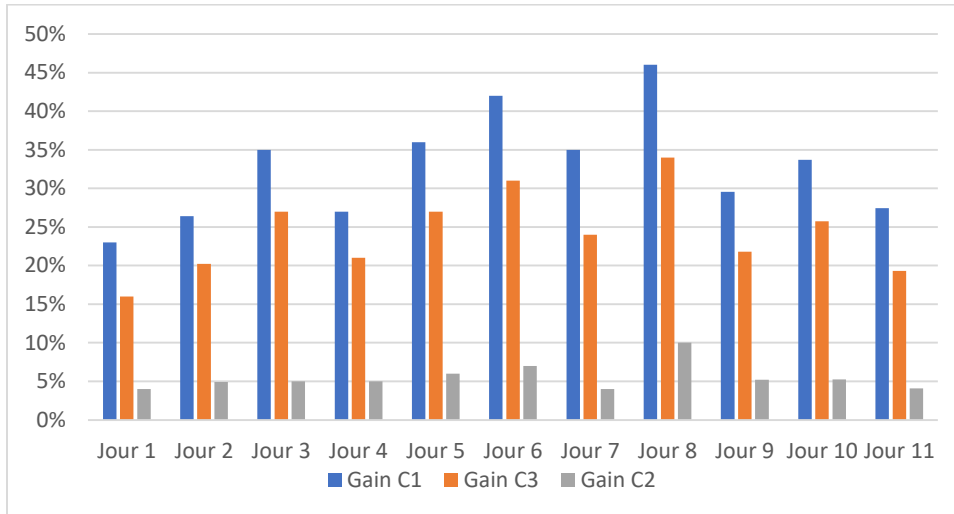


Figure 27 : Gains en temps de préparation pour les trois scénarii étudiés

Discussion générale des résultats

Nous observons que les performances obtenues par notre modèle de regroupement de commandes améliorent les performances de la stratégie actuelle en termes de critère choisi pour l'optimisation ainsi que pour le temps de préparation. De plus nous observons des gains conséquents que nous commentons au cours de cette section.

Nous notons que pour les grandes instances, CPLEX n'a pas pu prouver l'optimalité des résultats obtenus. Cela implique que des gains potentiellement plus importants peuvent être atteints. C'est notamment le cas pour l'entreprise Suisse. Nous observons sur le Tableau 9 un temps de résolution important quand le nombre de commandes est au-delà de 60. Pour ces instances nous stoppons le solveur CPLEX après qu'il ait effectué une heure de calcul comme expliqué plus tôt et nous retenons la dernière solution obtenue. Nous pouvons voir que malgré l'incertitude que nous avons sur l'optimalité de ses solutions, nous observons tout de même des gains significatifs concernant le critère d'optimisation et le temps de préparation de commandes.

Nous rappelons que des approches de type métaheuristiques sont testées en Section 4 de ce chapitre afin de résoudre efficacement les grandes instances où nous ne pouvons pas prouver l'optimalité avec CPLEX.

Effet du nombre de commandes

Tout comme pour la partie précédente considérant 1 VLM, nous allons étudier maintenant l'impact du nombre de commandes sur les performances de notre modèle de regroupement de commandes en lots. Pour chacune des entreprises, nous créons donc des sous-ensembles

d'instances en fonction de leur nombre de commandes. Tout comme le cas 1 VLM, nous utilisons la valeur médiane, permettant d'avoir un nombre (quasi) similaire d'instances dans chacun des sous-ensembles. Plus d'informations sur la méthode de création des sous-ensembles est disponible dans la partie précédente (Section 3.2).

Concernant l'Hôpital Français, le nombre d'instances étant pair, et la médiane étant de 48.5, nous incluons les instances ayant une valeur inférieure à la médiane dans le premier sous ensemble et les instances ayant une valeur supérieure à la médiane dans le second sous-ensemble. Concernant l'Entreprise Suisse, la médiane est de 103, et la moyenne est égale à 104, impliquant les deux sous-ensembles décrits tableau 11 ci-dessous.

Ces résultats ont la même tendance que ceux obtenus dans la partie un VLM. En ayant un plus grand nombre de commandes, nous observons des gains plus importants. En effet, accroître le nombre de commande augmente les possibilités d'avoir des commandes visitant les mêmes plateaux, et donc gagner en temps de préparation.

Tableau 11 : Effet du nombre de commandes sur les performances

Entreprise	Sous ensemble d'instances	Gain plateau moyen	Gain C_2 moyen
Hôpital Français	Instances ayant moins de 48.5 commandes	34%	27%
	Instances ayant plus de 48.5 commandes	46%	37%
Entreprise Suisse	Instances ayant 103 commandes ou moins	32%	24%
	Instances ayant plus de 103 commandes	32%	25%

Effet du nombre de lignes de commandes

Nous regardons maintenant l'effet du nombre de lignes de commandes par commande sur nos critères de comparaison. Pour l'Hôpital Français, le premier sous ensemble comprendra les instances ayant un nombre moyen de lignes de commandes par commande inférieur à 1.69. Les instances ayant un nombre moyen de lignes de commandes par commande supérieur à 1.69 sont incluses dans le second sous-ensemble. Concernant l'Entreprise Suisse, la médiane est de 4.34 et la moyenne est égal à 4.51. Les sous-ensembles créés ainsi que les résultats associés sont présentés dans le tableau 12 ci-dessous :

Tableau 12 : Effet du nombre de lignes de commandes par commande sur les performances

Entreprise	Sous ensemble d'instance	Gain plateau moyen	Gains C_2 moyen
Hôpital Français	Instance ayant une moyenne inférieure à 1.69 lignes par commandes	44%	35%
	Instance ayant une moyenne supérieure à 1.69 lignes par commandes	36%	29%
Entreprise Suisse	Instance ayant une moyenne inférieure ou égal à 4.34 lignes par commandes	37%	29%
	Instance ayant une moyenne supérieure à 4.34 lignes par commandes	29%	21%

Nous observons une adéquation entre ces résultats et ceux observés dans le cas 1 VLM. Un plus faible nombre de lignes de commandes par commande augmente le gain obtenu, et ceci, sur les deux critères de comparaison. L'explication donnée dans le cas 1 VLM est donc toujours valide : lorsque nous avons moins de lignes de commandes par commande, la probabilité d'avoir plusieurs commandes compatibles est plus importante. En d'autres mots il est plus plausible d'avoir des commandes visitant les mêmes plateaux des même VLMs.

Confirmation des résultats avec d'autres sous-ensembles

Nous avons montré que notre modèle de regroupement de commandes avait des performances dépendant du nombre de commandes, ainsi que du nombre de lignes de commandes par commande. Dans cette partie nous allons observer l'effet conjoint de ses paramètres sur notre modèle. Le nombre de commandes et de lignes de commandes par commande étant plus dispersé dans le cas de l'entreprise Suisse, nous nous focalisons dans cette section sur les résultats de cette entreprise (les résultats suivant la même tendance pour l'Hôpital Français).

Nous créons ainsi 4 sous-ensembles d'instances. Pour ce faire nous créons d'abord deux sous-ensembles en fonction du nombre de commandes par rapport à la valeur médiane. Ainsi deux sous ensemble sont créés en fonction du nombre de commande, puis nous scindons chacun de ses sous-ensembles en deux, en fonction du nombre de lignes de commandes. Les sous-ensembles sont présentés sur le tableau 13 ci-après. Les résultats confirment nos observations faites jusqu'ici. Nous observons que le sous ensemble apportant le meilleur résultat est celui ayant un grand nombre de commandes ainsi que peu de lignes par commande.

Tableau 13 : Résultats de l'influence des deux paramètres : Nombre de commandes et nombre moyen de lignes de commandes par commande

Instances ayant un nombre de commandes	Inférieur ou égal à 103		Supérieur à 103	
	Inférieur à 4.4	Supérieur à 4.4	Inférieur à 4.2	Supérieur ou égal à 4.2
Gain plateaux moyen	34.6%	29.6%	40%	30.7%
Gain C_2 moyen	26.1%	21.3%	30%	21.7%

3.4 Résolution par métaheuristique

Nous avons observé pour les grandes instances (i.e. celles ayant un nombre de commandes excédant 60), qu'un temps de calcul important est nécessaire. L'optimalité n'est pas prouvée par CPLEX pour ces instances. Nous rappelons que l'un de nos objectifs est de développer un outil de regroupement de commandes utilisable dans la pratique. Avoir une heure de calcul en début de journée pour un regroupement de commandes n'est bien évidemment pas possible industriellement. Nous devons garantir une bonne solution (proche optimal) sous un temps de calcul raisonnable. De plus, notre partenaire industriel voulant une solution spécifique codée dans son logiciel, nous retirons l'utilisation de logiciels commerciaux de notre panel de choix. Conformément aux préconisations de KLS Logistic, le temps de résolution ne peut pas dépasser 5 minutes. Ces arguments ont motivé notre choix d'implémentation d'une approche métaheuristique pouvant être utilisée par notre partenaire.

Nous avons tout d'abord testé et implémenté la métaheuristique Algorithme Génétique dans le logiciel de l'entreprise. Les premiers résultats étant intéressants, et il nous a semblé judicieux d'aller plus loin. Ainsi nous avons décidé, à l'aide d'un stage de master, d'élargir les pistes de recherche en testant des approches alternatives. Ainsi les métaheuristicques « Recherche tabou » et « Recuit simulé » sont testées et nous comparons les performances de l'ensemble des métaheuristicques testées.

Le choix des métaheuristicques testées provient de la littérature concernant le problème de regroupement des commandes utilisant un système traditionnel (Entrepôt manuel avec présence de rack sur lesquels sont entreposées les références), d'où notre choix. Plus d'informations concernant le regroupement de commandes en lots considérant un système de préparation de commandes traditionnel avec une résolution par métaheuristicques sont disponibles dans :

- Hsu et al (2005) et Tsai et al. (2008) pour l'Algorithme Génétique
- Li et Zhou (2013) pour la Recherche Tabou
- Matusiak et al. (2014) pour le Recuit Simulé

Les tests des deux dernières métaheuristicques furent possibles grâce à la collaboration avec une étudiante effectuant son projet master. Ainsi les paramétrages de la « Recherche Tabou » et du « Recuit Simulé » ont été effectués par Emna Laajili. Nous renvoyons le lecteur à Laajili et al. (2017) pour plus d'informations concernant ces deux dernières métaheuristicques. Le paramétrage des deux métaheuristicques testées durant cette thèse de master résulte de multiples essais

permettant de calibrer les différents paramètres de la Recherche Tabou et du Recuit Simulé. Nous gardons ici les paramètres amenant les meilleures performances pour chacune des deux métaheuristiques.

Au cours de cette section, nous allons décrire comment nous avons structuré puis paramétré chacune des métaheuristiques testées, avant de montrer la comparaison des performances de ces métaheuristiques par rapport aux résultats fournis par CPLEX.

3.4.1 Structure de la solution et paramétrage des métaheuristiques testées

Pour les trois métaheuristiques que nous avons testées, un choix du paramétrage a été fait et nous décrivons ici ces choix.

Tout d'abord la solution est encodée pour les trois métaheuristiques de la même façon. Une explication de l'encodage utilisé est montrée à travers un exemple (Cf. tableau. 14 ci-dessous). Dans cet exemple, 8 commandes doivent être regroupées, en ayant au maximum 3 commandes par lot. Ainsi un total de 3 lots sont constitués. Nous observons dans le tableau suivant que le lot 1 est constitué des commandes 3 et 4, le lot 2 est constitué des commandes 1, 5 et 6 et enfin le lot 3 est constitué des commandes 2, 7 et 8.

Tableau 14 : Codage de la Solution

Numéro de lot (solution encodée)	2	3	1	1	2	2	3	3
Numéro de commandes	1	2	3	4	5	6	7	8

Concernant le choix de la solution initiale, nous décidons de prendre la stratégie actuelle de regroupement de commandes de l'entreprise (celle-ci est décrite précédemment). Concernant le critère de fin des métaheuristiques, nous utilisons le temps de calcul, conformément à la volonté de notre partenaire industriel. Ainsi nous testons deux critères de fin, après 1 minute de calcul et après 5 minutes de calcul.

Avant de décrire les paramétrages intrinsèques à chacune des métaheuristiques testées, nous détaillons la Méthode SWAP utilisée par toutes nos métaheuristiques. Nous avons utilisé cette méthode pour l'algorithme Génétique lors des Mutations devant être faites sur une solution, ainsi que pour la recherche du voisinage des solutions pour la Recherche Tabou et le Recuit Simulé. Cette méthode est communément utilisée dans la littérature pour ces opérations, ce qui motive notre choix. Elle consiste à sélectionner aléatoirement deux numéros de commandes, et d'interchanger les numéros de lot dans lequel ces commandes sont incluses. Cette méthode a

l'avantage d'avoir une solution résultante faisable. Une solution est dite faisable si le nombre de commandes affecté à un lot n'excède pas le nombre maximal de commandes pouvant être attribué à un lot. Un exemple de la méthode SWAP est montré dans le tableau 15. Dans cet exemple les commandes 3 et 7 sont choisies aléatoirement. La solution résultante a les lots de ses 2 commandes interchangées :

Tableau 15 : Opérateur SWAP

Numéro de commande	1	2	3	4	5	6	7	8
Solution	2	3	1	1	2	2	3	3
Solution après le SWAP	2	3	3	1	2	2	1	3

Paramétrage de l'algorithme génétique

L'algorithme génétique est une métaheuristique inspirée de la nature. A partir d'une population de solutions, des opérations de croisement entre deux solutions ainsi que de mutation de chaque solution peuvent arriver, donnant une population plus importante de solutions. Cependant le nombre de solutions dans la population devant être constant d'une population à l'autre, un système de sélection de solutions est opéré. Au fur et à mesure des itérations, une amélioration (en termes de fonction objectif, appelée aussi « Fitness ») des individus de cette population se fait naturellement. Les informations nécessaires au paramétrage de cette métaheuristique sont données dans les paragraphes suivants.

Le nombre de solutions de la population initiale est égal à N (Nombre de commandes à regrouper). La première des solutions constituant la population initiale est la solution représentant la stratégie actuelle de regroupement de commandes. Les autres solutions constituant cette population sont des solutions aléatoires faisables.

La probabilité d'effectuer des croisements est de 70% comme classiquement recommandé. L'opérateur de croisement sert à générer deux solutions « enfants » à partir de deux solutions « parents ». Nous avons choisi d'effectuer un croisement à « un point ». Ce type de croisement consiste à sélectionner aléatoirement un numéro de commandes que nous nommons H. Par la suite, une solution enfant sera constituée des numéros de lots de la commande 1 à H de la première solution parent et des numéros de lots de la commande H+1 à N de la seconde solution parent. La seconde solution enfant est constituée des numéros de lots des commandes 1 à H de la seconde solution parent et des numéros de lots des commandes H+1 à N de la première solution parent. Un exemple de cet opérateur de croisement est disponible dans le tableau 16 ci-dessous.

Si le croisement de deux solutions amène à une solution infaisable, celle-ci est directement supprimée.

Tableau 16 : Opérateur de croisement à un point

Solution Parent 1	2	3	1	2	1	1	3	2
Solution Parent 2	1	3	2	2	2	3	1	3
↑								
Solution Enfant 1	2	3	1	2	2	3	1	3
Solution Enfant 2	1	3	2	2	1	1	3	2

La probabilité qu'une solution subisse une mutation est définie à 1% (valeur communément utilisée en pratique). L'opérateur de mutation utilisé est le SWAP défini précédemment.

Pour une population initiale donnée, après les opérations de croisement et de mutation sur la population initiale, nous devons définir les solutions constituant la prochaine population. Pour cela nous utilisons la méthode dite « Elitiste ». Cette méthode permet de garder les E solutions ayant les meilleures performances dans la population suivante, les $N-E$ restantes étant choisis aléatoirement parmi les solutions restantes. Nous choisissons $E = 1/3 * N$. Ainsi un tiers des solutions constituant notre population sont les solutions ayant les meilleures performances de la population précédente et deux tiers des solutions sont aléatoirement choisies parmi les solutions restantes de la précédente population.

Paramétrage de la Recherche Tabou

Concernant la recherche tabou, à partir d'une solution initiale, un voisinage de cette solution est créé permettant l'exploration de solutions voisines. Une liste dite « Tabou » d'un nombre fixe de solutions visitées précédemment est gardé. La meilleure solution du voisinage (celle ayant la fonction objectif la plus performante) n'étant pas sur la liste « Tabou » devient la nouvelle solution. Ainsi au fil des itérations successives, la solution s'améliore naturellement. Le paramétrage de cette métaheuristique est fait comme décrit ci-dessous :

Le nombre de solutions constituant le voisinage d'une solution est réglé à N (rappelons que N est le nombre de commandes du portefeuille à préparer). La méthode utilisée pour la recherche de voisinage est l'opérateur SWAP décrit précédemment. En d'autres mots, à partir d'une solution initiale, N solutions seront générées grâce à l'opérateur SWAP. La liste « Tabou » comprend $\frac{N}{2}$ solutions précédemment visitées. Si une solution générée dans le voisinage appartient à la liste

Tabou, elle est effacée du voisinage. La meilleure solution du voisinage créé est prise comme nouvelle solution.

Paramétrage du Recuit Simulé

Le Recuit Simulé est, tout comme la Recherche Tabou, une métaheuristique de recherche locale, permettant de visiter le voisinage d'une solution actuelle et de choisir une solution de ce voisinage comme nouvelle solution actuelle. Pour le recuit simulé, le fait de retenir une solution repose sur un paramètre appelé « Température ». Plus précisément, cette Température définit la probabilité d'accepter une solution détériorant les performances de la fonction objectif. Au cours des itérations cette température diminue, impliquant une probabilité plus faible d'accepter de « mauvaises solutions ». Cette métaheuristique est classique et de plus amples explications sur son fonctionnement sont disponibles dans Matusiak et al. (2014). Le paramétrage adopté de cette métaheuristique est fait comme décrit ci-dessous :

La température initiale est réglée à 0.1. Le facteur de décroissance (permettant d'abaisser la température d'une itération à l'autre) est de 0.9995. Le nombre de solutions constituant le voisinage est de $3 \cdot N$. La méthode utilisée pour la recherche locale est le SWAP décrit précédemment. Sachant que nous réglons le temps de calcul comme critère d'arrêt, nous n'avons pas besoin d'une température finale.

3.4.2 Test et comparaison des performances des métaheuristicques testées

Nous avons décrit rapidement au cours de la section précédente le choix des métaheuristicques ainsi que leurs calibrations, nous dédions cette section à la comparaison des métaheuristicques en termes de performances. Nous utilisons le critère retenu par notre modèle d'optimisation comme fonction objectif pour les métaheuristicques (la somme du nombre de plateaux visités par le VLM1 de chaque lot), permettant une comparaison avec les valeurs fournies par la stratégie actuelle et CPLEX.

De plus nous nous focalisons uniquement les instances de l'Entreprise Suisse car un grand nombre d'instances n'a pas été résolu à l'optimal malgré une heure de calcul de CPLEX. Nous comparons donc les résultats de nos métaheuristicques testées après 1 minute et 5 minutes de temps de calcul. Pour chaque instance, chaque métaheuristique est lancée 10 fois et la moyenne des 10 résultats obtenus est gardée, que nous appelons Valeur Fitness Moyenne (VFM). Nous utilisons les deux indicateurs décrits ci-dessous pour la comparaison.

- Le taux d'amélioration (TA) permettant de comparer les performances d'une métaheuristique donnée par rapport à la stratégie de regroupement de commandes actuelle de l'entreprise. En d'autres mots, un TA plus élevé implique une plus grande amélioration des performances par rapport à la stratégie actuelle de regroupement de commandes.

$$TA = \frac{\text{Critère stratégie actuelle} - \text{VFM}}{\text{Critère stratégie actuelle}}$$

- Le taux d'efficacité (TE) permettant de comparer les performances d'une métaheuristique donnée aux résultats fournis par CPLEX après une heure de calcul. Un TE supérieur à 100% montre que la métaheuristique fournit de meilleurs résultats que CPLEX (au bout d'une heure de calcul) pour une instance donnée.

$$TE = 1 - \frac{\text{VFM} - \text{Résultat CPLEX 1h de calcul}}{\text{Résultat CPLEX 1h de calcul}}$$

Les tableaux suivants présentent les résultats des différentes métaheuristicques pour les instances considérées, pour 1 minute de calcul (Tableau 17) et 5 minutes de calcul (Tableau 18). Si une métaheuristique apporte des résultats plus performants que les autres testées, nous la mettons en relief (**en gras**).

Tableau 17 : Récapitulatif des résultats des métaheuristicques testés (1 minute de calcul)

Instances	N	Critère approche actuelle	CPLEX	Recuit Simulé			Recherche Tabou			Algorithme Génétique		
				VFM	TA	TE	VFM	TA	TE	VFM	TA	TE
Jour 1	31	81	62	62	23,5%	100,0%	62	23,5%	100,0%	62	23,5%	100,0%
Jour 2	45	53	39	39	26,4%	100,0%	39	26,4%	100,0%	40	24,5%	97,4%
Jour 3	50	76	49	49,3	35,1%	99,4%	49	35,5%	100,0%	51	32,9%	95,9%
Jour 4	60	100	73	74	26,0%	98,6%	74	26,0%	98,6%	76	24,0%	95,9%
Jour 5	81	123	79	78,7	36,0%	100,4%	80	35,0%	98,7%	83	32,5%	94,9%
Jour 6	103	115	67	72,3	37,1%	92,1%	72,6	36,9%	91,6%	74	35,7%	89,6%
Jour 7	116	143	93	93	35,0%	100,0%	94,3	34,1%	98,6%	98	31,5%	94,6%
Jour 8	145	137	74	74	46,0%	100,0%	78	43,1%	94,6%	83	39,4%	87,8%
Jour 9	160	264	186	176,7	33,1%	105,0%	182,6	30,8%	101,8%	202,8	23,2%	91,0%
Jour 10	169	187	124	120,3	35,7%	103,0%	125,3	33,0%	99,0%	144,1	22,9%	83,8%
Jour 11	192	204	148	134,1	34,3%	109,4%	139	31,9%	106,1%	161,1	21,0%	91,1%
Moyenne					33,5%	100,7%		32,4%	99,0%		28,3%	92,9%

Tableau 18 : Récapitulatif des résultats des métaheuristiques testés (5 minutes de calcul)

Instances	N	Critère approche actuelle	CPLEX	Recuit Simulé			Recherche Tabou			Algorithme Génétique		
				VFM	TA	TE	VFM	TA	TE	VFM	TA	TE
Jour 1	31	81	62	62	23,5%	100,0%	62	23,5%	100,0%	62	23,5%	100,0%
Jour 2	45	53	39	39	26,4%	100,0%	39	26,4%	100,0%	39	26,4%	100,0%
Jour 3	50	76	49	49	35,5%	100,0%	49	35,5%	100,0%	51	32,9%	95,9%
Jour 4	60	100	73	73,7	26,3%	99,0%	73,3	26,7%	99,6%	76	24,0%	95,9%
Jour 5	81	123	79	77,7	36,8%	101,6%	78	36,6%	101,3%	83	32,5%	94,9%
Jour 6	103	115	67	69,3	39,7%	96,6%	71	38,3%	94,0%	74	35,7%	89,6%
Jour 7	116	143	93	91,7	35,9%	101,4%	93,3	34,8%	99,7%	98	31,5%	94,6%
Jour 8	145	137	74	72	47,4%	102,7%	73	46,7%	101,4%	83	39,4%	87,8%
Jour 9	160	264	186	174,3	34,0%	106,3%	176,8	33,0%	104,9%	193,9	26,6%	95,8%
Jour 10	169	187	124	117,4	37,2%	105,3%	121,7	34,9%	101,9%	141,7	24,2%	85,7%
Jour 11	192	204	148	130,1	36,2%	112,1%	133,6	34,5%	109,7%	155,4	23,8%	95,0%
Moyenne					34,5%	102,3%		33,7%	101,1%		29,1%	94,1%

Les différentes métaheuristiques, bien qu'améliorant toutes les performances par rapport à la stratégie actuelle, donnent des résultats bien différents. Nous représentons sur le tableau suivant les gains moyens en termes de TA et de TE pour une meilleure visibilité des résultats. Nous montrons aussi les min et max des valeurs TA et TE. A partir de ces résultats nous pouvons déduire que la métaheuristique apportant globalement les meilleurs résultats est le Recuit Simulé avec une moyenne de TA de 34.5% et de TE de 102.3% après 5 minutes d'exécution.

Tableau 19 : Récapitulatif des résultats des métaheuristiques

1 minute	TA			TE		
	Min	Max	Moyenne	Min	Max	Moyenne
Recuit simulé	23,5%	46,0%	33,5%	92,1%	109,4%	100,7%
Recherche tabou	23,5%	43,1%	32,4%	91,6%	106,1%	99,0%
Algorithme génétique	21,0%	39,4%	28,3%	83,8%	100,0%	92,9%

5 Minutes	TA			TE		
	Min	Max	Moyenne	Min	Max	Moyenne
Recuit simulé	23,5%	47,4%	34,5%	96,6%	112,1%	102,3%
Recherche tabou	23,5%	46,7%	33,7%	94,0%	109,7%	101,1%
Algorithme génétique	23,5%	39,4%	29,1%	85,7%	100,0%	94,1%

Nous observons que le recuit simulé améliore les performances par rapport à la stratégie actuelle de l'entreprise de 33.5% en moyenne au bout d'une minute de calcul et de 34.5% pour 5 minutes de calcul. De plus, une minute de calcul de cette métaheuristique suffit pour améliorer le résultat donné par CPLEX ; le taux d'efficacité au bout d'une minute est en moyenne de 100.7%. Ainsi après une minute de calcul, nous observons déjà une amélioration par rapport au résultat CPLEX obtenu après une heure pour les instances ayant un grand nombre de commandes. Ce taux d'efficacité

atteint même les 109.4% dans l'instance comportant le plus grand nombre de commandes à regrouper. Nous observons pour 5 minutes de calcul que ce taux d'efficacité est de 102.3% en moyenne et de 112,1% au maximum, justifiant l'intérêt de l'utilisation métaheuristique lorsqu'il y a un grand nombre de commandes.

De plus, les métaheuristicques testées donnent le résultat optimal fourni par CPLEX, pour la plupart des instances ayant un faible nombre de commandes. Nous observons même que la métaheuristique donnant les résultats les moins performants (l'algorithme génétique utilisant un paramétrage « standard ») donne des résultats très proches de celui de CPLEX pour la majorité des instances en une minute de calcul (avec un TE supérieur à 90% en moyenne). Enfin nous observons globalement que les résultats au bout de 5 minutes de calcul ne fournissent pas une amélioration significative par rapport à ceux obtenus au bout de 1 minute. C'est pourquoi nous préconisons une minute de calcul comme temps de calcul pour la métaheuristique.

Au cours de ce chapitre, nous avons évalué le temps de préparation de commandes lorsque les commandes sont regroupées en lots et que la collecte s'effectue dans une zone de préparation équipée de VLMS. A partir de cette évaluation, nous avons développé un modèle d'optimisation permettant d'affecter des commandes aux lots minimisant le temps de préparation de commandes. Enfin nous avons vu les limites de notre approche en termes de temps de résolution. Pour pallier à ce problème, nous avons développé et codé plusieurs métaheuristicques dont les résultats sont excellents. Le prochain chapitre se focalise sur le regroupement en lots lorsque la collecte s'effectue dans une zone de préparation équipée de Carousels

Chapitre 4 Regroupement de commandes en lots dans une zone de préparation équipée de Carousels

Dans ce chapitre, nous allons nous focaliser sur l'activité de regroupement de commandes en lots avec l'utilisation de stockeurs du type Carousel. Nous décrivons les spécificités de ce type de machine ainsi que leurs différences par rapport aux VLMs dans la section 1. La section 2 sera dédiée au cas d'un carousel tandis que la section 3 se focalisera sur le cas d'une préparation s'effectuant sur plusieurs Carousels. Pour chacune de ces deux sections, la définition des temps de préparation de commandes sera faite comme dans la partie VLMs, nous permettant d'établir un modèle d'optimisation que nous testerons sur des études de cas avec des données réelles d'entreprises. Enfin la section 4 présentera une métaheuristique intéressante dans le cas où la résolution exacte ne peut pas prouver l'optimalité des résultats en un temps raisonnable de calcul. Les travaux étudiés dans ce chapitre ont aussi été présentés dans Lenoble et al. (2017a).

4.1 Préparation de commandes avec des carousels

Cette section va énumérer les différences entre VLMs et Carousels. Tout d'abord les spécificités des stockeurs de type Carousel vont être décrites avant de brièvement expliquer comment la préparation de commandes utilisant des carousels est effectuée.

4.1.1 Spécificité du Carousel

Le Carousel, tout comme le VLM, est composé de plateaux sur lesquels sont rangés les références stockées. Cependant, à la différence du VLM où les plateaux n'ont pas de connexion entre eux, le Carousel est composé d'une chaîne effectuant une rotation verticale ou horizontale. Sur cette chaîne effectuant des rotations en boucle fermée sont accrochés les plateaux.

Deux familles de carousels existent : les carousels horizontaux dont les plateaux effectuent une rotation par rapport à un axe vertical, et les carousels verticaux ayant les plateaux effectuant une

rotation par rapport à un axe horizontal. Leur fonctionnement est le même et seul le type de gain de place recherché diffère. Dans le cas du carousel vertical, un gain de place grâce à l'utilisation de l'espace vertical est obtenu tandis que le carousel horizontal permet un gain de place grâce à l'utilisation de l'espace horizontal.

Historiquement les carousels pouvaient effectuer des rotations uniquement dans un seul sens, ce type de carousel étant appelé carousel Monodirectionnel. Cependant grâce à l'avancée de la technologie, les Carousels tournent actuellement dans les deux sens. Ces Carousels sont appelés Bidirectionnels. Nous considérons dans cette section uniquement des carousels Bidirectionnels verticaux (cf. Figure 28).

Lorsqu'un produit est demandé, le carousel effectue une rotation à partir du plateau actuellement présent au niveau de l'ouverture jusqu'au plateau demandé. Ainsi le temps pour passer d'un plateau i à un plateau j va dépendre du nombre de plateaux entre i et j .

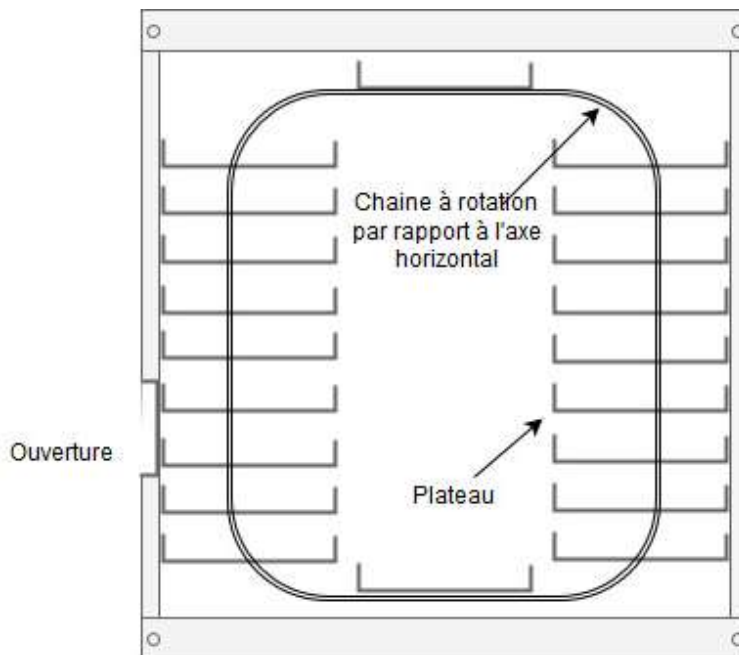


Figure 28 : Illustration d'un carousel vertical

Notons enfin que le temps pour passer d'un plateau d'un carousel au plateau suivant (du i au $i+1$) est très rapide dans la pratique. Ce temps est variable en fonction du type de machine et du constructeur mais n'excède jamais quelques secondes tandis que le temps pour passer d'un plateau à l'autre dans le cas des VLMs était beaucoup plus élevé, plusieurs dizaines de secondes.

4.1.2 Procédé de picking considérant un regroupement de commandes sur un ou plusieurs Carousels

Nous considérons donc un ou plusieurs Carousels sur lesquels doit s'effectuer l'opération de préparation de commandes. Les commandes sont regroupées en lots, permettant de collecter plusieurs commandes simultanément. Tout comme dans la partie VLM, nous considérons un seul préparateur de commandes en face d'une ou plusieurs machines. Nous décrivons ci-dessous la manière avec laquelle se passe l'opération de préparation de commandes avec l'utilisation de Carousels.

Pour chacun des lots, le préparateur de commandes reçoit un certain nombre de bacs (égal au nombre de commandes à collecter dans le lot). Un bac est associé à une commande et chaque fois que le préparateur collecte une ligne de commandes sur un plateau, il place cette quantité dans le bac associé à la commande. Le préparateur de commandes commence donc par la collecte des lignes de commandes positionnées sur le premier plateau du premier carousel à visiter. Pendant que ce carousel tourne jusqu'au prochain plateau à visiter, la collecte continue sur le carousel 2, ainsi de suite jusqu'au dernier carousel devant être visité. Une fois la première tournée des carousels effectuée, l'opérateur revient au Carousel 1 pour commencer la seconde tournée de plateaux à visiter. L'opérateur continue les tournées jusqu'à ce que le lot soit entièrement collecté, tout comme dans le cas VLMs. S'en suit la collecte de chacun des lots suivant le procédé décrit précédemment. Le processus est donc similaire à celui décrit dans la section VLM mais la différence essentielle est que le temps pour changer de plateaux en présence de carousels, n'est plus constant comme dans le cas VLM, mais dépend du nombre de plateaux entre les deux plateaux successivement visités, ce qui va expliquer le calcul du temps de préparation de commandes.

Ainsi le temps de préparation de commande sera divisé en deux temps, le temps de collecte des produits auquel s'ajoute d'éventuels temps d'attente devant les machines. Nous négligeons toujours le temps de marche du préparateur entre les différents carousels. Le procédé de préparation de commandes peut être décrit comme ci-dessous (Cf. Figure 29) avec Q carousels présents dans la zone de picking et P_l tournées de plateaux à visiter sur le lot l .

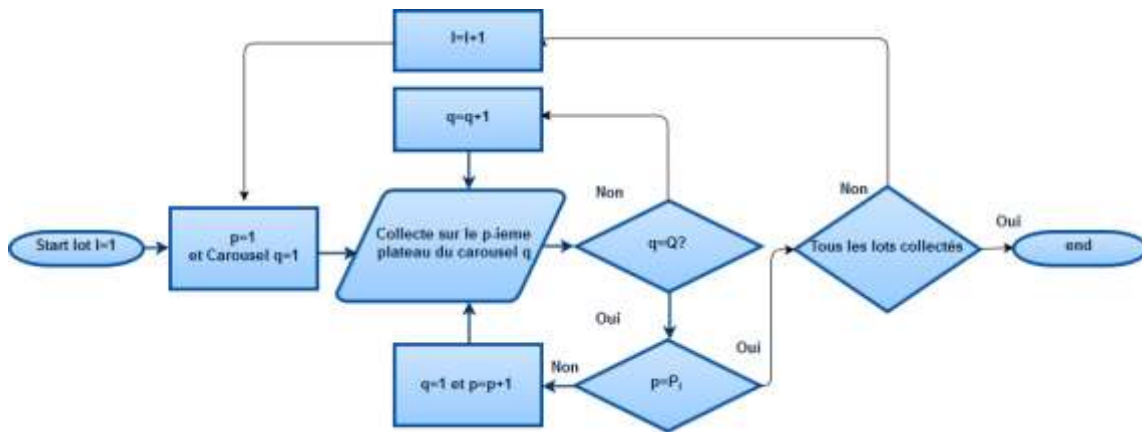


Figure 29 : Procédure de préparation de commandes regroupées en lots sur Q carousels

4.2 Regroupement de commandes dans un système de préparation de commandes utilisant 1 Carousel

Dans cette section nous considérons un préparateur devant effectuer la préparation sur 1 seul Carousel. Nous rappelons notre objectif principal, qui est de minimiser le temps de préparation de commandes pour la collecte d'un portefeuille de commandes. Minimiser le temps de préparation de commandes est l'objectif principal des entreprises opérant avec des carousels.

Nous allons suivre une démarche similaire à celle du chapitre VLM. Nous allons modéliser le temps de préparation de commandes, en présentant les notations et les hypothèses adoptées. Nous calculons et présentons par la suite les temps de préparation de commandes pour la collecte d'un portefeuille donné. Nous continuons par le développement d'un modèle d'optimisation permettant un regroupement de commandes en lots adapté au Carousel. En se basant sur des données réelles provenant de deux sociétés, nous testons alors notre modèle et comparons ses performances avec les méthodes de regroupement en lots utilisées dans les entreprises considérées.

4.2.1 Notations et hypothèses

Nous récapitulons ici les notations utilisées pour la définition du temps de préparation dans le cas carousel. Nous gardons les notations faites au cours du chapitre précédent que nous rappelons ci-dessous :

- N : le nombre de commandes clients à regrouper (taille du portefeuille de commandes à collecter).
- $\mathbb{N} = \{1, \dots, N\}$: l'ensemble des commandes clients à regrouper.
- S : la taille maximale des lots (i.e. le nombre maximum de commandes pouvant être attribué à un lot). Dans la pratique ce nombre est constant et dépend de la place disponible pour accueillir les bacs en face du préparateur de commandes.
- B : Nombre de lots de commandes à créer. En considérant le nombre de commandes et la taille de lots, nous calculons $B = \left\lceil \frac{N}{S} \right\rceil$ avec $\lceil \cdot \rceil$ la fonction borne supérieure.
- $\mathcal{B} = \{1, \dots, B\}$: Ensemble des lots de commandes.

- K : Nombre total de plateaux présents au sein du Carousel.
- $\mathcal{K} = \{1, \dots, K\}$: Ensemble des plateaux du Carousel.
- K^b : Nombre de plateaux devant être visités au cours du lot b .
- W : Temps total d'attente du préparateur (Somme de tous les temps où le préparateur est disponible pour la collecte mais le plateau sur lequel il doit collecter ne l'est pas).
- P : Temps total de picking, correspondant au temps durant lequel le préparateur collecte les références sur les plateaux et les pose dans les bacs.
- C : Temps total de préparation de commandes, $C = W + P$.
- W^b : Temps d'attente associé au lot b .
- P^b : Temps de picking associé au lot b .
- C^b : Temps de préparation de commandes associé au lot b : $C^b = W^b + P^b$.
- W_l^b : Temps d'attente de l'opérateur devant le l -ième plateau pour le lot b .
- P_l^b : Temps de picking pour la collecte des références situées sur le l -ième plateaux à visiter du lot b .
- C_l^b : Temps de préparation de commandes pour le l -ième plateau à visiter du lot b , $C_l^b = W_l^b + P_l^b$.
- L_l^b : Nombre de lignes de commandes devant être collectées sur le l -ième plateau devant être visité par le lot b .
- p : Temps de collecte d'une ligne de commandes sur un plateau quand celui-ci est disponible. De plus amples informations concernant ce temps sont disponibles dans la partie VLM.

En plus de ces notations, nous introduisons les nouvelles notations suivantes :

- K_l^b : Nombre de plateaux entre le $(l - 1)$ ième plateau et le l -ième plateau devant être visités par le lot b .

- r : Temps requis par le carrousel pour effectuer la rotation d'un plateau au plateau suivant (du i au $i+1$). Ce temps pour passer d'un plateau à celui d'après (ou d'avant comme nous considérons un carrousel bidirectionnel) est considéré constant.

Soit Q^b le vecteur de taille K^b comprenant les numéros des plateaux visités par le lot b dans l'ordre croissant. Plus précisément nous dénoterons $Q^b(1)$ la première composante du vecteur qui est égale au plus petit numéro du plateau visité pour le lot b . De même $Q^b(K^b)$ est la dernière composante du vecteur correspondant au plus grand numéro du plateau visité par le lot b . On a donc la relation suivante : $Q^b(1) < Q^b(2) < \dots < Q^b(K^b)$. Il est important de noter la différence entre K^b et Q^b . Le premier est une valeur indiquant le nombre de plateaux à voir pour un lot, tandis que le second est un vecteur comprenant l'ensemble des indices des plateaux à voir pour un lot.

Les hypothèses que nous utilisons sont les suivantes :

Nous considérons ici que chaque lot commence avec le premier plateau disponible. Nous faisons cette hypothèse car le temps de rotation est généralement faible et que le temps entre 2 lots est considéré suffisamment grand pour que le carrousel tourne jusqu'au plateau désiré. Cette hypothèse est d'autant plus vraie que nous considérons le carrousel bidirectionnel. Nous considérons donc que, pour chacun des lots, le premier plateau est directement disponible pour la collecte $K_1^b = 0 \forall b \in \mathcal{B}$.

De plus, nous considérons aussi que le temps pour passer d'un plateau à l'autre est proportionnel au nombre de plateaux intermédiaires. Nous négligeons donc l'accélération du carrousel. Cette hypothèse est réaliste d'après notre partenaire industriel. Ainsi le temps entre le $(l-1)$ et le l -ième plateau à visiter pour le lot b sera $K_l^b * r$.

Nous étudions le cas réaliste où le temps de collecte sur le l -ième plateau à visiter dépend du nombre de lignes de commandes à collecter sur ce plateau, donnant la relation suivante : $P_l^b = L_l^b * p, \forall b \in \mathcal{B}$ et $\forall l \in \{1, \dots, K^b\}$.

Nous rappelons enfin que le temps de préparation de commandes C est divisé en un temps d'attente et un temps de collecte, dont la relation est donnée par :

$$C = W + P = \sum_{b=1}^B (W^b + P^b) = \sum_{b=1}^B \sum_{l=1}^{K^b} (W_l^b + P_l^b) \quad (4-1)$$

Notons que par la suite, nous nous intéresserons à la distance parcourue par le carousel. La distance entre deux plateaux successifs étant constante, la distance parcourue entre deux plateaux visités est directement proportionnelle au nombre de plateaux intermédiaires et cette distance sera exprimée en ce nombre de plateaux intermédiaires.

4.2.2 Evaluation du temps de préparation de commandes sur 1 Carousel

Nous utilisons les notations ainsi que les hypothèses faites dans la section précédente afin d'évaluer le temps de préparation de commandes.

Proposition 4 : Dans un système de préparation de commandes équipé d'un Carousel, minimiser le temps de préparation de commandes est équivalent à minimiser la distance parcourue par le Carousel afin de collecter les différents lots. Lorsque le temps de collecte sur un plateau est proportionnel au nombre de lignes de commandes à collecter sur ce plateau, le temps de préparation de commandes est donné par : $C = r * \sum_{b=1}^B \sum_{l=1}^{K^b} K_l^b + p * \sum_{b=1}^B \sum_{l=1}^{K^b} L_l^b$

Nous présentons un exemple avant de démontrer la proposition 4.

Exemple 1. Temps de préparation de commandes considérant un temps de collecte sur un plateau proportionnel au nombre de lignes à collecter sur ce plateau

Nous considérons un lot b ayant 3 plateaux à visiter sur le carousel. Le nombre de lignes de commande à collecter étant : $L_1^b = 2$, $L_2^b = 1$ et $L_3^b = 2$. Nous rappelons que L_l^b est le nombre de lignes de commandes ayant besoin d'être collectées sur le l -ième plateau à visiter pour le lot b . La figure 30 ci-dessous illustre ce cas. Nous considérons $K_2^b = 3$ et $K_3^b = 2$.

Le préparateur de commandes commence par la collecte sur le premier plateau à visiter, prenant $P_1^b = 2 * p$ unités de temps. S'en suit l'attente en face du carousel jusqu'à ce que le second plateau devienne disponible, prenant $3 * r$ unités de temps (car $K_2^b = 3$). Une fois que ce plateau est disponible, il collecte la ligne de commandes prenant $P_2^b = p$ unités de temps. Après cela le carousel effectue une rotation jusqu'au troisième plateau à visiter, requérant $2 * r$ unités de temps ($K_3^b = 2$). L'opérateur collecte sur ce dernier, prenant $P_3^b = 2 * p$ unités de temps. Le temps de préparation de commandes de ce lot b est donc $C^b = W^b + P^b = \sum_{l=1}^{K^b} (W_l^b + P_l^b) = 2 * p + 3 * r + p + 2 * r + 2 * p = 5 * r + 5 * p$.

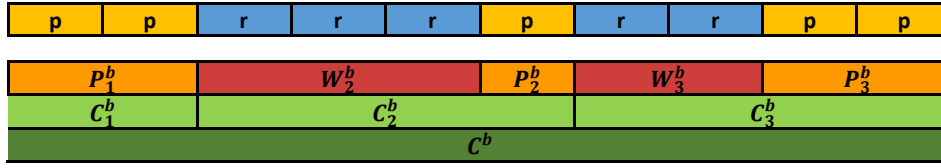


Figure 30 : Illustration du temps de préparation de commandes lorsque le temps de picking sur un plateau est considéré proportionnel au nombre de lignes à collecter sur ce même plateau

Preuve de la proposition 4

Nous rappelons tout d'abord que le temps de préparation de commandes est défini par $C = \sum_{b=1}^B C^b$ et $C^b = W^b + P^b$.

Le temps d'attente correspond au temps pendant lequel le carousel effectue des rotations, donnant la relation suivante pour un lot b donné : $W^b = \sum_{l=1}^{K^b} W_l^b = \sum_{l=1}^{K^b} K_l^b * r$. Le temps de collecte est défini précédemment : $P^b = \sum_{l=1}^{K^b} P_l^b = \sum_{l=1}^{K^b} L_l^b * p$.

Ainsi le temps de préparation pour l'ensemble des lots est donné par $C = \sum_{b=1}^B \left(\sum_{l=1}^{K^b} K_l^b * r + \sum_{l=1}^{K^b} L_l^b * p \right) = r * \sum_{b=1}^B \left(\sum_{l=1}^{K^b} K_l^b \right) + p * \sum_{b=1}^B \left(\sum_{l=1}^{K^b} L_l^b \right)$ sachant que r et p sont des constantes. Nous rappelons que le nombre total de lignes de commandes à collecter sur l'ensemble des lots, $\sum_{b=1}^B \sum_{l=1}^{K^b} L_l^b$, est une valeur constante ne dépendant pas du regroupement en lots choisis.

Minimiser le temps de préparation de commandes est donc équivalent à minimiser la distance parcourue par le carousel sur l'ensemble des lots à collecter $\left(\sum_{b=1}^B \left(\sum_{l=1}^{K^b} K_l^b \right) \right)$. ■

4.2.3 Modèle d'optimisation et étude de cas

Dans cette section, nous développons le modèle d'optimisation permettant un regroupement de commandes en lots lorsque la collecte s'effectue sur un Carousel. La seconde partie de cette section est dédiée à l'étude de cas ainsi qu'à la discussion des résultats obtenus.

4.2.3.1 Modèle d'optimisation

La principale décision de notre modèle est la construction de différents lots à partir d'un portefeuille de commandes donné. Les variables de décision sont :

- $x_{n,b}=1$ si la commande n est incluse dans le lot b ; 0 sinon.

- $T_{kb} = 1$ si le plateau k est visité durant le lot b ; 0 sinon. Par définition le nombre de plateaux devant être visité par le carrousel pour collecter le lot b est donné par $K^b = \sum_{k=1}^K T_{kb}$.

Nous considérons U_{kn} le paramètre d'entrée égal à 1 quand la commande n contient une référence devant visiter le plateau k , 0 sinon.

Soit D^b la distance minimale parcourue par le carrousel afin de collecter les références sur tous les plateaux devant être visités pour le lot b . Si un seul plateau doit être visité pour un lot, la distance parcourue par le carrousel sera considérée nulle pour ce lot. Rappelons que l'on suppose le carrousel positionné au premier plateau visité lorsque l'opérateur débute la collecte du lot. Dans le cas de plusieurs plateaux à visiter pour un lot donné, minimiser la distance parcourue par le carrousel est équivalent à trouver la distance minimale dans un cercle devant visiter une série de points. Soit D_k^b la distance entre le plateau k et le plus proche plateau de k devant être visité et ayant un numéro de plateau inférieur à k .

Pour le calcul de la distance, prenons l'exemple du cercle ci-dessous (Figure 31) devant visiter trois points (ici nous les nommons i , j et f), avec les distances suivantes inter-points $A = [i; j]$; $B = [j; f]$; $C = [f; i]$. Nous voyons que la distance minimale visitant ces 3 points est $A+B$, qui est aussi égale à la circonférence du cercle ($A+B+C$) moins la plus grande distance entre deux points consécutifs (ici la plus grande distance est C).

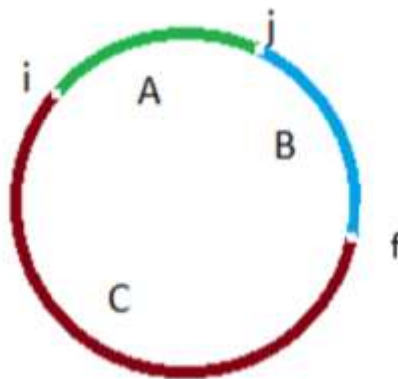


Figure 31 : Distance minimale dans un cercle

Nous utilisons ce résultat pour calculer la distance minimale à parcourir pour visiter les plateaux du carrousel pour le lot considéré. D'un côté nous prenons la distance totale du carrousel (égal à K), d'un autre côté nous cherchons la plus grande distance entre deux plateaux vus consécutivement

$(\max_{k \in \mathcal{K}} D_k^b)$. La distance minimale nécessaire pour visiter tous les plateaux pour un lot b sera $D^b = K - (\max_{k \in \mathcal{K}} D_k^b)$.

Rappelons aussi que la distance inter-lot n'est pas prise en compte ici. Nous considérons que cette distance est parcourue en temps masqué pendant que de nouveaux bacs arrivent dans la zone de préparation. Cette hypothèse est réaliste car nous considérons le carousel bidirectionnel, le carousel aura donc deux possibilités pour rejoindre le premier plateau à visiter.

Ainsi la fonction objectif de notre modèle d'optimisation prenant en compte un Carousel est donnée par :

$$\text{Minimiser } \sum_{b=1}^B D^b$$

Les contraintes de notre modèle sont :

$$\sum_{b=1}^B x_{nb} = 1, \forall n \in \mathbb{N} \quad (4-1)$$

$$\sum_{n=1}^N x_{nb} \leq S, \forall b \in \mathcal{B} \quad (4-2)$$

$$T_{kb} = \sum_{n=1}^N x_{nb} * U_{kn}, \forall b \in \mathcal{B}, \forall k \in \mathcal{K} \quad (4-3)$$

$$D^b = K - \max_{l \in \{1, \dots, K^b\}} D_l^b, \forall b \in \mathcal{B} \quad (4-4)$$

$$\text{Si } K^b = \sum_{k=1}^k T_{kb} = 1 \rightarrow D_{Q^b(1)}^b = 0, \forall b \in \mathcal{B} \quad (4-5)$$

$$\text{Sinon } K^b = \sum_{k=1}^k T_{kb} > 1, \forall b \in \mathcal{B}, \forall k \in \mathcal{K}$$

$$D_{Q^b(l)}^b = Q^b(l-1) - Q^b(l) \quad \forall l \in \{2, \dots, K^b\} \quad (4-6)$$

$$D_{Q^b(1)}^b = K - Q^b(K^b) + Q^b(1) \quad (4-7)$$

La contrainte (4-1) garantit que chaque commande client est incluse dans un lot et un seul. La contrainte (4-2) impose que le nombre maximal de commandes par lots n'excède pas S . La contrainte (4-3) permet le calcul de T_{kb} . En effet un lot doit visiter un plateau si au moins une des commandes incluses dans ce lot doit visiter ce plateau. La contrainte (4-4) permet le calcul de la distance comme décrit précédemment. Si un seul plateau du carousel est visité pour un lot b , alors la distance parcourue sera nulle sur ce carousel (contrainte (4-5)). Si plusieurs plateaux doivent être vus sur un carousel, nous calculons la distance entre chacun des niveaux consécutivement visités sur ce carousel (contraintes (4-6) et (4-7)).

4.2.3.2 *Etude de cas avec des données réelles*

Comme dans le cas VLM, nous avons pu faire ces expériences grâce à notre collaboration avec l'entreprise KLS Logistic Systems. Précisons que les clients collaborant avec notre entreprise ont généralement plus d'un carrousel dans leurs zones de préparation.

Pour pallier à cette problématique, nous avons utilisé les données de 2 carousels consolidées sur un seul carrousel, ce qui permet également de garder l'intégralité des commandes. Ainsi si une commande doit visiter une ligne sur le carrousel 1 au plateau i et une ligne sur le carrousel 2 au plateau j , nous considérons que cette commande visite 2 lignes (une sur le plateau i et une sur le plateau j) dans l'unique carrousel. Nous considérons les données d'un Hôpital en France (n'étant pas le même que celui de l'étude de cas VLM) que nous nommerons, par souci de confidentialité Hôpital Français 2.

Concernant la stratégie actuelle de regroupement de commandes, celle-ci est la même que celle expliqué dans le chapitre précédent (Chapitre 3 Regroupement de commandes en lots dans une zone de préparation équipée de VLMs).

Nous utilisons les deux critères de comparaison suivant : le gain en distance parcourue (critère d'optimisation retenu) et gain en temps de préparation.

La distance parcourue est importante pour les entreprises car elle affecte la durée de vie de la machine. Une plus grande distance parcourue implique une plus faible durée de vie de la machine.

$$\text{Gain Distance} = \frac{\text{Distance parcourue stratégie actuelle} - \text{Distance parcourue notre méthode}}{\text{Distance parcourue stratégie actuelle}}$$

Dans notre comparaison concernant le second critère, le temps de préparation de commandes, nous supposons que le temps pour aller d'un plateau au suivant est unitaire, $r=1$. D'après KLS Logistic, le temps de collecte d'une ligne de commandes dans les entrepôts représente « environ » le temps pour aller d'un plateau au suivant, motivant notre choix d'avoir $p=1$.

$$\text{Gain } C = \frac{C \text{ avec la stratégie actuelle} - C \text{ avec la notre méthode}}{C \text{ avec la stratégie actuelle}}$$

Nous avons codé puis résolu le problème d'optimisation décrit dans la section 2.3.1 avec CPLEX version 12.6.3.0 sur un ordinateur possédant une processeur Intel Core i3 dual 2.3 GHz avec 3Go de mémoire vive et fonctionnant sur l'environnement Windows 7 Professionnel 32 bits.

Etude de cas : Hôpital Français 2

Le tableau 20 ci-dessous permet la comparaison entre la stratégie actuelle de regroupement de commandes en lots de l'Hôpital Français 2 et notre modèle en termes de distance parcourue par le carousel sur l'ensemble des lots. Conformément au mode opératoire de l'Hôpital Français 2, le nombre maximal de commandes pouvant être traitées simultanément est de 10, impliquant $S = 10$. Le carousel dispose de 43 plateaux, donc $K = 43$.

Chacune des instances décrites ci-dessous correspond à une journée et représente les commandes clients à préparer durant cette journée. Nous indiquons le nombre de commandes pour chacune des instances ainsi que le nombre de lignes de commandes. Ce tableau récapitule les valeurs du critère utilisé pour la stratégie actuelle (Distance parcourue stratégie actuelle) et pour notre approche (Distance parcourue notre méthode). Enfin, ce tableau indique le gain C comme défini précédemment ainsi que le temps de résolution mis par CPLEX pour trouver la solution optimale pour notre approche.

Nous avons arrêté CPLEX au bout de 30 minutes de calcul, prenant la meilleure solution retenue si jamais l'optimalité n'était pas prouvée. Nous désignons ces instances par « 1800* » dans la colonne temps de résolution.

Tableau 20 : Les performances de notre modèle pour l'Hôpital français 2

Instance	NB Commandes	NB lignes de commandes	Distance stratégie actuelle	Distance notre approche	Gain distance	Gain C	Temps de résolution (en seconds)
Jour 1	31	234	132	100	24%	9%	163
Jour 2	32	101	138	84	39%	23%	184
Jour 3	46	128	186	120	35%	21%	1722
Jour 4	50	243	196	122	38%	17%	1800*
Jour 5	72	372	284	136	52%	23%	1800*

Ci-dessous une représentation graphique de ces gains est fournie Figure 32 :

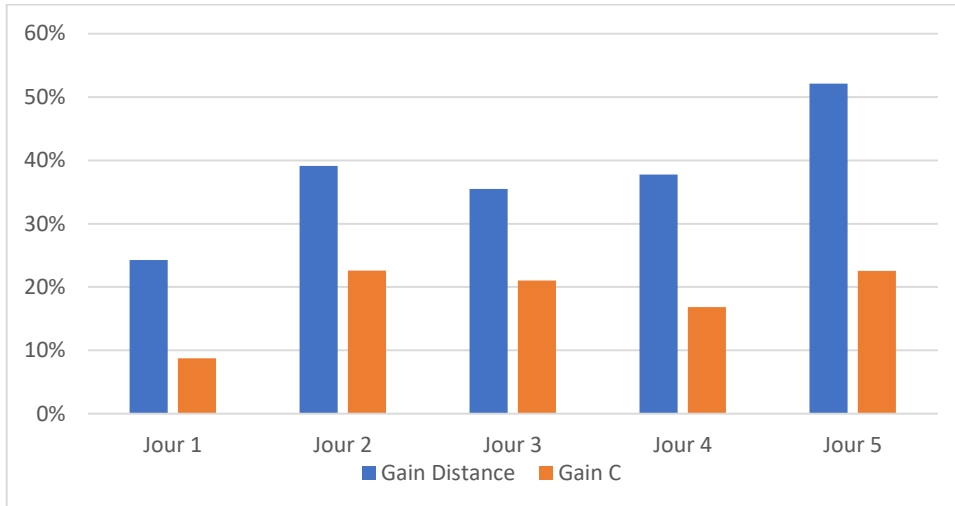


Figure 32 : Représentation graphique des Gains Distances et Gains C pour les instances de l'Hopital Francais 2

Discussion générale sur les résultats

Les résultats du tableau 20 permettent de comparer les gains obtenus sur les différentes instances considérées à la fois sur le critère et sur le temps de préparation.

Concernant ces gains, nous pouvons observer que le Gain Distance varie de 24% à 52%, avec une moyenne de 38%. Ces gains montrent que la machine tourne bien moins pour collecter le même nombre de lignes de commandes. Concernant le gain en temps de préparation, celui-ci varie de 9% à 23%, avec une moyenne de 18%.

4.3 Regroupement de commandes dans un système de préparation de commandes utilisant plusieurs Carousels

Nous considérons tout au long de cette section le cas d'une zone de préparation constituée de plusieurs Carousels, sur lesquels un préparateur de commandes effectue la collecte des références. Comme nous l'avons expliqué précédemment, l'avantage d'utiliser plusieurs carousels est la recherche de plateaux en temps masqué, réduisant les temps d'attente en face des machines. Dans cette section, nous utilisons la même méthodologie que celle des parties précédentes. Tout d'abord nous définissons les temps de préparation de commandes afin d'identifier un critère d'optimisation. Par la suite, nous développons un modèle d'optimisation et conduisons des expériences numériques, utilisant des données réelles afin d'observer les gains de notre méthode. Enfin nous discutons les résultats.

4.3.1 Notations et hypothèses utilisées

Nous gardons et adaptons les hypothèses utilisées dans la partie précédente. Nous ajoutons, aux notations présentées section 4.2.1, les notations suivantes :

- U : Nombre de Carousels considérés.
- $\mathcal{U} = \{1, \dots, U\}$: ensemble des Carousels.
- K_u : Nombre de plateaux du Carousel u .
- $\mathcal{K}_u = \{1, \dots, K_u\}$: Ensemble des plateaux du Carousel u .
- K_u^b : Nombre de plateaux devant être visités par le lot b sur le Carousel u
- W_{lu}^b : Temps d'attente devant le l -ième plateau sur le Carousel u pour le lot b . Nous avons la relation suivante : $W_l^b = \sum_{u=1}^U W_{lu}^b$.
- P_{lu}^b : Temps de collecte des références sur le l -ième plateau à visiter sur le Carousel u pour le lot b . Nous avons la relation suivante : $P_l^b = \sum_{u=1}^U P_{lu}^b$.
- L_{lu}^b : Nombre de lignes de commandes devant être collectées sur le l -ième plateau par le lot b sur le Carousel u .
- K_{lu}^b : Nombre de plateaux entre le $(l - 1)$ ième plateau et le l -ième plateau du carousel u pour le lot b .

$-Q_u^b$ le vecteur de taille K_u^b comprenant les numéros des plateaux visités du carrousel u au cours du lot b dans l'ordre croissant. Plus précisément nous dénoterons $Q_u^b(1)$ la première composante du vecteur étant égal au plateau visité pour le lot b ayant le numéro le plus faible. De même $Q_u^b(K_u^b)$ est la dernière composante du vecteur correspondant au plus grand numéro de plateau visité par le lot b sur le carrousel u . On aura donc la relation suivante : $Q_u^b(1) < Q_u^b(2) < \dots < Q_u^b(K_u^b)$.

Nous énonçons maintenant les différentes hypothèses que nous prenons en compte :

Nous considérons que chacun des carousels a son premier plateau disponible pour la collecte.

Ainsi $K_{1u}^b = 0 \forall u \in \mathfrak{U}$ et $\forall b \in \mathcal{B}$.

Nous généralisons aussi l'hypothèse concernant le temps de rotation dépendant uniquement du nombre de niveaux parcourus pour chacun des carousels. Ainsi le temps de rotation requis pour atteindre le l ième plateau du carrousel u est égal à $K_{lu}^b * r$.

Tout comme pour la partie VLM, nous ajoutons les hypothèses suivantes :

- Les carousels sont visités par ordre décroissant du nombre de plateaux à visiter pour chacun des lots. L'opérateur passe de carrousel en carrousel dans cet ordre. Nous renommons les carousels en fonction de cet ordre de passage pour chacun des lots. Conformément aux notations précédemment décrites, nous avons : $K_1^b \geq K_2^b \geq \dots \geq K_U^b \forall b \in \mathcal{B}$. Il est cependant important de noter que le Carrousel 1 d'un lot ne sera pas obligatoirement le Carrousel 1 du lot suivant.
- Le temps de marche entre la zone où les bacs sont disposés et les divers Carousels est à nouveau négligé, hypothèse réaliste au vu de la faible distance à parcourir entre les Carousels et les bacs.

Finalement, nous rappelons que nous étudions uniquement le cas réaliste où le temps de collecte sur un plateau dépend du nombre de lignes à collecter sur ce plateau.

4.3.2 Evaluation du temps de préparation de commandes sur plusieurs Carousels

Le but de cette section est d'évaluer le temps de préparation de commandes C associé à la collecte sur les U Carousels de B lots de commandes. Nous énonçons tout d'abord la proposition 5 puis, à travers un exemple, nous montrons comment nous calculons le temps de préparation de commandes avant de présenter les expériences numériques.

Proposition 5. Dans le cas de plusieurs Carousels, avec le temps de collecte par plateau proportionnel au nombre de lignes à collecter sur ce plateau, le temps de préparation de commande est donné par :

$$C = \sum_{b=1}^B \sum_{l=1}^{K_1^b} \sum_{u=1}^U (P_{lu}^b + W_{lu}^b), \text{ avec :}$$

$$W_{lu}^b = \max(K_{lu}^b * r - (\sum_{i=1}^{u-1} (L_{li}^b * p + W_{li}^b)) + \sum_{i=u+1}^U (L_{l-1i}^b * p + W_{l-1i}^b)); 0) \quad \forall 1 < u < U,$$

$$W_{l1}^b = \max(K_{l1}^b * r - \sum_{i=2}^U (L_{l-1i}^b * p + W_{l-1i}^b); 0),$$

$$W_{lU}^b = \max(K_{lU}^b * r - \sum_{i=1}^{U-1} (L_{li}^b * p + W_{li}^b); 0),$$

$$W_{lu}^b = 0 \text{ Si } l = 1 \text{ comme nous considérons } K_{1u}^b = 0$$

$$\text{Et } P_{lu}^b = L_{lu}^b * p.$$

Exemple 3 : Temps de préparation de commandes sur U carousels considérant un temps de collecte par plateau proportionnel au nombre de lignes à collecter.

Dans cet exemple, nous considérons 3 Carousels et les caractéristiques suivantes concernant la collecte du lot b :

$$K_1^b = 5 ; L_{11}^b=1 ; L_{21}^b=1 ; L_{31}^b=2 ; L_{41}^b=1 ; L_{51}^b=1 ; K_{21}^b=2 ; K_{31}^b=5 ; K_{41}^b=2 ; K_{51}^b=3.$$

$$K_2^b = 3 ; L_{12}^b=2 ; L_{22}^b=1 ; L_{32}^b=2 ; K_{22}^b=4 ; K_{32}^b=5.$$

$$K_3^b = 2 ; L_{13}^b=1 ; L_{23}^b=1 ; K_{21}^b=2.$$

La figure 33 ci-dessous illustre les différents temps nous permettant de calculer C .

p	r	r		p	r	r	r	r	r	p	p	r	r		p	r	r	r	p	
	p	p	r	r	r	r	p	r	r	r	r	r	p	p						
			p	r	r			p												
$P_1^b = P_{11}^b + P_{12}^b + P_{13}^b$				P_{21}^b		W_{22}^b		P_{22}^b	P_{23}^b	W_{31}^b		P_{31}^b	W_{32}^b		P_{32}^b	P_{41}^b		W_{51}^b		P_{51}^b
C_1^b				C_2^b						C_3^b						C_4^b		C_5^b		
C^b																				

Figure 33 : Illustration du temps de préparation de commandes lorsque le temps de picking sur un plateau est considéré proportionnel au nombre de lignes à collecter sur ce même plateau ($r = p$)

La préparation commence par un temps de collecte sur les Carousels, ces derniers ayant déjà leur premier plateau présent pour la collecte. Ainsi les différentes lignes du premier plateau présenté sur les différents Carousels seront collectées. Nous rappelons qu'une fois la collecte faite sur un carrousel, celui-ci se met à tourner pour atteindre le prochain plateau à visiter. Le temps de collecte lié au premier plateau à visiter est $C_1^b = P_1^b = P_{11}^b + P_{12}^b + P_{13}^b = (L_{11}^b + L_{12}^b + L_{13}^b) * p = 4 * p$.

L'opérateur retourne au premier carrousel, le second plateau étant déjà présent dans notre exemple. Mais un possible temps d'attente est présent si le temps que le carrousel met pour passer du premier au second plateau à visiter est plus grand que le temps que le préparateur passe sur les autres carousels. Sachant que $K_{21}^b = 2 * r$ et le temps passé sur les autres carousels est de $P_{12}^b + P_{13}^b = 3 * p$, un temps d'attente existe si $(2 * r) - (3 * p)$ est positif. Ainsi $W_{21}^b = \max(K_{21}^b * r - (P_{12}^b + P_{13}^b); 0)$. Une fois la collecte sur le second plateau du premier carrousel faite (prenant $P_{21}^b = p$ unités de temps sachant que $L_{21}^b = 1$), l'opérateur va devant le second carrousel. Un possible temps d'attente apparait si le temps de rotation du carrousel 2 ($K_{22}^b * r$) dépasse le temps que l'opérateur a passé sur le carrousel 3 pour la visite du premier plateau (P_{13}^b) et le temps sur le carrousel 1 pour la visite du second plateau ($W_{21}^b + P_{21}^b$). Ainsi $W_{22}^b = \max(K_{22}^b * r - (P_{13}^b + W_{21}^b + P_{21}^b); 0)$. S'en suit la collecte prenant $P_{22}^b = p$ unités de temps (sachant que $L_{22}^b = 1$). Pour la visite du second et dernier plateau sur le troisième carrousel, le temps d'attente existe si les temps d'attente plus les temps de collecte sur les deux carousels précédents sont plus faibles que le temps de rotation sur ce carrousel. Ainsi $W_{23}^b = \max(K_{23}^b * r - (W_{22}^b + P_{22}^b + W_{21}^b + P_{21}^b); 0)$. Enfin la ligne de commande est collectée, prenant p unité de temps.

Pour les autres temps associés aux autres plateaux à collecter sur les carousels, la même procédure est suivie :

- Le temps d'attente pour le troisième plateau du carrousel 1 est de $W_{31}^b = \max(K_{31}^b * r - (W_{22}^b + P_{22}^b + W_{23}^b + P_{23}^b); 0)$ unités de temps suivi d'un temps de collecte de $L_{31}^b = 2$ lignes prenant $2 * p$ unités de temps.
- Pour le dernier plateau à visiter sur le carrousel 2, le temps d'attente est de $W_{32}^b = \max(K_{32}^b * r - (W_{23}^b + P_{23}^b + W_{31}^b + P_{31}^b); 0)$. S'en suit la collecte prenant $L_{32}^b * p = 2 * p$ unités de temps.

- Il reste encore à visiter deux plateaux sur le premier VLM. Le temps d'attente associé au quatrième plateau tient compte du temps masqué, $W_{41}^b = \max(K_{41}^b * r - (W_{32}^b + P_{32}^b); 0)$ et la collecte qui s'en suit prend p unités de temps.
- L'opérateur doit maintenant attendre devant le carousel afin que celui-ci tourne jusqu'au dernier plateau à visiter, prenant $W_{51}^b = \max(K_{51}^b * r; 0) = K_{51}^b * r$. Enfin la collecte sur ce dernier plateau, prenant p unités de temps, termine la collecte du lot b .

Le temps de préparation de commandes est donc la somme de tous les temps précédemment cités.

Preuve de la Proposition 5.

Par définition, nous avons $C = \sum_{b=1}^B \sum_{l=1}^{K_1^b} \sum_{u=1}^U (P_{lu}^b + W_{lu}^b)$. Sachant que le temps de préparation de commandes sur un plateau est proportionnel au nombre de lignes de commandes ayant besoin d'être collectées sur ce même plateau, le temps de collecte sur le l -ième plateau du Carousel u pour le lot b est $P_{lu}^b = L_{lu}^b * p$. De plus ici le temps pour passer du $l - 1$ ième plateau au l ième plateau du carousel u , requiert un temps $K_{lu}^b * r$.

Le temps d'attente en face du l -ième plateau du Carousel u pour lot b : W_{lu}^b , est la différence entre le temps de rotation du carousel u entre le $(l - 1)$ ième plateau à visiter et le l ième plateau, et le temps passé entre le moment où le préparateur finit de collecter sur le plateau $(l - 1)$ du Carousel u et le moment où il revient sur ce Carousel u pour la collecte sur le l -ième plateau. Le temps d'attente W_{lu}^b est donnée par la formule (1) – (2) – (3) où :

- (1) Le temps requis par le Carousel u pour passer du $(l - 1)$ ième plateau au l -ième. Nous rappelons que ce temps est de $K_{lu}^b * r$. De plus nous avons $K_{1u}^b = 0 \forall u \in \mathbf{v}$.
- (2) Le temps passé par l'opérateur pour collecter les références du $(u + 1)$ ième au U ième Carousels sur le $(l - 1)$ ième plateau à visiter et du 1 au $(u - 1)$ ième Carousel sur le l -ième plateau. Ce temps est égal à $\sum_{i=u+1}^U L_{l-1 i}^b * p + \sum_{i=1}^{u-1} L_{li}^b * p$,
- (3) Les temps d'attente en face du $(u + 1)$ ième au U ième Carousel du $(l - 1)$ ième plateau à visiter et du 1 au $(u - 1)$ ième Carousel du l -ième plateau à visiter. Ces temps d'attente sont égaux à $\sum_{i=u+1}^U W_{l-1 i}^b + \sum_{i=1}^{u-1} W_{li}^b$.

Si la Formule (1) – (2) – (3) donne une valeur négative, alors il n’y aura pas de temps d’attente et dans ce cas $W_{lu}^b = 0$. Le temps d’attente en face du l -ième plateau du Carousel u pour lot b est donc donné par :

$$W_{lu}^b = \max\left(K_{lu}^b * r - \left(\sum_{i=1}^{u-1} (L_{li}^b * p + W_{li}^b) + \sum_{i=u+1}^U (L_{l-1i}^b * p + W_{l-1i}^b)\right); 0\right) \forall 1 < u < U, \forall l \in \{2, \dots, K_1^b\}$$

$$W_{l1}^b = \max\left(K_{l1}^b * r - \sum_{i=2}^U (L_{l-1i}^b * p + W_{l-1i}^b); 0\right), \forall l \in \{2, \dots, K_1^b\}$$

$$W_{lU}^b = \max\left(K_{lU}^b * r - \sum_{i=1}^{U-1} (L_{li}^b * p + W_{li}^b); 0\right) \forall 1 < u < U, \forall l \in \{2, \dots, K_1^b\}$$

$$W_{lu}^b = 0 \text{ si } l = 1 \text{ comme nous considérons } K_{1u}^b = 0 \blacksquare$$

4.3.3 Modèle d’optimisation et étude de cas

Dans cette section, nous développons le modèle d’optimisation permettant un regroupement de commandes en lots lorsque la collecte s’effectue sur plusieurs Carousels. Ensuite, nous présentons et discutons les résultats des études de cas permettant de valider notre modèle et d’évaluer ses performances.

Tout comme le cas VLM, nous observons que la formulation du temps de préparation de commandes dans le cas plusieurs carousels est extrêmement complexe. Inclure directement le temps de préparation de commandes, exprimé dans la proposition 5, comme fonction objectif de notre modèle d’optimisation donnerait un modèle non linéaire difficilement solvable par des méthodes exactes. C’est pourquoi nous décidons d’utiliser une fonction objectif intermédiaire, qui est la somme sur l’ensemble des lots des distances parcourues par le carousel parcourant la plus grande distance pour chaque lot. Nous rappelons que le premier Carousel est, par convention, celui devant visiter le plus grand nombre de plateaux pour chacun des lots, mais pas nécessairement celui parcourant la plus grande distance. Le choix du critère est motivé par plusieurs arguments :

Nous observons tout d’abord que ce critère est le critère minimisant le temps de préparation de commandes dans le cas 1 carousel. Deuxièmement nous observons que la recherche de plateaux en temps masqué impacte grandement le temps de préparation de commandes. Ce critère favorise le temps masqué en réduisant la somme des distances maximales parcourues sur les carousels, essayant donc de ne pas faire la collecte sur un seul carousel. De plus, minimiser la plus grande distance parcourue par les carousels va avoir pour but de diminuer les distances

parcourues des carousels pour effectuer un même nombre de lignes de commandes, permettant une collecte plus efficace. Nous notons par ailleurs que cette diminution des distances permet une meilleure durée de vie des machines. Enfin, nos études de cas montrent une corrélation positive entre la minimisation de ce critère et la minimisation du temps de préparation de commandes montré dans la Proposition 5, comme nous le verrons ultérieurement.

Ainsi nous utilisons le critère de « minimisation de la somme sur les différents lots des distances parcourues par le carousel ayant parcouru la plus grande distance pour chacun des lots » comme fonction objectif de notre modèle d'optimisation.

4.3.3.1 Modèle d'optimisation

La principale décision de notre modèle est la construction de différents lots à partir d'un portefeuille de commandes. Le calcul de la distance s'effectue de la même manière que dans la section précédente dans le cas d'une collecte sur un seul carousel. Ainsi nous ne rappellerons pas le principe du calcul mais inciterons le lecteur à relire la Section 4.2.3.1. Les variables de décision sont :

- $x_{nb} = 1$ si la commande n est incluse dans le lot b ; 0 sinon.

- $T_{kbu} = 1$ si le plateau k du carousel u est visité durant le lot b ; 0 sinon. Par définition le nombre de plateaux devant être visités par le carousel u pour collecter le lot b est donné par $K_u^b = \sum_{k=1}^K T_{kbu}$.

Nous considérons le paramètre d'entrée U_{knu} qui est égal à 1 quand la commande n contient une référence devant visiter le plateau k du carousel u ; 0 sinon.

Nous introduisons la distance minimale parcourue par le carousel u afin de collecter les références sur tous les plateaux devant être visités pour le lot b : D_u^b . Nous rappelons que si un seul plateau doit être visité pour un lot, la distance parcourue par le carousel sera considérée nulle pour ce lot. Dans le cas de plusieurs plateaux devant être visités pour un lot donné, minimiser la distance parcourue par le carousel u est équivalent à trouver la distance minimale dans un cercle devant visiter une série de points, comme précédemment expliqué. Soit D_{ku}^b la distance entre le plateau k et le plus proche plateau de k devant être visité sur le carousel u et ayant un numéro de plateau inférieur à k .

Ainsi la fonction objectif est :

$$\text{Minimiser } \sum_{b=1}^B \max_{u \in \mathcal{U}} D_u^b$$

Les contraintes de notre modèle sont :

$$\sum_{b=1}^B x_{nb} = 1, \forall n \in \mathbb{N} \quad (4-1)$$

$$\sum_{n=1}^N x_{nb} \leq S, \forall b \in \mathcal{B} \quad (4-2)$$

$$T_{kbu} = \sum_{n=1}^N x_{nb} * U_{knu}, \forall b \in \mathcal{B}, \forall k \in \mathcal{K}, \forall u \in \mathcal{U} \quad (4-3)$$

$$D_u^b = K - \max_{l \in \{1, \dots, K_u^b\}} D_{lu}^b, \forall b \in \mathcal{B}, \forall u \in \mathcal{U} \quad (4-4)$$

$$\text{Si } K_u^b = \sum_{k=1}^k T_{kbu} = 1 \rightarrow D_{1u}^b = 0, \forall b \in \mathcal{B} \quad (4-5)$$

$$\text{Sinon } K_u^b = \sum_{k=1}^k T_{kbu} > 1, \forall b \in \mathcal{B}, \forall u \in \mathcal{U}$$

$$\rightarrow D_{Q_u^b(l)u}^b = Q_u^b(l-1) - Q_u^b(l) \forall l \in \{2, \dots, K_u^b\} \quad (4-6)$$

$$\rightarrow D_{Q_u^b(1)u}^b = K_u - Q_u^b(K_u^b) + Q_u^b(1) \quad (4-7)$$

La contrainte (4-1) garantit que chaque commande client est incluse dans un lot et un seul. La contrainte (4-2) impose que le nombre maximal de commandes par lots n'excède pas S . La contrainte (4-3) permet le calcul de T_{kbu} . En effet un lot doit visiter un plateau du carousel u si au moins une des commandes incluses dans ce lot doit visiter ce plateau. La contrainte (4-4) permet le calcul de la distance comme décrit précédemment. Si un seul plateau du carousel est visité, alors la distance parcourue sera nulle sur ce carousel (contrainte (4-5)). Si plusieurs plateaux doivent être vus sur un carousel, nous calculons la distance entre chacun des niveaux consécutivement visités sur ce carousel (contrainte (4-6) et (4-7)).

4.3.3.2 Etude de cas avec des données réelles

Nous comparons donc les performances de notre stratégie de regroupement avec la stratégie utilisée par l'Hôpital Français 2 (disposant d'une zone comprenant 2 Carousels) et de l'entreprise Suisse (la même que celle de la partie VLM, disposant d'une zone comprenant 4 Carousels) en fonction des deux critères expliqués dans la partie 1 Carousel : le critère proposé d'optimisation et le temps total de préparation de commandes donné en Proposition 5.

$$\text{Gain Distance} = \frac{\text{Critère avec la stratégie actuelle} - \text{Critère avec notre méthode}}{\text{Critère avec la stratégie actuelle}}$$

$$\text{Gain } C = \frac{C \text{ avec la stratégie actuelle} - C \text{ avec notre méthode}}{C \text{ avec la stratégie actuelle}}$$

Tout comme dans le cas 1 VLM, nous supposons que le temps requis au carousel pour passer d'un plateau à un plateau adjacent est $r=1$. Pour le gain C , nous supposons que $p = 1$ comme pour la partie 1 carousel. Nous regarderons ces gains pour chacune des entreprises. Enfin nous

observerons l'effet d'une augmentation du temps de picking par rapport au temps unitaire de rotation de carousel, sur les temps de préparation.

Nous avons codé puis résolu le problème d'optimisation décrit section 3.3.1 avec CPLEX version 12.6.3.0 sur un ordinateur possédant un processeur Intel Core i3 dual 2.3 GHz avec 3Go de mémoire vive et fonctionnant sur l'environnement Windows 7 Professionnel 32 bits. De plus, nous stoppons CPLEX après 30 minutes de calcul si la solution optimale n'a pas été prouvée. Nous faisons référence à de telles instances en notant « * » pour le temps de résolution. Dans ce cas, nous prenons la dernière solution fournie par CPLEX.

Etude de cas n° 1 : Hôpital Français 2

La zone de préparation de l'Hôpital Français 2 prise en compte dans cette étude de cas est constituée de 2 Carousels disposant de 43 plateaux chacun. Tout comme le cas 1 Carousel, le nombre de commandes par lot est de 10 maximum. Ainsi nous réglons les paramètres $U = 2$, $K_1 = K_2 = 43$ et $S = 10$. Nous considérons 5 instances représentant chacune une journée de commandes.

Les tableaux 21 et 22 ci-dessous récapitulent les résultats de la comparaison de notre méthode par rapport à la stratégie actuelle de regroupement de commandes. La Figure 34 ci-après illustre les résultats.

Tableau 21 : Les performances de notre modèle en termes de gain Distance pour l'Hôpital français 2

Instance	NB Commandes	NB lignes de commandes	Distance stratégie actuelle	Distance notre approche	Gain distance	Temps de résolution (en secondes)
Jour 1	31	234	122	82	33%	441
Jour 2	32	101	126	76	40%	1130
Jour 3	46	128	164	116	29%	1800*
Jour 4	50	243	170	135	21%	1800*
Jour 5	72	372	253	190	25%	1800*

Tableau 22 : Les performances de notre modèle en termes de gain C pour l'Hôpital français 2

Instance	NB Commandes	NB lignes de commandes	C stratégie actuelle	C notre approche	Gain C
Jour 1	31	234	305	273	10%
Jour 2	32	101	200	157	22%
Jour 3	46	128	270	214	21%
Jour 4	50	243	382	309	19%
Jour 5	72	372	553	470	15%

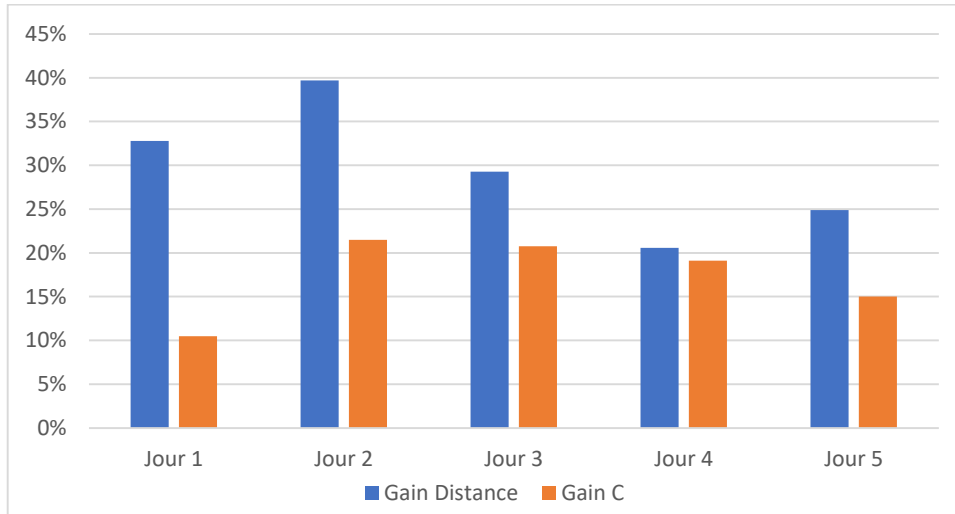


Figure 34 : Récapitulatif des gains pour l'hôpital Français 2

Etude de cas n° 2 : Entreprise Suisse

Pour l'entreprise Suisse, la zone de préparation de commandes est composée de 4 carousels avec 57 plateaux par Carousel. Le nombre maximal de commandes simultanément présentes en face des Carousels est de 8 commandes. Ainsi nous réglons les paramètres : $U = 4$, $K_1 = K_2 = K_3 = K_4 = 57$ et $S = 8$. Les tableaux 23 et 24 ci-dessous présentent les résultats obtenus et la Figure 35 illustre les gains en temps de préparation de commandes.

Tableau 23 : Les performances de notre modèle en termes de gain distance pour l'entreprise Suisse

Instance	NB Commandes	NB lignes de commandes	Distance stratégie actuelle	Distance notre approche	Gain distance	Temps de résolution (en secondes)
Jour 1	31	237	127	83	35%	1800*
Jour 2	36	302	128	91	29%	1800*
Jour 3	42	195	205	119	42%	1800*
Jour 4	52	142	255	134	47%	1800*
Jour 5	58	204	312	214	31%	1800*

Tableau 24 : Les performances de notre modèle en termes de gain C pour l'entreprise Suisse

Instance	NB Commandes	NB lignes de commandes	C stratégie actuelle	C notre approche	Gain C
Jour 1	31	237	328	282	14%
Jour 2	36	302	435	389	11%
Jour 3	42	195	356	296	17%
Jour 4	52	142	361	282	22%
Jour 5	58	204	512	360	30%

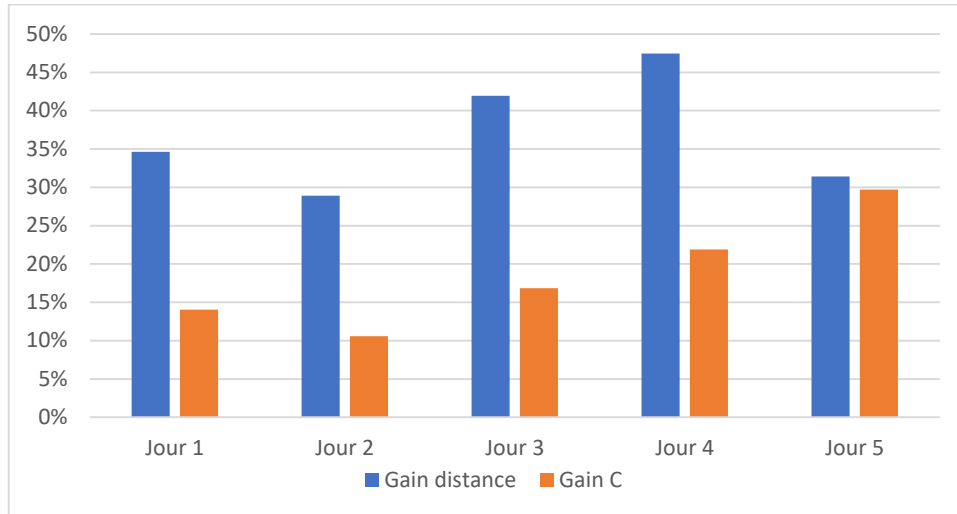


Figure 35 : Récapitulatif des gains pour l'Entreprise Suisse

Discussion générale des résultats

Nous observons que les résultats obtenus par notre modèle de regroupement de commandes améliorent ceux de la stratégie actuelle en termes de critère et de temps de préparation. De plus, nous observons des gains conséquents que nous commentons au cours de cette section. Nous observons que les gains Distance (rappelons que ce sont les gains au regard du critère utilisé) varie de 21% à 40% pour l'hôpital Français 2 avec une moyenne de 29%.

Concernant le temps de préparation, nous observons aussi un gain significatif allant de 10% pour l'instance « jour 1 » à 22% pour l'instance « jour 2 » avec une moyenne sur l'ensemble des instances de 17%. Pour l'entreprise Suisse, nous observons aussi des gains significatifs, allant de 29% à 47% pour le critère avec une moyenne de 37% et les gains en temps de préparation sont de 19% en moyenne et vont de 11% à 30%.

De plus, nous observons des temps de résolution très importants pour les deux études de cas. Nous observons que pour ces deux entreprises, la majorité des instances testées impliquent un temps de calcul très important, ne semblant plus dépendre uniquement du nombre de commandes, mais aussi du nombre de carousels présents.

Nous rappelons que des approches métaheuristiques sont testées en Section 4 de ce chapitre afin de résoudre efficacement les grandes instances où nous ne pouvons prouver l'optimalité avec CPLEX.

Etude du l'impact d'une augmentation du temps de picking sur le résultat.

Nous avons montré que notre modèle de regroupement de commandes apporte des solutions permettant une diminution significative du temps de préparation de commandes. Dans cette partie nous allons observer l'effet du temps de picking. En ayant plus de temps de collecte, nous augmentons le travail en temps masqué et donc diminuons le temps d'attente, mais inversement le temps de collecte total augmente.

Nous rappelons que le temps de collecte unitaire peut être variable. En effet une opération d'emballage ou de pesage peut s'ajouter dans un entrepôt, et le temps de collecte deviendra plus grand que le temps de rotation unitaire du carrousel. Nous étudions ici l'impact d'une augmentation du temps de collecte sur nos résultats.

Nous considérons toujours un temps de rotation égal à $r = 1$, et nous regardons pour les cas $p = 1$; $p = 2$ et $p = 3$, les résultats en termes de temps d'attente et de temps de préparation de commandes. Nous regardons les trois temps primordiaux pour notre étude, le temps d'attente, de picking et de préparation de commandes. Nous rappelons que le temps de préparation de commandes est la somme des deux temps précédemment cités. Nous observons dans chacun des cas ci-dessous que lorsque p augmente, le temps d'attente W diminue.

Hôpital Français 2

Les tableaux 25 et 26 ci-dessous montrent les différents temps (d'attente, de picking et de préparation de commandes) pour les différents scénarios de temps de collecte.

Tableau 25 : Temps d'attente, de picking ainsi que de préparation de commandes en fonction du temps unitaire de picking pour notre méthode appliquée à l'Hôpital Français 2

Notre méthode	r=p=1			r=1; p=2			r=1; p=3		
	W	P	C	W	P	C	W	P	C
Jour 1	39	234	273	28	468	496	21	702	723
Jour 2	56	101	157	36	202	238	26	303	329
Jour 3	86	128	214	55	256	311	39	384	423
Jour 4	66	243	309	48	486	534	32	729	761
Jour 5	98	372	470	62	744	806	43	1116	1159

Tableau 26 : Temps d'attente, de picking ainsi que de préparation de commandes en fonction du temps unitaire de picking pour la stratégie actuelle appliquée à l'Hôpital Français 2

Stratégie actuelle	r=p=1			r=1; p=2			r=1; p=3		
	W	P	C	W	P	C	W	P	C
Jour 1	71	234	305	47	468	515	35	702	737
Jour 2	99	101	200	68	202	270	46	303	349
Jour 3	142	128	270	106	256	362	80	384	464
Jour 4	139	243	382	108	486	594	90	729	819
Jour 5	181	372	553	134	744	878	96	1116	1212

Nous présentons dans le tableau 27 les gains en temps d'attente et en temps de préparation de commandes entre la stratégie actuelle de l'entreprise et notre méthode. Nous appelons Gain W le gain en temps d'attente :

Tableau 27 : Gains en temps d'attente et en temps de préparation de commandes pour L'hôpital Français 2

Comparaison	r=p=1		r=1 et p=2		r=1 et p=3	
	Gains W	Gains C	Gains W	Gains C	Gains W	Gains C
Jour 1	45,1%	10,5%	40,4%	3,7%	40,0%	1,9%
Jour 2	43,4%	21,5%	47,1%	11,9%	43,5%	5,7%
Jour 3	39,4%	20,7%	48,1%	14,1%	51,3%	8,8%
Jour 4	52,5%	19,1%	55,6%	10,1%	64,4%	7,1%
Jour 5	45,9%	15,0%	53,7%	8,2%	55,2%	4,4%

Entreprise Suisse :

Nous montrons dans les tableaux ci-dessous les différents temps d'attente, de picking et de préparation de commande en fonction des temps de picking unitaire.

Tableau 28 : Temps d'attente, de picking ainsi que de préparation de commandes en fonction du temps unitaire de picking pour notre méthode appliquée à l'Entreprise Suisse

Notre méthode	r=p=1			r=1; p=2			r=1; p=3		
	W	P	C	W	P	C	W	P	C
Jour 1	45	237	282	23	474	497	13	711	724
Jour 2	87	302	389	77	604	681	71	906	977
Jour 3	101	195	296	58	390	448	35	585	620
Jour 4	140	142	282	112	284	396	87	426	513
Jour 5	156	204	360	102	408	510	63	612	675

Tableau 29 : Temps d'attente, de picking ainsi que de préparation de commandes en fonction du temps unitaire de picking pour la stratégie actuelle appliquée à l'Entreprise Suisse

Stratégie actuelle	r=p=1			r=1; p=2			r=1; p=3		
	W	P	C	W	P	C	W	P	C
Jour 1	91	237	328	65	474	539	48	711	759
Jour 2	133	302	435	114	604	718	108	906	1014
Jour 3	161	195	356	109	390	499	73	585	658
Jour 4	219	142	361	173	284	457	142	426	568
Jour 5	308	204	512	232	408	640	178	612	790

Nous présentons dans le tableau suivant les gains en temps d'attente et en temps de préparation de commandes entre la stratégie actuelle et notre méthode. Nous appelons Gain W le gain en temps d'attente tandis que le Gain C reste inchangé :

Tableau 30 : Gains en temps d'attente et en temps de préparation de commandes pour l'Entreprise Suisse

Comparaison	r=p=1		r=1 et p=2		r=1 et p=3	
	Gains W	Gains C	Gains W	Gains C	Gains W	Gains C
Jour 1	50,5%	14,0%	64,6%	7,8%	72,9%	4,6%
Jour 2	34,6%	10,6%	32,5%	5,2%	34,3%	3,6%
Jour 3	37,3%	16,9%	46,8%	10,2%	52,1%	5,8%
Jour 4	36,1%	21,9%	35,3%	13,3%	38,7%	9,7%
Jour 5	49,4%	29,7%	56,0%	20,3%	64,6%	14,6%

Discussion des résultats :

Nous observons que le temps d'attente pour les deux entreprises diminue si le temps de picking augmente. Le temps de préparation augmente à cause du temps de picking qui devient prépondérant. Les gains seront donc plus importants lorsque nous aurons un faible nombre de lignes de commandes, puisque le temps de picking incompressible est plus faible. Les gains seront aussi plus robustes à une variation du temps de picking unitaire s'il y a un faible nombre de lignes à collecter, car le temps de picking n'augmentera que faiblement en valeur absolue. Nous voyons que les gains W sont d'autant plus importants que p augmente, ce qui montre que le critère choisi cible bien le temps d'attente et va permettre une meilleure réduction de ce temps qu'avec la stratégie actuelle. Par ailleurs, nous observons que le temps de picking devient prépondérant si p augmente, ce qui réduit grandement les gains obtenus en comparant la stratégie actuelle à notre méthode.

De par les résultats montrés ci-dessus, nous recommandons vivement l'utilisation de notre critère en cas de préparation de commandes avec des carousels.

4.4 Résolution métaheuristique

Nous avons observé pour la quasi-totalité des instances testées, qu'un temps de calcul important est nécessaire. L'optimalité n'est pas prouvée en 30 minutes par CPLEX pour de nombreuses instances. Nous rappelons que notre but est de développer un outil de regroupement de commandes utilisable dans la pratique. De plus notre partenaire industriel voulant une solution pouvant être intégrée rapidement au logiciel de KLS et donnant des solutions satisfaisant en un temps de calcul restreint (une à cinq minutes de calcul), nous optons pour l'utilisation de métaheuristicques.

Au vu de la littérature existante, nous avons pris la décision de coder un algorithme génétique. Nous souhaitons montrer au cours de cette partie, que nous pouvons obtenir des résultats satisfaisants. Nous pourrions certainement aller plus loin en testant d'autres métaheuristicques.

Ayant déjà convenu avec KLS de régler le temps de calcul à 1 minute nous montrerons dans cette partie uniquement les performances de notre algorithme génétique avec la calibration que nous avons trouvée la plus performante, et cela sous une minute de calcul. Cette calibration donne de meilleurs résultats que ceux de la calibration « standard », d'où notre choix. Pour plus d'information sur la structure de la solution et les décisions prises pour la calibration de l'algorithme génétique, nous reportons le lecteur à la partie 4 du chapitre 3 (section métaheuristique avec utilisation de VLMs). Seuls les pourcentages de croisement et de mutation utilisés diffèrent. En effet nous avons observé que les croisements ne donnent pas de solution réalisable dans une majorité des cas, ainsi nous utilisons la calibration suivante :

Pourcentage de Mutation : 70%

Pourcentage de Croisement : 1%

Nous utilisons le critère de notre modèle d'optimisation comme fonction objectif pour les métaheuristicques (la somme de la plus grande distance parcourue pour chaque lot), permettant une comparaison avec les valeurs fournies par la stratégie actuelle des entreprises et CPLEX. Pour chaque instance, la métaheuristique est lancée 10 fois et la moyenne des 10 résultats obtenus est gardée, que nous appelons Valeur Fitness Moyenne (VFM). Pour la comparaison des résultats, nous utilisons les mêmes critères que ceux utilisés dans la partie VLMs. :

$$TA = \frac{\text{Critère de la stratégie actuelle} - \text{VFM}}{\text{Critère de la stratégie actuelle}}$$

$$TE = 1 - \frac{\text{VFM} - \text{Résultat CPLEX 1h de calcul}}{\text{Résultat CPLEX 1h de calcul}}$$

Le tableau suivant présente les résultats pour les instances considérées, pour 1 minute de calcul (Tableau 31).

Tableau 31 : Récapitulatif des résultats des métaheuristiques testées (1 minute de calcul)

Entreprise	Instances	N	Critère approche actuelle	CPLEX	Algorithme Génétique		
					VFM	TA	TE
Hôpital Français 2	Jour 1	31	122	82	82,7	32%	99%
	Jour 2	32	126	76	76	40%	100%
	Jour 3	46	164	116	108,7	34%	106%
	Jour 4	50	170	135	119,8	30%	111%
	Jour 5	72	253	190	130,4	48%	131%
Entreprise Suisse	Jour 1	31	127	83	83	35%	100%
	Jour 2	36	128	91	100	22%	90%
	Jour 3	42	205	119	119,4	42%	100%
	Jour 4	52	255	134	84,2	67%	137%
	Jour 5	58	312	214	136	56%	136%
Moyenne						40,53%	111,15%

Nous observons un TA allant de 22% à 67% avec une moyenne de 40.5%. De même, nous observons un TE variant entre 90% et 137% et une moyenne de 111.2%. Ainsi l'algorithme génétique avec notre calibration donne des résultats significativement meilleurs que CPLEX après 30 minutes de calcul, et ceci avec un temps de calcul de seulement une minute. Nous avons donc implémenté cette métaheuristique avec la calibration proposée au sein du logiciel GILDAS de l'entreprise KLS Logistic pour le module de regroupement de commandes en lots.

Au cours de ce chapitre, nous avons fourni une méthodologie permettant le calcul du temps de préparation de commandes lorsque la collecte s'effectue dans une zone de préparation équipée d'un ou de plusieurs carousels. De plus, nous avons fourni un modèle d'optimisation permettant un regroupement de commandes en lots efficient et avons montré son efficacité grâce à de plusieurs études de cas. Enfin une approche métaheuristique a été implémentée chez KLS Logistic, permettant un regroupement de commandes en lots donnant des solutions performantes sous une minute de calcul.

Le prochain chapitre de cette thèse est consacré au séquençage des commandes dans un système « Drive ».

Chapitre 5 Séquencement des commandes dans un système Drive

Dans ce chapitre, nous allons nous focaliser sur l'activité de préparation de commandes dans un système dénommé Rolling Batching. Nous dirons aussi « Système Drive », car ce mode de picking est le plus souvent rencontré dans des magasins et entrepôts « Drive ». La première section de ce chapitre décrit les caractéristiques de ce type de système. La seconde partie sera dédiée à la définition d'un algorithme permettant le calcul du temps d'attente de l'opérateur lors de la collecte d'un portefeuille de commandes. La section 3 est dédiée aux heuristiques et à la métaheuristique utilisées afin de résoudre de manière approchée ce problème. Enfin une étude de cas permettant d'observer les performances des approches utilisées est présentée, concluant ce chapitre.

5.1 Caractéristiques du système « Drive »

Ces systèmes sont utilisés dans les magasins « Drive » ou plateforme e-commerce, qui ont besoin d'un rythme de préparation soutenu notamment lors des heures de pointes. La figure 36 permet de décrire le système drive. Sur cette image issue d'un partenaire de KLS Logistic, nous observons plusieurs préparateurs, deux d'entre eux préparent les commandes en face de plusieurs carousels horizontaux.

Les commandes en cours de collecte sont celles sur la table de préparation. Nous observons ici que les opérateurs n'ont que très peu de distance à parcourir pour collecter et déposer les produits, confirmant l'hypothèse faite au chapitre 4 concernant le temps de marche négligeable. Une fois une commande terminée, l'opérateur sur la droite vient récupérer ces commandes, les contrôle puis les dépose sur un convoyeur permettant à ces commandes d'aller sur la prochaine zone de préparation.



Figure 36 : Illustration d'un système "Drive" (source : www.kardex-remstar.be)

Nous allons identifier plusieurs caractéristiques importantes de ce système :

Tout d'abord, ce système est souvent caractérisé par un nombre élevé de machines, afin de bénéficier de changements de plateaux effectués en temps masqué. Ces machines sont généralement des carousels, adaptés aux produits stockés dans ces entrepôts.

Ce système est aussi caractérisé par le fait que plusieurs commandes sont simultanément en cours de collecte sur plusieurs carousels. Jusqu'ici, ce système est similaire au regroupement de commande en lots complets. La différence apparaît lorsqu'une commande se termine, elle est immédiatement remplacée par une nouvelle commande sur la table de préparation, tandis que dans le cas de regroupement de commandes en lots présenté dans les chapitres précédents, il faut attendre que toutes les commandes soient collectées avant que de nouvelles commandes arrivent. Ainsi dans le Drive, lorsqu'on collecte la dernière ligne d'une commande, celle-ci est retirée de la table de préparation et remplacée par la commande suivante devant être mise en préparation.

Il y aura donc un nombre constant de commandes dans la zone de préparation au cours de la collecte, jusqu'au moment où il ne restera plus de commandes à rajouter sur la table de préparation. Notre but revient donc à séquencer les commandes.

Au cours de ce chapitre nous nous intéressons à la définition d'un ordre d'ajout des commandes au sein de la zone de préparation pour minimiser le temps de préparation de commandes. La prochaine section décrit l'algorithme utilisé pour la définition du temps d'attente à partir d'un portefeuille de commandes ainsi qu'un ordre d'ajout des commandes connues.

5.2 Définition de l'algorithme permettant le calcul du temps d'attente dans un système « Drive »

Au cours de cette partie, nous définissons l'algorithme utilisé pour le calcul du temps d'attente. Tout d'abord nous listons les hypothèses prises en compte, puis nous détaillons les différentes étapes de notre algorithme.

5.2.1 Hypothèses prises en comptes pour le calcul du temps d'attente d'un système « Drive »

Nous considérons un nombre fixe de commandes simultanément en préparation, que nous nommons B . Ainsi lorsqu'une commande est préparée, elle est retirée immédiatement de la table de préparation et la commande suivante est ajoutée en fonction de l'ordre d'ajout.

Le carousel est Monodirectionnel, contrairement au chapitre 4. Cette hypothèse nous a été proposée par KLS Logistic car lors d'une grande cadence de commandes avec plusieurs commandes simultanément collectées, le fait que le carousel change de sens de rotation n'est pas justifiée, ou simplement pour les dernières commandes, une fois l'ajout de la dernière commande effectuée.

Lorsqu'une commande vient d'être préparée sur un carousel et que le préparateur la retire de la table de préparation et rajoute une commande, nous considérons que la collecte va reprendre sur le carousel ayant terminé la préparation. Concernant les autres carousels, nous considérons que ces carousels n'effectuent pas de rotation lors du changement de commandes et restent à leurs derniers plateaux visités jusqu'à la définition du prochain plateau à visiter. Cette hypothèse évite une rotation du carousel si la nouvelle commande a des références à collecter sur ce plateau, mais retarde inutilement la rotation du carousel dans le cas contraire.

Tout comme dans la Chapitre du regroupement en lots avec l'utilisation de carousels, nous considérons le temps de marche entre la table de préparation et les carousels négligeable (cf. Figure 36). Concernant les autres temps, nous utilisons l'hypothèse faite dans le Chapitre 4 : le temps de rotation entre deux plateaux est proportionnel au nombre de plateaux intermédiaires. Concernant le temps de collecte, nous considérons l'hypothèse réaliste où le temps de picking par plateau est proportionnel au nombre de lignes à collecter sur le plateau.

Ayant pris les hypothèses évoquées précédemment en considération, nous proposons l'algorithme suivant pour le calcul du temps de préparation de commandes dans un système drive.

5.2.2 Présentation de l'algorithme utilisé

Entrée :

- Vecteur « Ordre d'ajout » des commandes
- Portefeuille de commandes avec les emplacements à visiter pour chaque ligne de commandes

Début

Initialisation des paramètres :

- Nombre de commandes simultanément en préparation : B
- Nombre de carousels
- Emplacement initial de chaque carousel
- Création et définition des plateaux à voir pour chacune des commandes auxquelles on attribue le statut « en attente »

Ajout des premières commandes sur la table de préparation

- Les B premières commandes du vecteur d'entrée : « Ordre d'ajout » sont placées en préparation. Statut de ses commandes : « En préparation »

Tant qu'il reste des commandes sur la table de préparation

- Collecte jusqu'à la fin de la préparation d'une commande
 - Identification de la commande finissant en premier parmi celles de la table
 - Calcul des distances à parcourir entre chacune des lignes à collecter
 - Calcul du temps d'attente pour finir cette commande
 - Enlever cette commande préparée de la table de préparation en changeant son statut en « préparé »
- Actualisation des paramètres
 - Emplacement actuel de chacun des carousels
 - Niveaux restants à voir par chacune des commandes au cours de la préparation
 - Carousel terminant la collecte
- Si il reste des commandes « en attente »
 - Ajout d'une commande en fonction de l'ordre d'ajout

Fin si

Fin tant que

Calcul temps d'attente pour la préparation de l'ensemble des commandes

Fin

Sortie :

- Temps d'attente associé au vecteur « Ordre d'ajout »

Cet algorithme présente de manière synthétique les grandes étapes de calcul. Chacune de ces étapes nécessite des calculs importants, présentés en annexe sur un exemple. Nous allons dans cette section décrire les grandes étapes de l'algorithme :

Nous rappelons tout d'abord qu'en entrée de cet algorithme, nous avons un portefeuille de commandes à préparer ainsi qu'un vecteur définissant l'ordre d'ajout des commandes sur la table de préparation. Le portefeuille de commandes va nous permettre de savoir quels sont les plateaux à visiter par chacune des commandes, tandis que l'ordre d'ajout des commandes, comme son nom l'indique, est un vecteur permettant de connaître l'ordre des indices des commandes à rajouter sur la table de préparation.

L'initialisation des paramètres permet de régler tous les paramètres de l'algorithme dont nous nous servons, et qui caractérisent le système, à savoir : Le nombre de commandes simultanément réalisables, le nombre de carousels présents pour effectuer la préparation, ainsi que l'emplacement initial de chacun des carousels, que nous supposons être, sans perte de généralité, à l'emplacement 1 dans nos expérimentations. Nous attribuons à toutes les commandes le statut « en attente », statut qui évoluera à « en préparation » puis « préparée » au fur et à mesure de l'avancement des tâches.

Une fois les paramètres initialisés, nous ajoutons les B premières commandes sur la table, en changeant leur statut de « en attente » à « en préparation ».

L'étape de préparation de commandes peut commencer. Tout d'abord nous devons déterminer la commande qui va se terminer en premier (i.e. dont l'ensemble des lignes de commandes seront collectées). Nous déterminons l'ensemble des plateaux devant être visités par les commandes en préparation. A partir de ces plateaux, nous déterminons le nombre de plateaux à visiter pour chacune des commandes et ainsi nous déterminons la commande finissant en premier. Nous gardons en mémoire les plateaux visités avant d'avoir fini cette commande ainsi que chacune des distances à parcourir pour atteindre ces plateaux. Ces deux informations permettent de calculer le temps d'attente.

Ce calcul, fait comme dans la partie 4.3.3, est le suivant : nous regardons la distance parcourue pour un caroussel pour atteindre un plateau, cette distance multipliée par le temps de rotation unitaire du caroussel donne le temps pour passer d'un plateau au plateau suivant à visiter. Nous comparons ce temps au temps pris par l'opérateur sur les autres carousels pendant que le caroussel cherche le plateau à visiter. Nous rappelons que le temps sur les autres carousels est constitué de temps d'attente ainsi que de temps de picking, d'où l'importance de connaître le nombre de lignes à collecter sur chacun des plateaux. Nous réitérons ce procédé pour tous les plateaux vus jusqu'à ce qu'une commande se termine, nous donnant le temps d'attente associé à la collecte de cette commande.

Une fois cette commande terminée et le temps d'attente associé à la collecte de cette commande déterminé, nous actualisons les paramètres. Cette actualisation permet de définir les plateaux restant à visiter pour chacune des commandes en préparation, ainsi que l'emplacement de chacun des carousels. Le caroussel sur lequel se termine la commande sera celui sur lequel recommencera la collecte. Nous changeons aussi le statut de la commande venant d'être collectée à « Préparée ».

S'il reste des commandes ayant le statut « En attente », nous rajoutons alors en fonction de l'ordre d'ajout la prochaine commande sur la table de préparation en lui attribuant le statut « En préparation ».

Nous réitérons le procédé de préparation jusqu'à ce que l'ensemble des commandes aient le statut « préparée » et calculons le temps d'attente total en sommant les temps d'attente précédemment calculés.

Cet algorithme permet donc la simulation du temps d'attente pour la collecte d'un portefeuille de commandes sous un système de préparation « drive ». Cet algorithme a été validé par notre partenaire industriel. Ainsi un module du Logiciel Gildas distribué par l'entreprise KLS Logistic est en cours de développement permettant cette simulation. Nous avons donc défini une méthode permettant la simulation du temps d'attente à partir d'un portefeuille de commandes et d'un vecteur définissant l'ordre d'ajout des commandes. La section suivante permet la création d'un ordre d'ajout adapté aux systèmes « drive » avec pour objectif une amélioration du temps d'attente par rapport à la stratégie actuelle, que l'on quantifiera lors de l'étude de cas en fin de ce chapitre.

5.3 Résolution approchée

Nous avons, lors de la section précédente, défini la méthodologie nous permettant de calculer, à partir d'un ordre d'ajout de commandes donné, le temps de préparation de commandes correspondant. Dans cette section nous définissons des heuristiques constructives et une métaheuristique, que nous testons dans la prochaine section dédiée à l'étude de cas. Ces méthodes de résolution approchée visent à fournir une solution satisfaisante avec un temps de calcul raisonnable.

5.3.1 Définitions d'heuristiques constructives

Nous avons défini plusieurs heuristiques permettant la création d'un ordre d'ajout. Au cours du chapitre précédent, nous avons identifié la distance comme paramètre influençant la préparation de commandes. En effet si nous ajoutons une commande pour laquelle la distance que doit parcourir chaque carousel est très faible, alors cette commande sera plus rapidement préparée que si elle a une grande distance à parcourir sur chacun des carousels. Nous pouvons aussi citer le nombre de plateaux à visiter par chacune des commandes comme facteur impactant le temps de préparation de commandes. Ainsi nous avons défini plusieurs paramètres intrinsèques à une commande semblant influencer sur les temps de préparation et avons ainsi construit des heuristiques en fonction de ces paramètres. Les paramètres impactant que nous avons testés sont les suivants :

- Le nombre de plateaux devant être vus par une commande sur l'ensemble des carousels. Ce paramètre peut être défini a priori et nous calculons pour chacune des commandes n , $\sum_{u=1}^U \sum_{k=1}^{K_u} U_{k n u}$ qui nous donnera, pour chacune des commandes n , le nombre de plateaux total à visiter.
- Le nombre de plateaux devant être vus, pour une commande, sur le carousel visitant le plus de plateaux pour cette commande. Ce nombre est aussi intrinsèque à la commande et nous calculons pour chacune des commandes n , $\max_{u \in \mathbb{U}} \sum_{k=1}^{K_u} U_{k n u}$ qui nous donnera, pour chacune des commandes n , le nombre de plateaux maximum à visiter. Nous nommons ce paramètre « Nb plateaux max ».
- La distance à parcourir sur le carousel parcourant la plus grande distance pour finir une commande. Ce paramètre change en fonction du plateau où se trouve le carousel. Il faudra donc calculer, à chaque fois qu'une commande se termine, pour chaque commande n non préparée, la distance maximale pour préparer cette commande n . Soit t , l'instant où la t -

ième commande est terminée. Soit $D_{n u}^t$ la distance que cette commande n doit parcourir à t pour être collectée sur le carousel u , la distance que l'on considérera dans nos heuristiques sera définie par $\max_{u \in \mathcal{U}} D_{n u}^t$.

- La densité de plateaux sur le carousel ayant la plus grande distance à parcourir. Donc pour chaque instant t , pour chaque commande n non préparée, soit j le carousel tel que

$$j = \underset{u \in \mathcal{V}}{\operatorname{argmax}} D_{n u}^t, \text{ la densité de la commande } n \text{ sera : } \frac{\sum_{k=1}^{K_j} U_{k n j}}{D_{n j}^t}.$$

A partir de ces paramètres nous définissons les heuristiques suivantes pour définir l'ordre d'ajout des commandes :

- Soit H1-1 l'heuristique rajoutant la commande ayant le plus grand nombre de plateaux total à visiter. Nous calculons donc le nombre de plateaux total à visiter pour chacune des commandes « en attente » et ajoutons celle ayant le plus grand nombre de plateaux total à visiter.
- Soit H1-2 l'heuristique rajoutant la commande ayant le plus faible nombre de plateaux total à visiter. Nous calculons donc le nombre de plateaux total à visiter pour chacune des commandes « en attente » et ajoutons celle ayant le plus faible nombre de plateaux total à visiter.
- Soit H2-1 l'heuristique rajoutant la commande ayant la plus grande valeur de « Nb plateaux max ». Nous calculons ce paramètre, pour chacune des commandes « en attente », et ajoutons celle ayant la plus grande valeur.
- Soit H2-2 l'heuristique rajoutant la commande ayant la plus faible valeur de « Nb plateaux max ». Nous calculons ce paramètre, pour chacune des commandes « en attente », et ajoutons celle ayant la plus faible valeur.

Ces premières heuristiques permettent, dès l'instant $t=0$, de définir l'ordre d'ajout pour l'ensemble des commandes à collecter (si on ne considère pas d'éventuelles nouvelles commandes arrivant au fil de l'eau).

- Soit H3-1 l'heuristique rajoutant la commande ayant la plus grande distance. A chaque fois qu'une commande est préparée, nous calculons, à partir du plateau sur lequel est positionné chacun des carousels, la distance pour finir la commande. Pour chaque commande en

attente, nous rajoutons sur la table de préparation la commande ayant la plus grande distance.

- Soit H3-2 l'heuristique rajoutant la commande ayant la plus faible distance. A chaque fois qu'une commande est préparée, nous calculons la distance pour préparer chacune des commandes « en attente », à partir du plateau sur lequel est positionné chacun des carousels. Nous rajoutons sur la table de préparation la commande ayant la plus faible distance pour être terminée.
- Soit H4 l'heuristique rajoutant la commande ayant la plus forte densité de plateaux à visiter, la densité étant calculée comme décrit précédemment.

Pour ces dernières heuristiques (H3-1, H3-2, H4), le calcul se fait à chaque fois qu'il faut ajouter une commande, et ceci pour toutes les commandes « en attente ».

A partir des heuristiques présentées ci-dessus, nous construisons des groupes d'heuristiques. En effet, nous avons observé que rajouter des commandes ayant uniquement des faibles distances (ou un faible nombre de plateaux), permet de finir rapidement les premières commandes, mais laisse à la fin les commandes devant faire le tour du carrousel pour être finies. Similairement, commencer par les plus grandes commandes conduit aussi à des mauvais résultats. Nous avons donc décidé d'avoir des heuristiques mêlant des commandes ayant un faible nombre de plateaux à voir (ou de distance à parcourir) avec celles ayant un grand nombre de plateaux à voir (ou distance à parcourir).

Nous testons ici deux groupes d'heuristiques. Ces groupes d'heuristiques reposent sur une notion de statut appelé *G*C*i*, indiquant que cette commande est la plus « Grande Commande » au moment où on la choisit. Le nombre de « Grandes Commandes » dans un lot en cours de collecte sera fixe. Ainsi une fois le nombre de commandes en préparation ayant le statut *G*C*i* atteint, la commande suivante sera introduite en utilisant une autre heuristique en l'occurrence, dans notre travail, on recherchera la plus « Petite Commande » au moment considéré. Nous allons donc mixer les « Grandes Commandes » avec les « Petites Commandes » et créer une heuristique permettant d'avoir toujours le même nombre de grandes commandes sur la table de préparation. La notion de plus « Grande Commande » et plus « Petite Commande » sera précisée lorsqu'on définira les groupes d'heuristiques.

Ne sachant cependant pas à priori quel est le nombre de « Grandes Commandes » permettant d'avoir les meilleurs résultats, nous testons toutes les possibilités. Nous aurons donc deux groupes

d'heuristiques présentés ci-dessous comprenant $B+1$ heuristiques chacun (rappelons que B est le nombre maximal de commandes sur la table de préparation).

Groupe d'heuristiques 1 :

Statut GC1 : statut attribué à une commande lors de sa mise en préparation, indiquant qu'elle était celle devant visiter le plus grand nombre de plateaux pour être terminée au moment de son introduction.

Heuristique GC1-x : Cette heuristique consiste à avoir x commandes ayant le statut GC1 présentes sur la table de préparation. Par exemple GC1-4 indique qu'il faut qu'il y ait toujours exactement quatre commandes avec le statut GC1 sur la table de préparation.

Le groupe d'heuristiques GC1 comprend $B+1$ heuristiques GC1-x, pour $x=0$ à $x=B$. Le fonctionnement de chacune des heuristiques est comme suit : à $t=0$, x commandes sont ajoutées en préparation suivant l'heuristique H1-1, et $B-x$ suivant l'heuristique H1-2. Par la suite, lors du choix de la commande à rajouter, s'il y a moins de x commandes avec le statut GC1 en préparation, nous ajoutons une commande suivant l'heuristique H1-1 (pour avoir toujours x commandes avec le statut GC1), sinon nous ajoutons une commande suivant l'heuristique H1-2. Ainsi pour $x=0$, l'heuristique GC1-0 sera équivalente à l'heuristique H1-2 car nous ne voulons aucune commande avec le statut GC1 et pour $x=B$, l'heuristique GC1-B est la même que H1-1 car nous voulons tout le temps B commandes sur la table ayant le statut GC1.

Groupe d'heuristiques 2 :

Statut GC2 : statut attribué à une commande lors de sa mise en préparation, indiquant qu'elle était celle devant visiter la plus grande distance pour être terminée au moment de son introduction.

Heuristique GC2-x : Cette heuristique consiste à avoir x commandes ayant le statut GC2 présentes sur la table de préparation. Par exemple GC2-1 indique donc qu'il faut qu'il y ait toujours une et une seule commande avec le statut GC2 sur la table de préparation.

Le groupe d'heuristique GC2 comprend $B+1$ heuristiques, pour $x=0$ à $x=B$. Le fonctionnement des heuristiques GC2-x est similaire à celui des GC1-x, seules les heuristiques pour l'ajout des commandes changent. Ici les heuristiques H3-1 et H3-2 sont utilisés : s'il y a moins de x commandes avec le statut GC2 en préparation lors du choix de l'ajout d'une commande, nous en ajoutons une suivant l'heuristique H3-1 et lui attribuons le statut GC2, sinon nous ajoutons une

commande suivant l'heuristique H3-2. Comme pour le groupe d'heuristiques GC1, ici GC2-0 sera équivalente à H3-2 et GC2-B sera équivalente à H3-1.

5.3.2 Définition de la Métaheuristique

Afin d'aller plus loin dans la résolution, nous avons aussi implémenté une métaheuristique permettant une résolution approchée donnant des solutions performantes sous un temps de calcul raisonnable pour l'entreprise. Nous avons choisi, afin d'obtenir des résultats rapidement, de coder une métaheuristique de type recherche locale, nous permettant de voir les marges d'amélioration réalisables en comparaison des précédentes heuristiques et par rapport à la stratégie actuelle de l'entreprise. Nous décrivons ci-dessous le paramétrage de la recherche locale utilisée :

Encodage de la solution

Pour l'encodage de la solution, nous avons utilisé le vecteur d'ajout, ainsi un vecteur A contenant N éléments, les indices des commandes. Cet ordre d'ajout indique dans quel ordre les commandes seront ajoutées à la table de préparation.

Soit l'exemple suivant ayant 4 commandes. Si la table de préparation peut contenir 2 commandes simultanément, les commandes Re4 et 2 sont ajoutées en premier. Lorsqu'une de ces deux commandes se termine, la commande 3 est ajoutée, et enfin lorsqu'une commande se termine la commande 1 est rajoutée sur la table de préparation.

Tableau 32 : Encodage de la solution

Ordre d'ajout	4	2	3	1
---------------	---	---	---	---

A partir de la méthode de calcul précédemment expliquée, nous calculons le temps d'attente résultant à partir de cette solution.

Recherche d'une solution voisine :

Pour passer d'une solution à la solution voisine, nous utilisons la méthode SWAP expliquée dans le chapitre 3 section 3.4.2.1

Fonctionnement de l'algorithme :

Tout d'abord, les précédentes heuristiques et groupes d'heuristiques sont testées, la solution la plus performante (en termes de temps d'attente) devient la solution initiale de la

métaheuristique. Cela permet de fournir une solution initiale performante. A partir de cette solution améliorée, un SWAP est effectué et le temps d’attente de la solution est évalué. Si ce temps d’attente est plus performant que le temps d’attente de la solution actuelle, alors cette solution voisine devient la solution actuelle. Sinon, une nouvelle solution voisine est définie avec la méthode SWAP et son temps d’attente est calculé. Ce procédé est réitéré pour un temps de calcul donné ou un nombre d’itérations donné. Nous implémentons donc une recherche locale avec une seule solution voisine testée. Cette démarche très basique permet à l’entreprise KLS de pouvoir observer les gains que nous obtenons très rapidement par rapport aux performances actuelles des entrepôts étudiés.

La figure 37 résume la méthodologie utilisée pour l’obtention du temps d’attente.

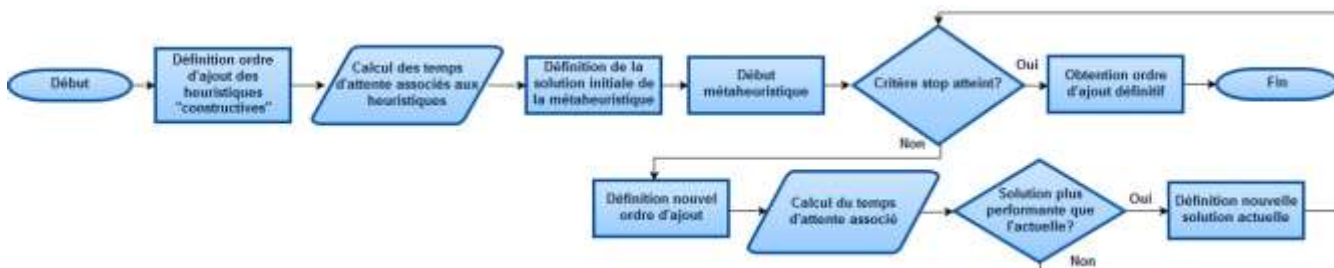


Figure 37 : Méthodologie utilisée pour définir l’ordre d’ajout des commandes dans un système « Drive »

5.4 Etudes de cas

Nous avons testé notre modèle en utilisant les données des entrepôts des études de cas précédentes. N'ayant pas de données issues de cas réels de type « drive », nous avons choisi plusieurs configurations de données. Nous avons comparé notre stratégie de séquencement des commandes avec l'ordre d'ajout des commandes des entreprises considérées.

Nous rappelons que le temps de préparation de commandes est défini comme étant $C = W + P$. Le temps P de collecte étant le même quel que soit l'heuristique testée, le gain se porte sur le temps d'attente de l'opérateur, W . Nous considérons, dans les expériences numériques ci-dessous, l'hypothèse réaliste pour laquelle le temps pour que le carrousel effectue la rotation d'un plateau au plateau adjacent est égal au temps de collecte d'une ligne de commande ($r=p=1$).

Le tableau ci-dessous récapitule le temps d'attente de chacune des instances sous les différentes heuristiques étudiées. Nous représentons ici uniquement le temps d'attente car c'est sur ce temps d'attente que se jouent les performances. Les instances 1 et 2 proviennent de l'entreprise Suisse (8 commandes sont simultanément présentes sur la table de préparation), tandis que les autres instances proviennent de l'Hôpital Français 2 (10 commandes sont simultanément présentes sur la table de préparation). Concernant les données issues de l'Hôpital Français 2, les données de plusieurs jours sont regroupées afin d'obtenir des instances réalistes dans le cadre d'un système drive (nombre de commandes suffisamment grand, supérieur à 100 commandes). Le tableau 33 ci-dessous montre les temps d'attente des différentes heuristiques de base décrites ci-dessus en section 5.3.1.

Tableau 33 : Temps d'attente sous les différentes heuristiques de base

Instance	NB Commandes	NB lignes de commandes	NB carousels	Stratégie actuelle	H1-1	H1-2	H2-1	H2-2	H3-1	H3-2	Densité
Jour 1	36	198	4	356	327	320	375	316	462	368	301
Jour 2	58	189	4	661	722	781	727	667	954	531	533
Jour 3	111	698	2	344	375	484	341	446	500	223	185
Jour 4	122	206	2	814	642	569	575	528	1 124	237	257
Jour 5	208	367	2	1014	997	948	993	1000	2 134	345	303
Jour 6	108	2614	2	113	211	246	250	260	195	179	153
Jour 7	200	8472	2	202	189	211	186	186	177	152	185

Nous observons que les heuristiques de base ayant les meilleures performances sont l'heuristique H3-2 (celle incluant la commande ayant la plus faible distance) et l'heuristique Densité. Nous incluons, à ces deux heuristiques de base que nous jugeons les plus performantes, les résultats

associés aux deux groupes d'heuristiques, présentés sur le tableau 34. Sur ce dernier, nous nommons les deux dernières colonnes GC1 et GC2 qui réfèrent au meilleur résultat des heuristiques GC-x testées précédemment.

Tableau 34 : Temps d'attente des groupes d'heuristiques

Instance	NB Commandes	NB lignes de commandes	Nb moyen ligne	NB carousels	Stratégie actuelle	H3-2	Densité	GC1	GC2
Jour 1	36	198	5,5	4	356	368	301	320	332
Jour 2	58	189	3,3	4	661	531	533	619	531
Jour 3	111	698	6,3	2	344	223	185	290	199
Jour 4	122	206	1,7	2	814	237	257	569	197
Jour 5	208	367	1,8	2	1014	345	303	930	335
Jour 6	108	2614	24,2	2	113	179	153	130	93
Jour 7	200	8472	42,4	2	202	152	185	142	137

A partir du tableau 34, nous observons que le groupe d'heuristiques 2 ainsi que l'heuristique Densité donnent de meilleures performances que les autres heuristiques ainsi que la stratégie actuelle. A partir de ses heuristiques, nous obtenons une solution améliorée qui est la solution initiale de notre métaheuristique présentée en section 5.3.2. Les résultats de cette métaheuristique présentant les temps d'attente de chacune des instances sont présentés dans le tableau 35 et une représentation graphique des gains obtenus est présentée sur la figure 38. Concernant la représentation graphique, nous ne montrons que les heuristiques ayant les meilleures performances : l'heuristique densité, GC2 ainsi que les résultats de la métaheuristique. Le calcul du résultat de la métaheuristique est fait comme précédemment (section 3.4 et section 4.4) avec le calcul de la Valeur Fitness Moyenne (VFM) : la métaheuristique est lancée 10 fois et la VFM correspond à la moyenne de ces 10 résultats.

Tableau 35 : Temps d'attente avec utilisation de la métaheuristique

Instance	NB Commandes	NB lignes de commandes	Nb moyen ligne	NB carousels	Stratégie actuelle	Densité	GC2	VFM
Jour 1	36	198	5,5	4	356	301	332	229,6
Jour 2	58	189	3,3	4	661	533	531	417,6
Jour 3	111	698	6,3	2	344	185	199	158,3
Jour 4	122	206	1,7	2	814	257	197	187,2
Jour 5	208	367	1,8	2	1014	303	335	281
Jour 6	108	2614	24,2	2	113	153	93	63,4
Jour 7	200	8472	42,4	2	202	185	137	100,4

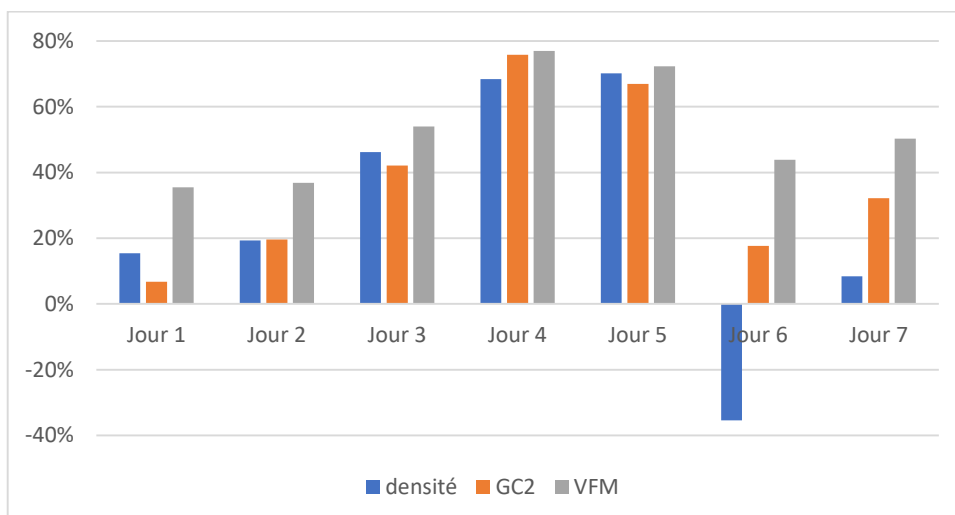


Figure 38 : Gains en temps d'attente

Nous observons donc des gains conséquents pour le temps d'attente que nous cherchons à minimiser. Ce gain, par rapport à la stratégie actuelle, est significatif. Il est compris entre 36% et 77% en comparant la VFM et la stratégie actuelle. Nous observons donc que le temps d'attente peut être grandement réduit. Nous voyons bien, grâce à l'utilisation d'heuristiques ainsi qu'une métaheuristique basique, que la marge de gain concernant le temps d'attente est conséquente.

Nous regardons maintenant les gains en termes de temps d'attente et de préparation de commandes dans le tableau 36 ci-dessous.

Tableau 36 : Gain en temps d'attente et temps de préparation de commandes

Instance	Gain W			Gain C		
	Densité	GC1	VFM	Densité	GC1	VFM
Jour 1	15%	7%	36%	10%	4%	23%
Jour 2	19%	20%	37%	15%	15%	29%
Jour 3	46%	42%	54%	15%	14%	18%
Jour 4	68%	76%	77%	55%	60%	61%
Jour 5	70%	67%	72%	51%	49%	53%
Jour 6	-35%	18%	44%	-1%	1%	2%
Jour 7	8%	32%	50%	0%	1%	1%

Nous voyons bien, dans le tableau ci-dessus, l'influence du temps de picking sur le temps de préparation de commandes. Nous observons que lorsque le nombre de ligne de commandes est faible (jour 1 à jour 5), les gains en temps d'attente sont conséquents. Pour les instances correspondant aux jours 1 à 5, le nombre de ligne de commandes est assez faible (de 1.7 à 5.5 lignes par commandes), et donc les gains en temps de préparation de commandes seront encore conséquents (allant de 23% à 61%). Cependant lorsque le nombre de lignes de commandes

devient très élevé (le nombre de lignes de commandes par commande moyen est de 24.2 pour le jour 6 et 42.4 pour le jour 7) nous observons que le temps d'attente ne représente qu'une infime partie du temps de préparation et donc les gains associés seront très faibles.

Au cours de ce chapitre nous avons étudié un système particulièrement adapté aux nouveaux modes de distribution et de vente par internet : le système que nous avons dénommé « Drive ». Ce type de système est caractérisé par de multiples machines sur lesquelles un grand nombre de commandes est collecté. Lorsqu'une d'entre elles est terminée, elle est immédiatement remplacée par une nouvelle commande à collecter. Nous avons fourni un algorithme permettant le calcul du temps d'attente pour la collecte d'un portefeuille de commandes et avons étudié la marge de gain possible grâce au développement d'heuristiques et d'une métaheuristique. Ces gains nous montrent que le temps d'attente peut être amélioré de manière significative et justifie donc l'intérêt pratique de ce type de modèle pour une amélioration de la productivité.

Chapitre 6 Conclusion et perspectives de recherche

Au cours de ce manuscrit, nous avons étudié les problèmes d'Order Batching présents dans le cas de préparation de commandes effectuée sur des stockeurs automatisés de type VLM et Carousel. Nous avons étudié deux contextes industriels, le Batching et le Rolling Batching. Afin de permettre une productivité accrue pour les entreprises disposant d'une zone de préparation de commandes, nous avons développé une méthodologie et défini des méthodes de résolution permettant une optimisation du temps de préparation de commandes. Ce sujet fut choisi pour répondre au manque de littérature concernant la préparation se faisant sur des stockeurs automatisés de type VLMs ou carousels, pourtant en plein développement dans les entrepôts actuels. Ainsi nous avons montré que le temps de préparation pouvait être significativement réduit afin de collecter un portefeuille de commandes sur des VLMs ou des Carousels grâce à un regroupement ou un séquençement efficient des commandes.

Pour le regroupement de commandes en lots complets ou Batching, nous avons tout d'abord défini les temps de préparation lorsque la collecte s'effectue sur un stockeur. Nous sommes ensuite passés au cas plus complexe, avec l'utilisation de plusieurs stockeurs, et la prise en compte de la recherche de plateaux effectuée en temps masqué.

Lorsque la zone de préparation est équipée de VLMs, nous avons observé que le temps de préparation de commandes dépendait du nombre de plateaux visités par le VLM en visitant le plus. En minimisant ce nombre grâce à de nouvelles combinaisons de commandes, nous diminuons les périodes où les produits restant à collecter sont dans un seul stockeur. De ce fait, ce critère réduit les temps d'attente de l'opérateur, lui permettant donc d'être plus productif. Les gains entre la stratégie actuelle de deux entreprises et notre modèle sont significatifs.

Lorsque la zone de préparation est équipée de Carousels, le temps de préparation va dépendre à la fois du nombre de plateaux à visiter ainsi que de la distance à parcourir par les différents carousels. Nous avons choisi de minimiser la distance que parcourt le carousel qui tourne le plus pour chacun des lots. En faisant ainsi, outre le gain en durabilité des machines, nous assurons que

tous les produits sont collectés en faisant tourner le moins possible le carrousel effectuant le plus de rotation sur chacun des lots. Nous observons des gains significatifs en termes de temps de préparation pour l'hypothèse réaliste concernant les temps de picking.

Nous résumons les différents objectifs utilisés ainsi que les gains moyens de chacune des études de cas dans le tableau récapitulatif ci-dessous. Pour le gain moyen, nous effectuons la moyenne des gains C sur l'ensemble des lots effectués sous l'hypothèse réaliste concernant les temps de picking.

Tableau 37 : Récapitulatif des critères utilisés ainsi que des gains associés lors du regroupement de commandes en lots

Type de stockeur	Nb stockeurs	Critère utilisé	Gain moyen en temps de préparation
VLM	1	Minimiser $\sum_{b \in Lot} \text{Nombre de plateaux visités pour le lot } b$	Entreprise Suisse :22% Hôpital Français 1 :27%
	>1	Minimiser $\sum_{b \in Lot} \max_{v \in VLM} \text{Nombre de plateaux visités par VLM } v \text{ pour le lot } b$	Entreprise Suisse :24% Hôpital Français 1 :32%
Carrousel	1	Minimiser $\sum_{b \in Lot} \text{Distance parcourue pendant le lot } b$	Hôpital Français 2 :18%
	>1	Minimiser $\sum_{b \in Lot} \max_{u \in Carousels} \text{Distance parcourue par carrousel } u \text{ pour le lot } b$	Entreprise Suisse :19% Hôpital Français 2 :17%

Nous observons que les gains en temps de préparation sont loin d'être négligeables à la vue des chiffres montrés ci-dessus. Nous avons poussé notre étude en montrant que plus le nombre de commandes que nous regroupons est élevé, plus les gains sont élevés. Ceci est dû au plus grand choix dont nous disposons pour combiner plus efficacement les commandes, et ainsi trouver des commandes similaires. De même nous avons observé que le nombre de lignes de commandes par commande est un facteur important. Moins de lignes de commandes induisent de plus grands gains. En effet, avec un nombre de lignes de commandes plus faible, nous disposons de plus de commandes potentiellement similaires tandis qu'un grand nombre de lignes de commandes, va réduire le nombre de commandes potentiellement similaires.

Enfin, nous avons permis une réalisation pratique de notre problème par le développement d'un module au sein de l'entreprise KLS Logistic appelé « SCOP » utilisant une métaheuristique de type algorithme génétique afin de résoudre le problème de manière approchée et dont les résultats sous une minute de calcul prouvent leur efficacité. Nous obtenons des résultats prometteurs permettant un regroupement de commandes très satisfaisant.

Concernant le système de collecte que nous avons dénommé « Rolling Batching », nous avons défini au cours de ce manuscrit un algorithme permettant de calculer le temps d'attente en fonction d'un ordre d'ajout des commandes donné. Ainsi utilisant cet algorithme, nous avons évalué les performances de différentes heuristiques. Ces heuristiques permettent l'ajout des commandes en mode dynamique et en fonction de caractéristiques des commandes étant en attente de préparation. Ces heuristiques nous fournissent la solution initiale d'une métaheuristique. Afin d'observer les potentiels marges de gains par rapport au séquençement actuel, nous avons testé une métaheuristique basique. Le gain en temps d'attente sur les instances considérées s'élève à 53% tandis que le gain moyen en temps de préparation de commandes est de 27%. Ces chiffres confirment l'importance d'un séquençement de commandes dans de tels systèmes. De plus, les heuristiques que nous employons permettent la prise en compte des commandes arrivant au fil de l'eau, et permet un séquençement des commandes performant.

Nos travaux effectués, étant pionniers sur le regroupement de commandes en lots et de séquençement des commandes au sein d'une zone de préparation équipée de VLMs ou Carousels, ouvrent la voie vers diverses perspectives de recherche.

- Tout d'abord une généralisation de la définition des temps de préparation de commandes aux divers types de stockeurs automatisés serait intéressante. Cela permettrait d'obtenir, pour chacun d'entre eux, un critère pertinent pouvant être utilisé par les divers managers pour une préparation de commandes efficiente. Les problèmes abordés seraient proches de ceux présents dans ce manuscrit pour les systèmes disposant de racks mobiles, ainsi que de stockeurs à double ouverture qui permettent d'avoir deux plateaux simultanément présents pour la collecte des produits.
- De plus, concernant les méthodes de résolutions approchées, d'autres métaheuristiques pourraient être testées. En effet, pour le cas VLM, nous avons implémenté au premier abord l'algorithme génétique comme métaheuristique et avons testé, au travers d'un stage master d'autres métaheuristiques. Nous avons montré que le recuit simulé donnait les meilleurs résultats. Il serait intéressant de généraliser ses résultats pour le cas Carousel et pour le séquençement de commandes au chapitre 5. De plus d'autres métaheuristiques à base de population ou de voisinage pourrait être étudiées, et permettraient potentiellement d'accroître les gains obtenus. Il serait donc intéressant d'élargir à d'autres méthodes approchées.

- Enfin les modèles que nous avons utilisés ne prennent en compte ni les instants d'arrivées des commandes, ni les deadlines des commandes. Il serait intéressant d'aborder le problème de regroupement et de séquençement des commandes tout en incluant ces contraintes d'arrivée et de deadlines. Les heuristiques développées dans le chapitre 5 peuvent immédiatement s'appliquer au cas d'arrivées au fil de l'eau des commandes. Par contre il faudrait les faire évoluer pour tenir compte des due dates des commandes. De plus divers objectifs pourraient être testés tel que la minimisation des retards des commandes (critère intéressant pour les entreprises ayant de fortes pénalités liées aux retards) ou bien la minimisation des coûts, en incluant les coûts de stockage et de préparation ainsi que les pénalités de retard.

Bibliographie

- Armstrong, R. D., Cook, W. D., & Saipe, A. L. (1979). Optimal batching in a semi-automated order picking system. *Journal of the operational research society*, 711–720.
- Azadnia, A. H., Taheri, S., Ghadimi, P., Mat Saman, M. Z., & Wong, K. Y. (2013). Order batching in warehouses by minimizing total tardiness: a hybrid approach of weighted association rule mining and genetic algorithms. *The Scientific World Journal*, 2013.
- Bartholdi III, J. J., & Platzman, L. K. (1986). Retrieval strategies for a carousel conveyor. *IIE transactions*, 18(2), 166–173.
- Bartholdi, J. J., & Hackman, S. T. (2011). *Warehouse and Distribution Science: Release 0.95*, The Supply Chain and Logistics Institute, School of Industrial and Systems Engineering. Atlanta.
- Boysen, N., Briskorn, D., & Emde, S. (2017a). Parts-to-picker based order processing in a rack-moving mobile robots environment. *European Journal of Operational Research*, 262(2), 550–562.
- Boysen, N., Briskorn, D., & Emde, S. (2017b). Sequencing of picking orders in mobile rack warehouses. *European Journal of Operational Research*, 259(1), 293–307.
- Bukchin, Y., Khmelnsky, E., & Yakuel, P. (2012). Optimizing a dynamic order-picking process. *European Journal of Operational Research*, 219(2), 335–346.
- Burinskiene, A. (2015). Optimising Forklift Activities in Wide-Aisle Reference Warehouse. *International Journal of Simulation Modelling*, 14(4).
- Carlo, H. J., & Vis, I. F. (2012). Sequencing dynamic storage systems with multiple lifts and shuttles. *International Journal of Production Economics*, 140(2), 844–853.

- Caron, F., Marchet, G., & Perego, A. (2000). Layout design in manual picking systems: a simulation approach. *Integrated Manufacturing Systems*, 11(2), 94–104.
- Chan, F. T., & Chan, H. K. (2011). Improving the productivity of order picking of a manual-pick and multi-level rack distribution warehouse through the implementation of class-based storage. *Expert Systems with Applications*, 38(3), 2686–2700.
- Chen, T.-L., Cheng, C.-Y., Chen, Y.-Y., & Chan, L.-K. (2015). An efficient hybrid algorithm for integrated order batching, sequencing and routing problem. *International Journal of Production Economics*, 159, 158–167.
- Corley Jr, H. W., & Roberts, S. D. (1972). A partitioning problem with applications in regional design. *Operations Research*, 20(5), 1010–1019.
- De Koster, R., Dekker, R., Roodbergen, K. J., & Van Kalleveen, H. (2004). Improving order-picking response time at Ankor's warehouse. *Interfaces*, 34(4), 303–313.
- De Koster, R., Chen, C.-M., Gong, Y., & Van Nunen, J. A. (2010). A flexible evaluative framework for order picking systems. *Production and Operations Management*, 19(1), 70–82.
- De Koster, R., Le-Duc, T., & Roodbergen, K. J. (2007). Design and control of warehouse order picking: A literature review. *European Journal of Operational Research*, 182(2), 481–501.
- De Koster, R., & Poort, E. V. D. (1998). Routing orderpickers in a warehouse: a comparison between optimal and heuristic solutions. *IIE transactions*, 30(5), 469–480.
- Drury, J. (1988). Towards more efficient order picking. *IMM monograph*, 1.
- Dukic, G., Opetuk, T., & Lerher, T. (2015). A throughput model for a dual-tray Vertical Lift Module with a human order-picker. *International journal of production economics*, 170, 874–881.
- Ekren, B. Y. (2011). Performance evaluation of AVS/RS under various design scenarios: a case study. *The International Journal of Advanced Manufacturing Technology*, 55(9), 1253–1261.
- Ekren, B. Y., & Heragu, S. S. (2011). Simulation based performance analysis of an autonomous vehicle storage and retrieval system. *Simulation Modelling Practice and Theory*, 19(7), 1640–1650.

- Fukunari, M., & Malmberg, C. J. (2008). An efficient cycle time model for autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 46(12), 3167–3184.
- Gademann, A., Van Den Berg, J. P., & Van Der Hoff, H. H. (2001). An order batching algorithm for wave picking in a parallel-aisle warehouse. *IIE transactions*, 33(5), 385–398.
- Gademann, N., & Velde, S. (2005). Order batching to minimize total travel time in a parallel-aisle warehouse. *IIE transactions*, 37(1), 63–75.
- Goetschalckx, M., & Ashayeri, J. (1989). Classification and design of order picking. *Logistics World*, 2(2), 99–106.
- Gu, J., Goetschalckx, M., & McGinnis, L. F. (2010). Research on warehouse design and performance evaluation: A comprehensive review. *European Journal of Operational Research*, 203(3), 539–549.
- Gue, K. R., & Meller, R. D. (2009). Aisle configurations for unit-load warehouses. *IIE Transactions*, 41(3), 171–182.
- Hall, R. W. (1993). Distance approximations for routing manual pickers in a warehouse. *IIE transactions*, 25(4), 76–87.
- Henn, S. (2012). Algorithms for on-line order batching in an order picking warehouse. *Computers & Operations Research*, 39(11), 2549–2563.
- Henn, S., Koch, S., & Wäscher, G. (2012). Order batching in order picking warehouses: a survey of solution approaches. In *Warehousing in the Global Supply Chain* (p. 105–137). Springer. Consulté à l'adresse http://link.springer.com.gaeInomade-2.grenet.fr/chapter/10.1007/978-1-4471-2274-6_6
- Henn, S., & Schmid, V. (2013). Metaheuristics for order batching and sequencing in manual order picking systems. *Computers & Industrial Engineering*, 66(2), 338–351.
- Henn, S., & Wäscher, G. (2012). Tabu search heuristics for the order batching problem in manual order picking systems. *European Journal of Operational Research*, 222(3), 484–494.

- Heragu*, S. S., Du, L., Mantel, R. J., & Schuur, P. C. (2005). Mathematical model for warehouse design and product allocation. *International Journal of Production Research*, 43(2), 327–338.
- Heskett, J. L. (1963). Cube-per-order index-a key to warehouse stock location. *Transportation and distribution Management*, 3(1), 27–31.
- Hong, S., Johnson, A. L., & Peters, B. A. (2012). Batch picking in narrow-aisle order picking systems with consideration for picker blocking. *European Journal of Operational Research*, 221(3), 557–570.
- Hsu, C.-M., Chen, K.-Y., & Chen, M.-C. (2005). Batching orders in warehouses by minimizing travel distance with genetic algorithms. *Computers in Industry*, 56(2), 169–178.
- Iraji, M. S., Jafari, S., & Sookhtsaraei, R. (s. d.). Algorithms for On-line Order Batching for Minimized Total Tardiness in an Order-Picking Warehouse.
- Jarvis, J. M., & McDowell, E. D. (1991). Optimal product layout in an order picking warehouse. *IIE transactions*, 23(1), 93–102.
- Kallina, C., & Lynn, J. (1976). Application of the cube-per-order index rule for stock location in a distribution warehouse. *Interfaces*, 7(1), 37–46.
- Kim, J.-Y., & Kim, Y.-D. (1995). Graph theoretic heuristics for unequal-sized facility layout problems. *Omega*, 23(4), 391–401.
- Koch, S., Wäscher, G., & others. (2011). A grouping genetic algorithm for the order batching problem in distribution warehouses. Otto-von-Guericke University Magdeburg, Faculty of Economics and Management.
- Kuo, P.-H., Krishnamurthy, A., & Malmborg, C. J. (2007). Design models for unit load storage and retrieval systems using autonomous vehicle technology and resource conserving storage and dwell point policies. *Applied Mathematical Modelling*, 31(10), 2332–2346.
- Kuo, P.-H., Krishnamurthy, A., & Malmborg, C. J. (2008). Performance modelling of autonomous vehicle storage and retrieval systems using class-based storage policies. *International Journal of Computer Applications in Technology*, 31(3-4), 238–248.

- Laajili, E., Lenoble, N., Frein, Y. (2017). Order batching optimization in automated warehouses with metaheuristics. 20th IFAC World Congress, Toulouse (France), July 9 to July 14
- Lambert, D. M., Cooper, M. C., & Pagh, J. D. (1998). Supply chain management: implementation issues and research opportunities. *The international journal of logistics management*, 9(2), 1–20.
- Lambert, D. M., Stock, J. R., & Ellram, L. M. (1998). Fundamentals of logistics management. McGraw-Hill/Irwin.
- Lawler, E. L., Lenstra, J. K., & Rinnooy Kan, A. H. (1985). The traveling salesman problem.
- Le-Duc, T. (2005). Design and control of efficient order picking processes.
- Le-Duc, T., & De Koster, R. (2005). Determining number of zones in a pick-and-pack orderpicking system.
- Le-Duc, T., & De Koster, R. (2005). Layout optimization for class-based storage strategy warehouses. *Supply Chain Management-European Perspective*, 191–214.
- Lee, S.-D., & Kuo, Y.-C. (2008). Exact and inexact solution procedures for the order picking in an automated carousal conveyor. *International Journal of Production Research*, 46(16), 4619–4636.
- Lenoble, N., Frein, Y., Hammami, R. (2016). Optimization of order batching in a picking system with a Vertical Lift Module. 6th International Conference on Information Systems, Logistics and Supply Chain, June 6 to June 10
- Lenoble, N., Frein, Y., Hammami, R. (2017a). Optimization of order batching in a picking system with a Carousels. 20th IFAC World Congress, Toulouse (France), July 9 to July 14
- Lenoble, N., Frein, Y., Hammami, R. (2017b). Order batching in an automated warehouse with several Vertical Lift Modules: Optimization and experiments with real data, *European Journal of Operational Research*, 3rd review in progress.
- Lerher, T. (2013). Modern automation in warehousing by using the shuttle-based technology, Arent, D.; Freebush, M. *Automation Systems of the 21st Century: New Technologies, Applications and Impacts on the Environment & Industrial Processes*, 51–86.

- Lerher, T. (2016). Travel time model for double-deep shuttle-based storage and retrieval systems. *International Journal of Production Research*, 54(9), 2519–2540.
- Lerher, T., Edl, M., & Rosi, B. (2014). Energy efficiency model for the mini-load automated storage and retrieval systems. *International Journal of Advanced Manufacturing Technology*, 70.
- Lerher, T., Ekren, B. Y., Sari, Z., & Rosi, B. (2016). Method for evaluating the throughput performance of shuttle based storage and retrieval systems/Metoda za procjenu protodne performance automatiziranih skladisnih sustava s vozilima. *Tehnicki Vjesnik-Technical Gazette*, 23(3), 715–724.
- Lerher, T., Ekren, Y. B., Sari, Z., & Rosi, B. (2015). Simulation analysis of shuttle based storage and retrieval systems.
- Li, Z., & Zhou, Z. (2013). An effective batching method based on the artificial bee colony algorithm for order picking. In *Natural Computation (ICNC), 2013 Ninth International Conference on* (p. 386–391). IEEE. Consulté à l'adresse <http://ieeexplore.ieee.org/gaelnomade-2.grenet.fr/abstract/document/6818006/>
- Litvak, N., & Adan, I. (2002). On a class of order pick strategies in paternosters. *Operations Research Letters*, 30(6), 377–386.
- Litvak, N., Adan, I., Wessels, J., & Zijm, W. H. M. (2001). Order picking in carousel systems under the nearest item heuristic. *Probability in the Engineering and Informational Sciences*, 15(2), 135–164.
- Litvak, N., & Vlasious, M. (2010). A survey on performance analysis of warehouse carousel systems. *Statistica Neerlandica*, 64(4), 401–447.
- Malmborg, C. J. (2002). Conceptualizing tools for autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 40(8), 1807–1822.
- Malmborg, C. J., Balachandran, S., & Kyle, D. M. (1986). A model based evaluation of a commonly used rule of thumb for warehouse layout. *Applied Mathematical Modelling*, 10(2), 133–138.

- Malmborg, C. J., & Bhaskaran, K. (1989). Optimal storage assignment policies for multiaddress warehousing systems. *IEEE Transactions on Systems, Man, and Cybernetics*, 19(2), 197–204.
- Malmborg, C. J., & Bhaskaran, K. (1987). On the optimality of the cube per order index for conventional warehouses with dual command cycles. *Material Flow*, 4(3), 169–175.
- Malmborg, C. J., & Bhaskaran, K. (1990). A revised proof of optimality for the cube-per-order index rule for stored item location. *Applied Mathematical Modelling*, 14(2), 87–95.
- Mantel, R. J., Schuur, P. C., & Heragu, S. S. (2007). Order oriented slotting: a new assignment strategy for warehouses. *European Journal of Industrial Engineering*, 1(3), 301–316.
- Manzini, R., Gamberi, M., & Regattieri, A. (2006). Design and control of an AS/RS. *The International Journal of Advanced Manufacturing Technology*, 28(7), 766–774.
- Marchet, G., Melacini, M., Perotti, S., & Tappia, E. (2012). Analytical model to estimate performances of autonomous vehicle storage and retrieval systems for product totes. *International Journal of Production Research*, 50(24), 7134–7148.
- Marchet, G., Melacini, M., Perotti, S., & Tappia, E. (2013). Development of a framework for the design of autonomous vehicle storage and retrieval systems. *International Journal of Production Research*, 51(14), 4365–4387.
- Marvel, J. H., Shell, R. L., & Weckman, G. R. (2001). An application of heuristic algorithms for determining inventory location in a distribution warehouse. *International Journal of Industrial Engineering*, 8, 5–15.
- Matusiak, M., de Koster, R., Kroon, L., & Saarinen, J. (2014). A fast simulated annealing method for batching precedence-constrained customer orders in a warehouse. *European Journal of Operational Research*, 236(3), 968–977.
- ELA European Logistics Association/AT Kearney Management Consultants, 2004. Differentiation for Performance, Deutscher Verkehrs-Verlag GmbH, Hamburg.
- Meller, R. D., & Gau, K.-Y. (1996a). Facility layout objective functions and robust layouts. *International journal of production research*, 34(10), 2727–2742.

- Meller, R. D., & Gau, K.-Y. (1996b). The facility layout problem: recent and emerging trends and perspectives. *Journal of manufacturing systems*, 15(5), 351–366.
- Meller, R. D., & Klote, J. F. (2004). A throughput model for carousel/VLM pods. *IIE transactions*, 36(8), 725–741.
- Pan, J. C.-H., Shih, P.-H., & Wu, M.-H. (2012). Storage assignment problem with travel distance and blocking considerations for a picker-to-part order picking system. *Computers & Industrial Engineering*, 62(2), 527–535.
- Pazour, J. A., & Meller, R. D. (2013). The impact of batch retrievals on throughput performance of a carousel system serviced by a storage and retrieval machine. *International Journal of Production Economics*, 142(2), 332–342.
- Petersen, C. G. (1997). An evaluation of order picking routing policies. *International Journal of Operations & Production Management*, 17(11), 1098–1111.
- Petersen, C. G., & Aase, G. (2004). A comparison of picking, storage, and routing policies in manual order picking. *International Journal of Production Economics*, 92(1), 11–19.
- Petersen, C. G., Aase, G. R., & Heiser, D. R. (2004). Improving order-picking performance through the implementation of class-based storage. *International Journal of Physical Distribution & Logistics Management*, 34(7), 534–544.
- Roodbergen, K. J., & De Koster, R. (2001). Routing order pickers in a warehouse with a middle aisle. *European Journal of Operational Research*, 133(1), 32–43.
- Roodbergen, K. J., & Koster, R. D. (2001). Routing methods for warehouses with multiple cross aisles. *International Journal of Production Research*, 39(9), 1865–1883.
- Roodbergen, K. J., & Vis, I. F. (2009). A survey of literature on automated storage and retrieval systems. *European journal of operational research*, 194(2), 343–362.
- Roodbergen, K.-J. (2001). Layout and routing methods for warehouses.
- Rosi, B., Grasic, L., Dukic, G., Opetuk, T., & Lerher, T. (2016). Simulation-based performance analysis of automated single-tray vertical lift module. *International journal of simulation modelling*, 15(1), 97–108.

- Shouman, M. A., Khater, M., & Boushaala, A. (2007). Comparisons of order picking routing methods for warehouses with multiple cross aisles.
- Shouman, M. A., Khater, M., & Boushaala, A. A. (2005). Comprehensive survey and classification scheme of warehousing systems. In *Proceedings of the 2005 International Conference on Simulation and Modeling V. Kachitvichyanukul, U. Purintrapiban, P. Utayopas, eds., Nakompathon, Thailand* (Vol. 1).
- Tompkins, J. A., White, J. A., Bozer, Y. A., & Tanchoco, J. M. A. (2010). *Facilities Planning*. John Wiley & Sons.
- Tsai, C.-Y., Liou, J. J., & Huang, T.-M. (2008). Using a multiple-GA method to solve the batch picking problem: considering travel distance and order due time. *International Journal of Production Research*, 46(22), 6533–6555.
- Van Den Berg, J. P., & Gademann, A. (2000). Simulation study of an automated storage/retrieval system. *International Journal of Production Research*, 38(6), 1339–1356.
- Williams, E. J., & Fosso, K. A. (1995). A simulation modeler's view of a carousel. In *Proceedings of the 27th conference on Winter simulation* (p. 925–929). IEEE Computer Society.
- Yang, M. (1988). Analysis and optimization of class-based dedicated storage systems. Georgia Institute of Technology.
- Zhao, X., He, L., & Hu, J. (2012). Picking Path Optimization for Two-way Picking Parallel Multi-carousels System. *Advances in Information Sciences and Service Sciences*, 4(21).
- Zollinger, H. (2014). AS/RS application, benefits and justification in comparison to other storage methods: A white paper. Automated Storage Retrieval Systems Production Section of the Material Handling Industry of America,

Annexe

Annexe. Calcul du temps d'attente dans un système drive

Pour des raisons pédagogiques, nous illustrons la démarche adoptée pour la collecte d'un portefeuille de commandes par exemple composé de 4 commandes. Nous considérons de plus que la zone de préparation est caractérisée par :

- La table de préparation permettant de collecter simultanément 3 commandes.
- Elle est équipée de 3 Carousels ayant chacun 5 plateaux.

Nous allons reprendre et illustrer sur cet exemple l'algorithme présenté en section 5.2.2.

Soit t l'événement indiquant que la t -ième commande est préparée. $t=0$ sera l'évènement où les premières commandes sont mises en préparation. $t=N$ indiquera la fin de la préparation avec la N -ième commande préparée.

- Soit les commandes suivantes :

	Commande 1	Commande 2	Commande 3	Commande 4
Carousel 1	1 0 0 1 0	0 1 1 0 0	1 1 0 0 0	0 1 0 0 1
Carousel 2	0 1 1 0 0	0 1 0 0 0	1 0 0 0 1	0 1 0 0 1
Carousel 3	0 0 1 1 1	0 1 0 0 0	0 0 0 1 0	1 0 1 0 0

Le tableau précédent nous indique par un 1 si un plateau d'un carousel doit être visité par une commande. Par exemple, la commande 1 visite les plateaux 1 et 4 du stockeur 1. Soit $L_{n k u}$ le nombre de lignes à collecter sur le niveau k du carousel u par la commande n . Nous introduisons le paramètre $E_{n k u}^t$ égal à 1 si la commande n doit encore visiter le plateau k du carousel u à l'instant où se termine la préparation de la t -ième commande.

Soit le vecteur d'ajout des commandes $A = \{1,2,3,4\}$ impliquant que l'on ajoute sur la table la commande 1 puis la 2, ensuite la 3 et enfin la 4. La table de préparation permettant de collecter 3 commandes simultanément, les commandes 1,2 et 3 sont donc introduites ensembles à $t=0$.

Soit Y_n^t le booléen égal à 1 si la commande n est sur la table de préparation à l'instant où la t -ième commande se termine.

Initialisation des paramètres

Nous initialisons les paramètres :

- Le nombre de commandes simultanément en préparation : $B = 3$
- Nous initialisons le nombre de carousels et de plateaux par carousel $U = 3$ et $K_1 = K_2 = K_3 = 5$
- Nous définissons l'emplacement de chacun des carousels à 1. Si V_u^t l'emplacement du carousel u à l'instant où se termine la t -ième commande, à l'état initial on a donc $V_u^0 = 1 \forall u \in v$.
- Soit U^t le carousel sur lequel reprend la collecte après la t -ième commande préparée. A l'état initial, $U^0 = 1$
- Les 4 commandes présentées ci-dessus sont incluses dans le portefeuille de commandes, avec le statut « En attente » à l'instant initial.

Ajout des premières commandes sur la table de préparation

- Les 3 premières commandes de l'ordre d'ajout, i.e. les commandes 1, 2 et 3 ont leur statut qui changent pour « En préparation » $\forall i \in \{1 \dots B\} Y_{A(i)}^0 = 1$. Ici $Y_1^0 = Y_2^0 = Y_3^0 = 1$

Préparation : Tant qu'il reste des commandes sur la table de préparation

- Collecte jusqu'à la fin de la préparation d'une commande
 - **Identification de la commande finissant en premier parmi celles de la table**

Afin d'identifier la commande finissant en premier, parmi celles « en préparation », plusieurs sous étapes sont nécessaires :

Soit $D_{n u}^t$ la distance qu'il reste à parcourir pour finir la commande n , sur le carrousel u entre t et $t+1$. Cela revient à regarder pour chacune des commandes en préparation, la distance que celle-ci doit parcourir pour être entièrement collectée sur chacun des carousels, à partir de l'emplacement de chacun des carousels. A $t=0$: $D_{n c}^0 = \max_{k \in \{1 \dots K_u\}} \left((k - V_u^0) * E_{n k u}^0 \right) \text{Mod}(K_u)$; ici

	$D_{n c}^0$		
Commande	1	2	3
Carousel 1	3	2	1
Carousel 2	2	1	4
Carousel 3	4	1	3

On détermine les plateaux devant être vus par les commandes de la table : Soit $N_{k u}^t$ le booléen égal à 1 si le plateau k du carousel u doit être visité par les commandes en cours de préparation entre t et $t+1$. Cela revient à avoir un tableau étant l'union des tableaux des visites des commandes en cours de préparation.

$$N_{k c}^0 = \sum_{n=1}^N E_{n k c}^0 * Y_n^0$$

	$N_{k u}^0$				
Carousel 1	1	1	1	1	0
Carousel 2	1	1	1	0	1
Carousel 3	0	1	1	1	1

Nous regardons maintenant pour chacune des commandes en préparation le nombre de plateaux à visiter sur chacun des carousels. Sachant qu'il y a alternance de visite des carousels, la commande finissant le plus tôt parmi celles présentes sur la table de préparation à l'évènement t , sera celle ayant son nombre maximal de plateaux à visiter sur les différents carousels le plus faible.

Soit $NP_{n u}^t$ à l'évènement t , le nombre de plateaux à visiter sur le carousel u pour terminer la commande n . Ce paramètre par carousel, par commande, indique le nombre de plateaux vus par le carousel considéré entre le plateau V_u^0 et $V_u^0 + D_{n u}^0$ par l'ensemble des commandes présentes en préparation. Deux cas peuvent apparaître :

- Soit lorsque $V_u^0 + D_{n u}^0 < K_u$ impliquant que le carousel tourne du plateau V_u^0 au niveau $V_u^0 + D_{n u}^0$ sans avoir dépassé le niveau K_u . A $t=0$: $NP_{n u}^0 = \sum_{k=V_u^0}^{V_u^0 + D_{n u}^0} N_{k u}^0 \quad \forall u \in v ; \forall n$ tel que $Y_n^0 = 1$
- Soit lorsque $V_u^0 + D_{n u}^0 > K_u$ impliquant que le carousel visite du plateau V_u^0 au plateau K_u puis du plateau 1 au plateau $V_u^0 + D_{n u}^0 - K_u$. A $t=0$: $NP_{n u}^0 = \sum_{k=V_u^0}^{K_u} N_{k u}^0 + \sum_{k=1}^{V_u^0 + D_{n u}^0 - K_u} N_{k u}^0 \quad \forall u \in v ; \forall n$ tel que $Y_n^0 = 1$

Nous avons obtenu le nombre de plateaux que chacun des carousels doit visiter à partir de leur emplacement actuel, pour chacune des commandes sur la table de préparation, afin d'être préparée.

	$NP_{n u}^0$		
Commande	1	2	3
Carousel 1	4	3	2
Carousel 2	3	2	4
Carousel 3	4	1	3

Nous en déduisons la commande n finissant en premier tel que $n = \underset{n}{\operatorname{argmin}} \left(\max_{u \in \{1 \dots U\}} (NP_{n u}^0) \right)$.

Soit P le vecteur nous indiquant les numéros des commandes dans l'ordre auquel sont collectées ses commandes. $P(1) = \underset{n}{\operatorname{argmin}} \left(\max_{u \in \{1 \dots U\}} (NP_{n u}^0) \right) = 2$. En d'autres mots, ce sera la commande 2 dont le nombre maximal de plateaux à visiter (3 plateaux) est le plus faible. Le carousel finissant

la préparation sera le carousel u tel que $\operatorname{argmin}_n \left(\max_{u \in \{1 \dots U\}} (NP_{n u}^t) \right) = 2$ dans notre exemple, soit le carousel 1.

Dans le cas particulier où deux commandes ont un même nombre max de plateau à visiter, nous prenons alors en compte l'alternance de visite des carousels en compte afin de déterminer lequel des carousels sera fini avant l'autre, cela donnant la commande finie en premier. Ainsi pour ces deux commandes, nous calculons une distance entre les carousels (la distance entre le carousel sur lequel a débuté la collecte et le carousel finissant la collecte de la commande). Cette distance est, pour chaque stockeur u ayant un nombre max de plateaux à voir égale à $(u - U^t) \operatorname{Mod}(U)$. Nous définirons la commande finissant par celle ayant la plus faible distance.

○ **Calcul des distances à parcourir entre les lignes à collecter**

Une fois la commande terminant en premier déterminée, nous regardons les niveaux visités afin de la collecter. Ainsi pour chacun des plateaux collectés, nous observons la distance que le carousel a dû parcourir.

Soit $PV_{k u}^t$ égal à 1 si le plateau k est visité sur le carousel u pour terminer la t – ième commande. Pour ce paramètre, nous utilisons le fait que nous visitons les plateaux en alternant les stockeurs. Si le carousel 1 doit visiter trois plateaux, alors en commençant au stockeur1, nous en visiterons deux sur le carousel 2 et deux sur le carousel 3. Cependant si nous avons commencé sur le carousel 1 et finit sur le carousel 2 qui devait visiter 3 plateaux, alors nous aurions visité 3 plateaux du carousel 1 et du carousel 2, mais seulement deux plateaux du carousel 3.

	$PV_{k u}^0$				
Carousel 1	1	1	1	0	0
Carousel 2	1	1	0	0	0
Carousel 3	0	1	1	0	0

Nous énumérons maintenant l'ordre des plateaux visités. Pour ce faire nous énumérons les plateaux tel que $PV_{k u}^0 = 1$. Et nous les ordonnons par ordre croissant des numéros de plateaux visités (i.e. de telle sorte que pour les k tels que $(k - V_u^t) \operatorname{Mod}(K_c)$ soient croissant). Ici, le premier plateau visité par le stockeur 1 sera le plateau 1 puis le second plateau est le plateau 2 puis le troisième est le plateau 3. Pour le stockeur 2, le premier plateau est le plateau 1 puis le second plateau est le plateau 2. Enfin pour le troisième stockeur, le premier plateau visité est le 2, et le second plateau est le plateau 3.

A partir de ce tableau nous en tirons les distances à parcourir entre les plateaux à visiter, soit $NN_{l_u}^t$, le nombre de plateaux que le carousel doit visiter pour atteindre le l – ième plateau visité du carousel u entre l’instant t et l’instant $t+1$. Ici pour le premier carousel, le premier plateau à visiter est le plateau 1, sur lequel est positionné le carousel à $t=0$, donc pour atteindre ce plateau, $NN_{1_1}^0 = 0$. Pour le second plateau, une rotation d’un niveau est à faire : $NN_{2_1}^0 = 1$, de même pour le troisième plateau et les plateaux à visiter pour les autres carousels.

	$NN_{l_u}^0$		
l-ième plateaux visité	1	2	3
Carousel 1	0	1	1
Carousel 2	0	1	0
Carousel 3	1	1	0

Nous regardons maintenant le nombre de lignes collectées sur chacun des plateaux. Soit $L_{l_u}^t$ le tableau recensant les lignes que doivent visiter les commandes en préparation sur les plateaux visités entre l’instant t et $t+1$. Ainsi nous sommes toutes les lignes $L_{n_k u}$ devant être vues par chacune des commandes en préparation (commandes tel que $Y_n^t = 1$) pour chacun des plateaux visités par les commandes en préparation.

	$L_{l_u}^0$		
l-ième plateaux visité	1	2	3
Carousel 1	2	2	1
Carousel 2	1	2	0
Carousel 3	1	2	0

Cas particulier : Si la collecte commence sur un carousel autre que le carousel 1, nous incluons en première colonne un 0 pour les carousels ayant un numéro plus faible que le numéro du carousel finissant, ceci afin de calculer le temps d’attente comme dans le chapitre 3.

○ **Calcul du temps d’attente pour finir cette commande**

A partir de ces deux tableaux nous calculons le temps d’attente comme dans le chapitre 3. Ici nous regarderons pour chacun des plateaux visités par chacun des carousels si le temps de rotation ($NN_{l_u}^t$) est plus grand que les temps de picking et d’attente.

$$W^t = \min_{u \in \{1 \dots U\}} \max_{(NP_{n_u}^0)} \sum_{l=1}^u \sum_{u=1}^u W_{l_u}^t$$

$$W_{l_u}^t = \max \left(r * NN_{l_u}^t - \left(\sum_{v=1}^{u-1} (W_{l_v}^t + L_{l_v}^t * p) \right) + \sum_{v=u+1}^c (W_{l_{-1}v}^t + L_{l_{-1}v}^t * p); 0 \right)$$

Pour notre exemple nous aurons un temps d'attente de $W^0 = 0$.

- **Enlever cette commande préparée de la table de préparation**

Le paramètre Y_n^t devra être mis à 0 pour la commande venant d'être terminée. Ainsi $Y_{P(1)}^1 = Y_2^1 = 0$

- Actualisation des paramètres

Une fois le calcul de la première commande terminée, nous devons réactualiser les paramètres afin de retrouver les données nécessaires pour calculer la seconde commande terminée et le temps d'attente associé.

- Emplacement actuel de chacun des carrousels

Le paramètre V_u^t désigne l'emplacement du carrousel u sur lequel s'est terminée la commande à l'évènement t . Nous rappelons que pour terminer une commande, il faut donc collecter sur un nombre de niveaux égal à $\min_n \max_{u \in \{1 \dots U\}} (NP_{nu}^0)$ sur chacun des carrousels. Pour le calcul de V_u^{t+1} , nous utilisons NN_{lu}^t .

$$V_u^{t+1} = (V_u^t + \sum_l NN_{lu}^t) \text{Mod}(K_u)$$

Dans notre exemple $V_1^1 = V_1^0 + NN_{11}^0 + NN_{21}^0 + NN_{31}^0 = 1 + 2 = 3$;

De même $V_2^1 = 1 + 1 = 2$ et $V_3^1 = 1 + 2 = 3$.

- Niveaux restants à voir par chacune des commandes au cours de la préparation

Le paramètre E_{nk}^1 doit être calculé pour les commandes étant présentes sur la table de préparation à l'évènement 0. En effet celles-ci ont visité les niveaux allant de V_u^0 à V_u^1 .

- Niveaux restants à voir par chacune des commandes au cours de la préparation

Donc pour les commandes qui étaient présentes à $t=0$, leurs niveaux à voir sont réactualisés. Donc $\forall n$ tel que $Y_n^0 = 1$

Si $V_u^0 < V_u^1$ alors $E_{nk}^1 = 0 \forall k \in \{V_u^0 \dots V_u^1\}$ et $E_{nk}^1 = E_{nk}^0 \text{ k } \notin \{V_u^0 \dots V_u^1\}$

Si $V_u^0 > V_u^1$ alors $E_{nk}^1 = 0 \forall k \in \{V_u^0 \dots K_u\} \cup \{1 \dots V_u^1\}$ et $E_{nk}^1 = E_{nk}^0 \text{ k } \notin \{V_u^0 \dots K_u\} \cup \{1 \dots V_u^1\}$.

Il reste à voir les niveaux suivant pour chacune de nos commandes. Sachant que la commande 2 est collectée, il ne reste plus de niveaux à visiter.

	Commande 1					Commande 2					Commande 3					Commande 4				
Carousel 1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
Carousel 2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	1
Carousel 3	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0

- Carousel terminant la collecte

Ce carousel sera celui où sur lequel le dernier plateau a été visité. Donc le carousel u étant égal à $\underset{u}{\operatorname{argmax}}(NP_{P(1)u}^0)$, ici c'est le carousel 1.

Cas particulier : si une commande finissant a le même nombre max à visiter sur plusieurs carousels, nous réutilisons la distance entre les carousels pour déterminer celui qui terminera la collecte.

- Si il reste des commandes « en attente »

- Ajout d'une commande en fonction de l'ordre d'ajout

Nous avons simplement à rajouter la commande suivante de l'ordre d'ajout en changeant son statut de « en attente » à « en préparation », $Y_{A(B+t)}^t = 1$. Donc $Y_4^1 = 1$.

Nous avons maintenant actualisé l'ensemble des paramètres permettant le calcul des variables pour le calcul de $P(2)$, Nous pouvons maintenant calculer pour n'importe quel autre t en suivant le même raisonnement itératif. Nous avons donc les commandes 1, 3 et 4 sur la table de préparation, nous réitérons les étapes précédentes pour finir la collecte :

Les distances à parcourir par les trois commandes en préparation à partir de l'emplacement actuel des carousels :

	D_{nc}^1		
Commande	1	3	4
Carousel 1	1	0	4
Carousel 2	1	3	3
Carousel 3	2	1	3

Les plateaux à visiter pour l'ensemble des commandes sont :

	N_{ku}^1				
Carousel 1	0	1	0	1	1
Carousel 2	0	1	1	0	1
Carousel 3	1	0	1	1	1

Le nombre de plateaux à visiter pour chaque commande sur chacun des carousels est :

	NP_{nu}^1		
Commande	1	3	4
Carousel 1	2	0	3
Carousel 2	1	3	3
Carousel 3	3	1	4

Nous en déduisons que la commande finissante est la commande 3, et que le carousel finissant est le carousel 2. Nous en déduisons les niveaux à visiter pour collecter cette commande.

	PV_{ku}^1				
Carousel 1	0	1	0	1	1
Carousel 2	0	1	1	0	1
Carousel 3	0	0	1	1	0

Nous voyons ici l'importance du carousel commençant la collecte et de l'alternance de visite des carousels. Sachant que nous commençons la collecte sur le carousel 1 et que la collecte se finit sur le carousel 2, le carousel 1 et le carousel 2 visiteront 3 plateaux tandis que le carousel 3 n'en visitera que 2. Nous en déduisons les distances que les carousels doivent parcourir entre les différents plateaux sur chaque carousel ainsi que le nombre de lignes à collecter sur chacun des plateaux à visiter.

	NN_{lu}^1		
l-ième plateaux visité	1	2	3
Carousel 1	1	1	2
Carousel 2	0	1	2
Carousel 3	0	1	0

	L_{lu}^t		
l-ième plateaux visité	1	2	3
Carousel 1	1	1	1
Carousel 2	1	1	2
Carousel 3	1	2	0

Enfin nous calculons le temps d'attente présent lors de la collecte de cette commande $W^1 = 1$.

La commande numéro 3 est terminée, ainsi $Y_3^1 = 0$ et il ne reste plus que les commandes 1 et 4 à collecter : $Y_1^1 = Y_4^1 = 1$. Les emplacements des carousels, dans notre exemple sont les suivants : $V_1^2 = (3 + 4)(Mod 5) = 2$; $V_2^2 = 2 + 3 = 5$ et $V_3^2 = 3 + 1 = 4$.

Les niveaux restant à visiter pour chacune des commandes sont :

	Commande 1					Commande 2					Commande 3					Commande 4				
Carousel 1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Carousel 2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Carousel 3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Nous observons que la collecte ne s'effectue maintenant que sur un seul carrousel. Les distances à parcourir par les trois commandes en préparation à partir de l'emplacement actuel des carrousels :

D_{nc}^2		
Commande	1	4
Carrousel 1	0	0
Carrousel 2	0	0
Carrousel 3	1	2

Les plateaux à visiter pour l'ensemble des commandes sont :

	N_{ku}^2				
Carrousel 1	0	0	0	0	0
Carrousel 2	0	0	0	0	0
Carrousel 3	1	0	0	0	1

Le nombre de plateaux à visiter pour chaque commande sur chacun des carrousels est :

	NP_{nu}^2	
Commande	1	4
Carrousel 1	0	0
Carrousel 2	0	0
Carrousel 3	1	2

Nous en déduisons que la commande finissante est la commande 1, et que le carrousel finissant est le carrousel 3. Nous en déduisons les niveaux à visiter pour collecter cette commande :

	PV_{ku}^2				
Carrousel 1	0	0	0	0	0
Carrousel 2	0	0	0	0	0
Carrousel 3	0	0	0	0	1

Nous en déduisons les distances que les carrousels doivent parcourir entre les différents plateaux à visiter ainsi que le nombre de lignes à collecter sur chacun des plateaux à visiter.

	NN_{lu}^2
I	1
Carrousel 1	0
Carrousel 2	0
Carrousel 3	1

	L_{lu}^2
I	1
Carrousel 1	0
Carrousel 2	0
Carrousel 3	1

Enfin nous calculons le temps d'attente $W^2 = 1$.

De la même façon nous obtenons $W^3 = 1$

Calcul temps d'attente pour la préparation de l'ensemble des commandes

Nous sommes les divers temps d'attente : $W = \sum_t W^t = 3$.

Ainsi donc en sortie de l'algorithme, nous avons pour l'ordre d'ajout $\{1,2,3,4\}$ le temps d'attente associé :

3. Pour information, un ordre d'ajout permettant d'obtenir un temps d'attente optimal égal à 2 est $\{2,3,4,1\}$.