

République Algérienne Démocratique Et Populaire
Ministère De L'enseignement Supérieur Et De La Recherche Scientifique
Université Abou Bakr Belkaid-Tlemcen
Faculté De Technologie
Département De Génie Electrique Et Electronique

Thèse De Doctorat En Génie Industriel

Intitulée :

**Optimisation du temps moyen de déstockage dans un AS/RS
profond en se basant sur des techniques de l'Intelligence Artificielle**

Présentée par :

AMARA Zakarya

Président :

BETAOUAF Hichem M.C.A Université De Tlemcen–Algérie

Examineur :

KARA Redouane Professeur Université De Tizi Ouzou–Algérie

**BRAHAMI Moustapha
ANWAR** M.C.A Ecole supérieur de sciences
Appliquées Tlemcen –Algérie

Directrice de thèse :

KOULOUGHLI Sihem Professeur Université De Tlemcen–Algérie

Invité(e) :

GHOMRI Latefa Professeur Université De Tlemcen–Algérie

2023/2024

Table des matières

Table des matières.....	2
Liste des figures	5
Liste des tableaux.....	7
Liste des abréviations.....	8
Remerciements.....	10
Introduction Générale	11
Chapitre 1: Les Systèmes Automatisés de Stockage/Déstockage à profondeur multiple.....	14
1.1. Introduction	15
1.2. Définition, Composition, Avantages et Inconvénients	15
1.2.1. Définition	15
1.2.2. Composition	15
1.2.3. Terminologie	16
1.2.4. Avantages	19
1.2.5. Inconvénients	19
1.3. Types d'AS/RS.....	19
1.3.1. Les AS/RS à charge unitaire	20
1.3.2. Les AS/RS multi allées	20
1.3.3. Les AS/RS à racks glissants.....	21
1.3.4. Les AS/RS à mini charge	22
1.3.5. Les AS/RS à carrousel.....	22
1.3.6. Les AS/RS à personne embarquée	23
1.3.7. Les AS/RS à étagères profondes	23
1.3.8. Les AS/RS à convoyeur gravitationnel	24
1.3.9. Les AS/RS à convoyeur gravitationnel Bidirectionnel	25
1.3.10. Les AS/RS à convoyeur gravitationnel à une seule machine	26
1.4. Méthodologies de Stockage	27
1.4.1. Stockage dédié.....	27
1.4.2. Stockage aléatoire	28
1.5. Etat de l'Art.....	28

Table des matières

1.5.1.	Les AS/RS à une seule profondeur	30
1.5.2.	Les AS/RS Multi profondeur	32
1.5.2.1.	Temps de cycle	32
1.5.2.2.	Optimisation des dimensions des AS/RS compacts	35
1.5.2.3.	La position du point repos de la machine S/R et la station d'entrée/sortie (E/S) : 37	
1.5.2.4.	L'ordonnancement des transactions de déstockage pour les AS/RS :.....	38
1.5.2.5.	Les politiques de stockage :.....	39
1.6.	Conclusion.....	40
Chapitre 2 : Minimisation du temps de déstockage en utilisant l'apprentissage automatique		41
2.1.	Introduction	42
2.2.	Machine Learning	42
2.2.1.	Apprentissage supervisé.....	43
2.2.1.1.	La classification.....	44
2.2.1.2.	La régression.....	48
2.2.2.	Apprentissage non supervisé.....	50
2.2.3.	Apprentissage par renforcement.....	51
2.3.	AS/RS à convoyeur gravitationnel	52
2.3.1.	Stockage	53
2.3.2.	Déstockage	54
2.4.	Mise en œuvre de l'apprentissage machine	54
2.4.1.	Construction de la base de données.....	56
2.4.1.1.	Python Jupyter :	58
2.4.1.2.	Calcul du temps de déstockage (label)	58
2.4.2.	La phase d'apprentissage	58
2.4.2.1.	L'application des méthodes de classification.....	58
2.4.2.2.	L'application des méthodes de régression	60
2.4.3.	La phase de test	62
2.4.3.1.	L'arbre de décision	63
2.4.3.2.	La régression linéaire multi variable	63
2.4.3.3.	La régression polynomiale multi variable	64
2.4.4.	La méthode de stockage	64
2.5.	Resultats	64

Table des matières

2.6. Conclusion.....	66
Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage).....	67
3.1. Introduction	68
3.2. Les Métaheuristiques:	68
3.2.1. Recuit simulé.....	68
3.2.1.1. Avantages	69
3.2.1.2. Inconvénient	69
3.2.2. Recherche taboue	70
3.2.2.1. Avantages	70
3.2.2.2. Inconvénients.....	71
3.3. La méthode NN	71
3.4. Description du problème	71
3.5. Modélisation mathématique et résolution du problème d'optimisation étudié.....	73
3.5.1. Modélisation mathématique	73
3.6. Proposition d'une heuristique de résolution	75
3.7. Résultats	80
3.7.1. Temps moyen de déstockage/ Taille de lot = 2	81
3.7.2. Temps moyen de déstockage/ Taille de lot = 3	82
3.7.3. Temps moyen de déstockage/ Taille de lot = 5	85
3.7.4. Temps moyen de déstockage/ Taille de lot = 8	87
3.7.5. Temps moyen de déstockage/ Taille de lot = 10	89
3.7.6. Temps moyen de déstockage/ Taille de lot = 15	91
3.7.7. Résumé	93
3.8. Conclusion.....	94
Conclusion Générale	95
References	97
Resumé.....	104
Abstact	104
ملخص	105

Liste des figures

Figure 1. 1 : Casier de stockage	16
Figure 1. 2: Colonnes de casier.....	16
Figure 1. 3: Rack de stockage	17
Figure 1. 4: Une allée entre deux racks de stockage	17
Figure 1. 5: Palettes de stockage.....	18
Figure 1. 6: La machine de stockage/déstockage.....	18
Figure 1. 7: AS/RS à charge unitaire	20
Figure 1. 8: AS/RS multi allées	21
Figure 1. 9: AS/RS à racks glissants	21
Figure 1. 10: AS/RS à mini charge	22
Figure 1. 11: AS/RS à carrousel	23
Figure 1. 12: AS/RS à personne embarquée	23
Figure 1. 13: AS/RS à étagères profondes	24
Figure 1. 14: AS/RS à convoyeur gravitationnel	25
Figure 1. 15: AS/RS à convoyeur gravitationnel Bidirectionnel	26
Figure 1. 16: AS/RS à convoyeur gravitationnel à une seule machine.....	27
Figure 1. 17: Stockage par classe.....	28
Figure 2. 1: L'apprentissage automatique	43
Figure 2. 2: L'apprentissage non supervisé	44
Figure 2. 3: Exemple d'arbre de décision	45
Figure 2. 4: La régression logistique.....	46
Figure 2. 5: K-nearest neighbors.....	47
Figure 2. 6: Support Vector Machine (SVM)	47
Figure 2. 7 : Régression linéaire simple.....	49
Figure 2. 8: La régression non-linéaire	50
Figure 2. 9: L'apprentissage non supervisé	51
Figure 2. 10 : Apprentissage par renforcement.....	51
Figure 2. 11: AS/RS à convoyeur gravitationnel	52
Figure 2. 12 : La face de stockage dans un AS/RS à convoyeur gravitationnel	53
Figure 2. 13: La face de stockage dans un AS/RS à convoyeur gravitationnel	54
Figure 2. 14: la méthode de résolution.....	56
Figure 2. 15 : Les coefficients de la régression linéaire multi variable	63
Figure 2. 16 : Les coefficients de la régression polynomiale multi variable	64
Figure 2. 17 : le temps de déstockage pour chaque méthode.....	66
Figure 3. 1: La solution obtenue par le solveur gurobi	74
Figure 3. 2 : L'Organigramme de l'heuristique proposée.....	76
Figure 3. 3: AS/RS contient 4 casiers	77
Figure 3. 4 : Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 2	82
Figure 3. 5 : Le temps de déstockage pour la taille de lot = 2	82

Liste des figures

Figure 3. 6 : Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 3	84
Figure 3. 7 : Le temps de déstockage pour la taille de lot = 3	84
Figure 3. 8 : Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 5	86
Figure 3. 9 : Le temps de déstockage pour la taille de lot = 5	86
Figure 3. 10 : Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 8.....	88
Figure 3. 11 : Le temps de déstockage pour la taille de lot = 8	88
Figure 3. 12 Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 10	90
Figure 3. 13 : Le temps de déstockage pour la taille de lot = 10	90
Figure 3. 14: Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 15.....	92
Figure 3. 15 : Le temps de déstockage pour la taille de lot = 15	92
Figure 3. 16 : le temps de calcul des méthodes pour chaque taille du lot.....	93

Liste des tableaux

Tableau 1. 1 : Les décisions à prendre pour chaque classe de recherche	29
Tableau 2. 1: résumé des algorithmes de la classification	48
Tableau 2. 2 : Echantion d'un configuration du rack.....	57
Tableau 2. 3: Echantion de La matrice des entrées pour la classification	59
Tableau 2. 4: Echantion de la matrice des entrées pour la Régression linéaire multi-variable	60
Tableau 2. 5: Echantion La matrice des entrées pour La Régression polynomiale multi-variable	62
Tableau 2. 6 Le temps de déstockage moyen d'un produit pour chaque méthode	65
Tableau 3. 1: le temps moyen de déstockage d'un produit pour une taille de lot égale à 2.	81
Tableau 3. 2 le temps moyen de déstockage d'un produit pour une taille de lot égale à 3.....	83
Tableau 3. 3 le temps moyen de déstockage d'un produit pour une taille de lot égale à 5.....	85
Tableau 3. 4 : le temps moyen de déstockage d'un produit pour une taille de lot égale à 8.	87
Tableau 3. 5 le temps moyen de déstockage d'un produit pour une taille de lot égale à 10.....	89
Tableau 3. 6 : le temps moyen de déstockage d'un produit pour une taille de lot égale à 15.	91
Tableau 3. 7: le temps de calcul pour un produit en fonction de la taille du lot pour chaque	93

Liste des abréviations

bg_t : un meilleur groupe de produits	79
MTG_(min) : est une matrice qui contiennent les temps de déstockage des groupes utilisant les temps des produits les plus proches $\llbracket P \rrbracket_r$	75
t'_h : Le temps de déplacement horizontal d'un casier au suivant	53
t^_v : Le temps de déplacement vertical d'un casier au suivant	53
ACO : La colonie de fourmis.....	68
AI : L'intelligence artificielle	12
AS/RS : Les systèmes automatisés de stockage et de déstockage	12
dataset : les données d'entrées.....	59
g_s : un groupe de produits trouvés dans le lot et dans le même casier.....	77
GA : Algorithme génétique.....	68
Lot : est un vecteur de dimension q où chaque élément représente un type de produit à déstocker	75
M : le nombre de couches dans le casier.....	56
MG : Est la matrice de séquence	75
MR : est une matrice de dimension $N_h \times N_h \times M$. L'élément $MR(i,j,k)$ est égal au type de la charge stockée dans la position (i, j, k) et est égal à zéro si la position (i, j, k) est vide.....	75
MTG : est une matrice qui contiennent les temps de déstockage des groupes.....	75
N_h : Le nombre de casiers verticaux	53
N_l : Le nombre de casiers horizontaux.....	53
P : est un vecteur de dimension q qui contient la position du type demandé qui a le temps de déstockage minimal	75
Q : la quantité globale demandés pour tout les types.....	73
q_l : Est un vecteur qui contient les quantités demandés pour chaque type	73
R : le stockage aléatoire	64
S/R : storage/retrieval	13
SA : Le recuit simulé	13
Smin : le stockage dans l'emplacement minimal	64
T : est un vecteur qui contient le temps de déstockage minimal pour chaque élément dans le lot	79
V_v : La vitesse de déplacement vertical	53
APU : La carte graphique (GPU) est intégrée dans le CPU	43
BFR : L'AS/RS à convoyeurs gravitationnels bidirectionnels	32
CPU : unité centrale de traitement.....	43
DOS : Durée de séjour	38
E/S : La station d'entrée/ sortie	37
FF- AS/RS : free-FALL à convoyeur gravitationnel	34

Liste des abréviations

FIFO : First In First Out.....	31
GMS : heuristique (le groupement, la selection et le séquençement).....	38
GPU : Unité de traitement graphique.....	43
K-NN : k nearest neighbor.....	46
L : L'hauteur du rack	52, 53
ML : Machine learning	43
NN : la méthode de voisin le plus proche	31
PPR-SL : heuristique appelée la priorité en pourcentage aux demandes de stockage avec la distance la plus courte	38
SBS/RS : Système AS/RS par navette	35
SFR : simple AS/RS à convoyeur gravitationnel.....	38
SVM : Support vector machine.....	47
t_h : le temps de déplacement horizontal de la station entrée/sotrie à la colonne la plus lointaine	52
t_v : le temps de déplacement vertical de la station entrée/sotrie à la colonne la plus lointaine ..	53
TS : La recherche tabou	13
V_h : la vitesse de déplacement horizontal	52

Remerciements

Je rends grâce à Dieu l'Omniscient qui m'a prêté vie et concédé une infime partie de son incommensurable savoir.

Je ne remerciais jamais assez mes parents pour les sacrifices qu'ils m'ont consentis, m'armant pour la vie, me donnant l'éducation morale adéquate et m'inculquant les meilleurs enseignements à même de l'affronter, bien aguerri, pour parer les aléas.

Tous les mérites reviennent, de toute évidence, à mes illustres professeurs, que je hisse sur les plus hauts piédestaux et à qui je témoigne mon profond respect, mon abnégation et ma très haute considération.

Je me dois, en l'occurrence, d'exprimer ma profonde gratitude à Mesdames KOULOUGHLI Siham et GHOMRI Latefa, Professeurs de l'Université de Tlemcen, mes directrices de thèse, dont la rigueur de travail et les conseils avisés m'ont été très utiles dans ma recherche. Ses entières disponibilités, ses soutiens, ses patiences et ses lectures minutieuses de mes analyses ont été indispensables à bonne finalisation de mon doctorat.

Je tiens également à remercier le Président de jury Monsieur BETOUAF Hichem, Maître de conférences à l'université de Tlemcen, et messieurs les membres du jury KARA Redouane, Professeur à l'université De Tizi Ouzou–Algérie et BRAHAMI Mustapha, Maîtres de conférences à l'école supérieure de science appliquées Tlemcen, d'avoir eu l'amabilité de consentir à la soutenance de ma présente thèse.

Toute ma gratitude à messieurs RIMOUCHE Ali, BOUSSEROUAL Mourad et KHAMRI Bilal, ainsi que tous les membres de ma famille, Pour ses précieuses prestations informatiques, mes amis qui ont toujours su me témoigner leur sympathie et m'encourager, par leur assistance, dans le cadre de mes recherches d'études.

Dédicace

Je dédie ce travail, en tout premier lieu, à ma regrettée ma mère, sous les pieds de qui, se trouve mon paradis et à feu mon père qui n'a eu de cesse de m'apprendre les innombrables vertus du savoir: à la Résurrection, ils ce privilège de porter des couronnes lumineuses.

A mes illustres instituteurs et professeurs, j'exprime toute ma gratitude et leur tir toutes mes révérences.

"L'enseignant a failli être prophète." dit un précepte religieux qui en sublime ce rôle, on ne peut plus, sacré.

Introduction Générale

Introduction Générale

Les systèmes automatisés de stockage et de déstockage (AS/RS) sont devenus un élément clé de l'industrie et de l'entrepôt modernes, en raison de leur capacité à améliorer l'efficacité, la précision et la productivité des opérations de stockage et de déstockage. Les AS/RS comportent deux éléments de base, à savoir : une composante physique comportant des racks, des machines de stockage/déstockage, des stations de dépôts/livraison, ... La seconde composante est le système de contrôle, dont la fonction est la commande de la première composante.

Les AS/RS peuvent avoir plusieurs configurations physiques. Qui peuvent être classées suivant la capacité de chaque casier. En effet, dans la forme de base des AS/RS, chaque casier est conçu pour contenir une seule charge. Ce qui rend l'accès à chaque charge, immédiat et le contrôle plutôt facile. L'autre famille de configurations est dite à profondeur multiple. Dans cette famille de configuration, chaque casier est conçu pour contenir plusieurs charges, l'une derrière l'autre. Un des paramètres de cette famille de configuration est la capacité d'un casier. Dans ce second cas, le système de contrôle doit être plus performant, car l'accès à une charge n'est pas toujours immédiat. Souvent le déstockage d'une charge donnée nécessite le retrait de toutes celles qui la précèdent dans le casier.

Une de mesures de performances les plus utilisées pour l'évaluation d'AS/RS est le temps de cycle. En fait, ce temps est défini comme le temps moyen nécessaire pour que la machine de stockage/déstockage accède à un casier soit pour y stocker une charge ou pour déstocker une charge depuis. Cette définition du temps de cycle s'applique parfaitement aux cas des AS/RS à profondeur unitaire. Toutefois, dans le cas d'AS/RS à profondeur multiple, le temps moyen de stockage est différent du temps moyen de déstockage et les valeurs des deux paramètres dépendent de la configuration physique de l'AS/RS.

Dans ce travail de thèse, nous nous intéressons aux AS/RS à profondeur multiple plus particulièrement, aux AS/RS à convoyeur gravitationnel. C'est une des formes les plus basiques des AS/RS profonds. C'est un AS/RS profond avec un seul rack et deux machines, une dédiée aux Opérations de stockage et l'autre aux opérations de déstockage. Chaque casier est muni d'un convoyeur gravitationnel, qui permet aux charges de glisser le long. Lors d'une opération de stockage, la machine de stockage, met la charge sur l'extrémité d'un casier, cette charge va glisser le long du casier, jusqu'à atteindre le premier emplacement vide du casier. Lors d'une opération de déstockage, la machine de déstockage, retire une à une toutes les charges précédentes celle désirée, jusqu'à arriver à la bonne pièce qu'elle va déstocker. Dans ce travail, nous nous proposons de minimiser le temps moyen de déstockage d'un lot de charges. Pour cela, nous avons adapté des techniques de l'intelligence artificielle.

L'optimisation des systèmes industriels reste un défi important pour les chercheurs en génie industriel. L'intelligence artificielle (IA) a été utilisée ces dernières années pour l'optimisation des performances de ces systèmes. C'est une famille d'approches qui consiste à utiliser les technologies pour améliorer les processus et augmenter l'efficacité et la productivité de l'entreprise. L'IA peut être utilisée pour automatiser les tâches répétitives, améliorer la prise de décisions, augmenter la qualité des produits et services, améliorer l'expérience client, et réduire les coûts. Il peut aider les entreprises à mieux comprendre les données de leurs clients, de leurs employés et de

leurs processus d'affaires pour améliorer leur performance globale. Il est important de noter que pour utiliser efficacement l'IA dans l'optimisation des performances, il est nécessaire d'avoir des données de qualité, une équipe compétente pour mettre en place et gérer les systèmes d'IA et une stratégie claire pour utiliser ces technologies.

Cette thèse se compose de trois chapitres, dans le premier chapitre nous présentons une définition formelle d'un AS/RS, les différents types existants, les principaux composants qui les constituent, ainsi que leurs avantages et inconvénients. Tout cela en utilisant des images et des schémas pour illustrer nos propos. Enfin, nous clôturerons ce chapitre par une analyse de la littérature existante sur les AS/RS.

Dans le deuxième chapitre, nous présentons une méthode d'apprentissage automatique que nous avons développé pour la minimisation du temps de déstockage dans un AS/RS profond. Le but est de stocker les charges à l'AS/RS à convoyeur gravitationnel de tel sorte que leur temps de déstockage soit optimal. En d'autres termes, nous rangeons les charges à l'intérieur des racks, afin de pouvoir prédire le meilleur emplacement pour chaque charge de manière à minimiser son temps de déstockage, le temps total (moyen) de déstockage est également minimisé. Pour cela, nous avons utilisé des ensembles de données de différentes dimensions pour obtenir un modèle machine learning généralisé, qui peut être utilisé pour les AS/RS de n'importe quelle dimension. Nous avons entraîné le modèle à classer les emplacements dans les AS/RS en fonction de leur état (vide ou plein) en utilisant la classification. Ensuite, nous l'avons entraîné à calculer le temps de déstockage requis pour chaque emplacement à l'aide de la régression polynomiale multi-variable, ce qui nous a donné des résultats très satisfaisants. À partir de ces résultats, nous déterminons la meilleure solution (temps d'accès minimal), ce qui nous donne un temps de déstockage moyen minimal. Enfin, nous avons comparé le modèle avec le stockage aléatoire et le stockage dans l'emplacement donnant un temps de déstockage minimal.

Dans le troisième chapitre nous étudions le cas des systèmes AS/RS à convoyeur gravitationnel. Cette configuration d'AS/RS utilise deux machines S/R, la première s'occupant des opérations de stockage et l'autre des opérations de déstockage. Dans ce type d'AS/RS, les opérations de déstockage sont assez complexes, puisque le déstockage d'une charge nécessite le restockage de toutes les charges qui la précèdent dans l'emplacement. Pour cette raison, il est préférable de considérer le retrait de lots de charges plutôt que de considérer le retrait de charges une par une. Le problème qui nous intéresse ici consiste à considérer une disposition de charges dans un rack profond et un lot de types de charges à déstocker (une commande) ; nous allons d'abord, localiser dans le rack les charges à déstocker (pour satisfaire la commande), puis définir la bonne séquence d'opérations de déstockage ; dans le but de minimiser le temps global de déstockage du lot. Le problème que nous nous proposons d'étudier ici est assez particulier et ne ressemble à aucun des problèmes classiques que l'on trouve dans le domaine du génie industriel. La première contribution de ce chapitre est la modélisation mathématique du problème décrit. La seconde contribution est une heuristique permettant de résoudre le modèle mathématique proposé, nous l'appelons heuristique de minimisation du temps de déstockage par lots (BRTM-heuristique). Nous adaptons également les deux métaheuristiques : le recuit simulé (SA) et la recherche tabou (TS) à la résolution du problème. Les résultats de l'heuristique BRTM seront comparés aux résultats donnés par les deux métaheuristiques afin de valider la précision de l'heuristique BRTM.

Chapitre 1: Les Systèmes Automatisés de Stockage/Déstockage à profondeur multiple

Chapitre 1

Les Systèmes Automatisés de Stockage/Déstockage à profondeur multiple

1.1. Introduction

Dans ce premier chapitre nous allons présenter le système physique sur laquelle l'étude sera menée. Il s'agit des systèmes automatisés de stockage et de déstockage à profondeur multiple. Dans ces systèmes de stockage, chaque casier est conçu pour le stockage de plusieurs charges. Ce chapitre va donc comporter une description détaillée des systèmes de stockage, d'abord à profondeur unique, ensuite à profondeur multiple. Ce chapitre comporte aussi un état de l'art détaillé des travaux relatifs à la problématique de cette thèse.

1.2. Définition, Composition, Avantages et Inconvénients

1.2.1. Définition

Les systèmes de stockage/déstockage sont largement utilisés dans les entreprises industrielles pour gérer les produits de telle sorte que le temps, l'espace et les équipements soient utilisés de manière rentable. Un AS/RS peut être défini comme une combinaison d'équipements et de systèmes de contrôle qui permettent de stocker et de déstocker automatiquement des charges (matières premières, produits finis et outils) dans l'entrepôt avec une haute vitesse et une grande précision. Le système AS/RS est un système intégré dans l'entrepôt et non pas un entrepôt complet.

L'AS/RS élimine l'intervention humaine dans l'exécution de l'ensemble des opérations de base telles que le stockage d'une charge dans un lieu présélectionné (selon de la politique de stockage) à partir d'un point d'entrée et le déstockage d'une charge depuis un lieu de stockage puis le transporte vers le point de sortie. Un AS/RS peut également effectuer d'autres opérations supplémentaires comme l'identification, la réception, la classification et l'emballage... des composants, des matières premières, des produits finis et des outils.

1.2.2. Composition

Les principaux composants d'un système automatisé de stockage et de déstockage, quel que soit son type, sont les suivants :

- Des machines de stockage/déstockage.
- Des stations d'entrée/sortie, également appelés stations de dépôt/livraison.
- Un ensemble de racks.
- Un certain nombre d'allées.
- Un système de contrôle.

1.2.3. Terminologie

- Casier : c'est un emplacement dans le rayonnage utilisé pour le stockage d'une charge ou plus (Figure 1.1). La structure de base de casier est à simple profondeur, il y a d'autres systèmes qui utilisent double, triple... profondeur, où les charges (les palettes) sont placées les unes devant les autres.

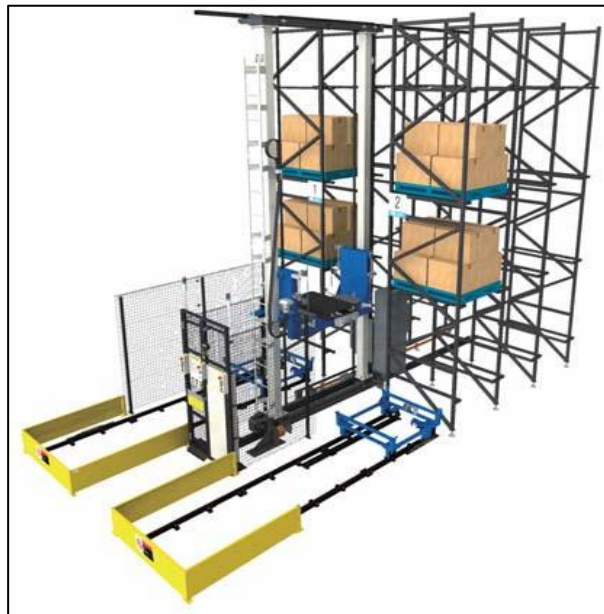


Figure 1. 1 : Casier de stockage

- Baie : la baie est une colonne constituée de plusieurs casiers placés les uns sur les autres (Figure 1.2).

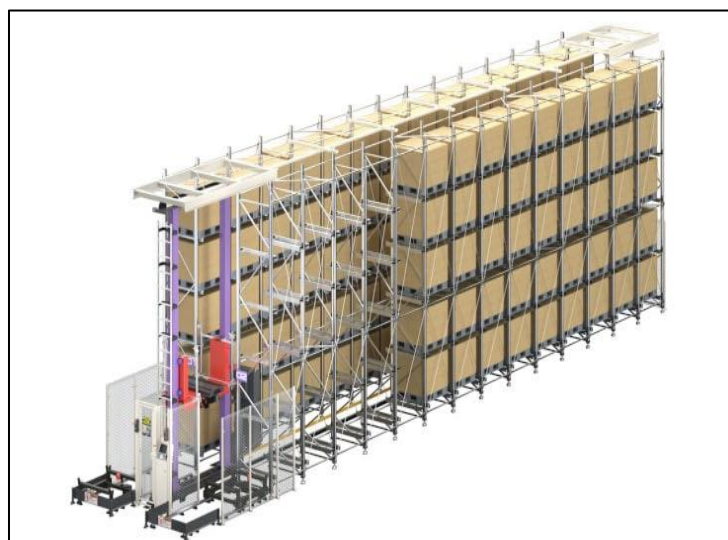


Figure 1. 2: Colonnes de casier

- Rack : c'est l'ensemble des baies positionnées les unes à la suite des autres (Figure 1.3).

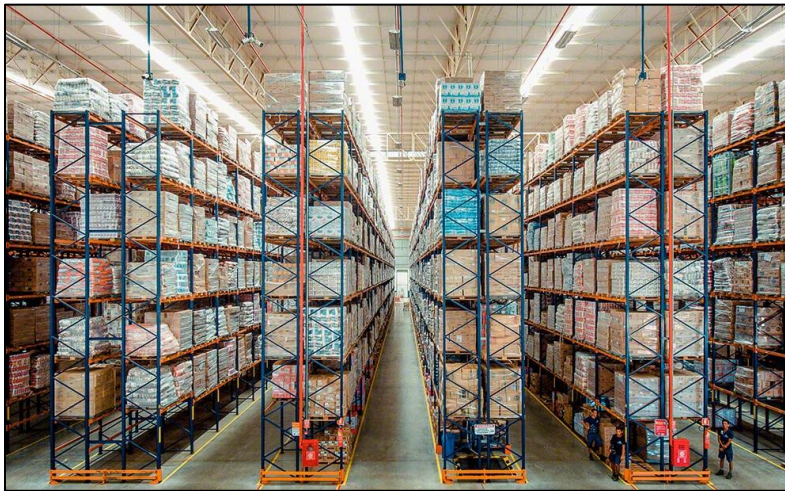


Figure 1. 3: Rack de stockage

- Allée : c'est l'espace entre deux rack qui permet aux machines de stockage/déstockage de se déplacer (Figure 1.4).



Figure 1. 4: Une allée entre deux rack de stockage

- Modules de stockage : ce sont des palettes ou des conteneurs en fil métallique, des bacs, des boîtes en carton ou en plastique et des tiroirs spéciaux utilisés pour contenir les articles en stock (Figure 1.5). Ils sont généralement fabriqués selon une taille et une forme de base standard, capables d'être déplacés par des machines S/R et stockés dans des casiers.

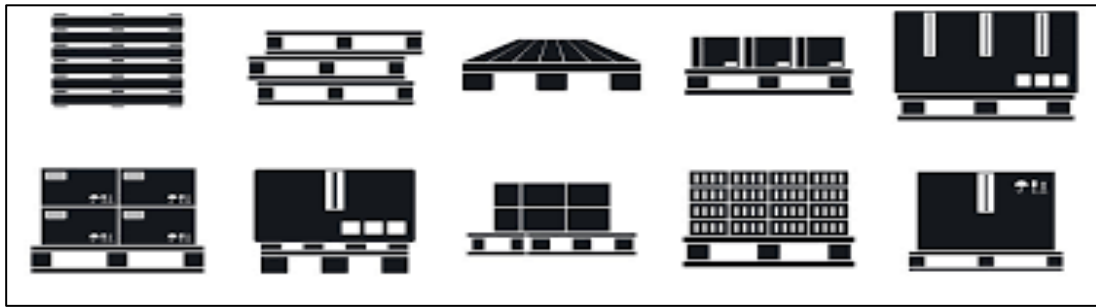


Figure 1. 5: Palettes de stockage

- Machine S/R : La machine de stockage/déstockage (S/R) est une machine entièrement automatisée utilisée pour déplacer des charges. La machine est capable de se déplacer horizontalement (du sol au plafond) et verticalement sur le long de l'allée (il existe plusieurs types de déplacement ; le plus utilisé est le Chebyshev), la machine est guidée par un système de rail (Figure 1.6). Il existe une variété de choix pour l'emplacement et le nombre de machines S/R suivant le type d'AS/RS.



Figure 1. 6: La machine de stockage/déstockage

- Les stations d'entrées/sorties : aussi appelées stations de dépôt/livraison, ce sont des endroits où les charges entrantes sont transportées et prêtes à être stockées ainsi que les charges sortantes qui sont transportées et prêtes à être déposées. La localisation et le nombre de stations P/D dépendent de la source des charges entrantes et de la destination des charges sortantes.

1.2.4. Avantages

Les AS/RS possèdent de nombreux avantages dans l'entrepôt, les plus importants sont les suivants :

- Utilisation efficace de l'espace.
- Amélioration de la gestion et du contrôle des stocks.
- Temps de réponse réduit.
- Capacité de stockage augmentée.
- Réduction du taux de rupture de stock.
- Réduction des ressources humaines.
- Augmentation du rendement, de la sécurité et de la flexibilité de fonctionnement.
- Diminution des coûts de fonctionnement.

1.2.5. Inconvénients

Les systèmes AS/RS présentent deux inconvénients majeurs liés à leur structure. Le premier est que lorsqu'un système AS/RS est installé, il est difficile de le modifier en raison de leur niveau de complexité et de leur rigidité. Le second est que le coût d'installation est très élevé.

1.3. Types d'AS/RS

Les AS/RS peuvent être classés en fonction de plusieurs paramètres tels que la taille, le poids ou les dimensions des produits, le nombre d'emplacement dans le casier, le nombre de machines S/R utilisées, la méthode de stockage et de déstockage, le nombre et les positions des points d'entrée/sortie, le nombre d'allées, les domaines d'applications... La liste suivante comprend quelques-uns des types de AS/RS les plus importants et les plus utilisés :

- Les AS/RS à charge unitaire.
- Les AS/RS multi allées.
- Les AS/RS à racks glissants.
- Les AS/RS à mini charge.
- Les AS/RS à carrousel.
- Les AS/RS à personne embarquée.
- Les AS/RS à étagères profondes.
- Les AS/RS à convoyeur gravitationnel.
- L'AS/RS à convoyeurs gravitationnels à une seule machine.
- L'AS/RS à convoyeurs gravitationnels bidirectionnels.

1.3.1. Les AS/RS à charge unitaire

Les AS/RS à charge unitaire (Figure 1.7) sont des systèmes de stockage/déstockage utilisés pour stocker et déstocker des charges importantes et standardisées, qui sont palettisées ou stockées dans des conteneurs, la charge stockée peut atteindre une masse de 250 kg et une hauteur de plus de 30 mètres. Le système est contrôlé par une supervision informatique. Il est composé de plusieurs allées et est équipé par des machines de stockage/déstockage appelées machines S/R, chacune de ces machines étant placée entre deux allées. La machine S/R est guidée par des rails fixés au sol, constituée d'un cadre perpendiculaire qui se déplace verticalement à l'aide des rails et ce cadre est



Figure 1. 7: AS/RS à charge unitaire

composé d'un plateau qui peut se déplacer horizontalement à l'aide du cadre. Ce mécanisme permet à la machine de stocker/déstocker les charges dans n'importe quel casier. Le système de charge unitaire est le système de base de l'AS/RS (Sari 2003).

1.3.2. Les AS/RS multi allées

Les AS/RS multi allées (Figure 1.8) sont des variations des systèmes de stockage/déstockage automatisés à charge unitaire. La seule différence est le nombre de machine S/R. dans l'AS/RS à charge unitaire il y a une machine S/R par allée, mais dans les systèmes AS/RS à multi-allées, la machine S/R peut se déplacer entre plusieurs allées à travers une allée qui s'enroule autour des allées (Sari 2003). Ce type est moins coûteux car il nécessite moins de machines S/R, mais son débit est inférieur à celui des systèmes à charges unitaires.

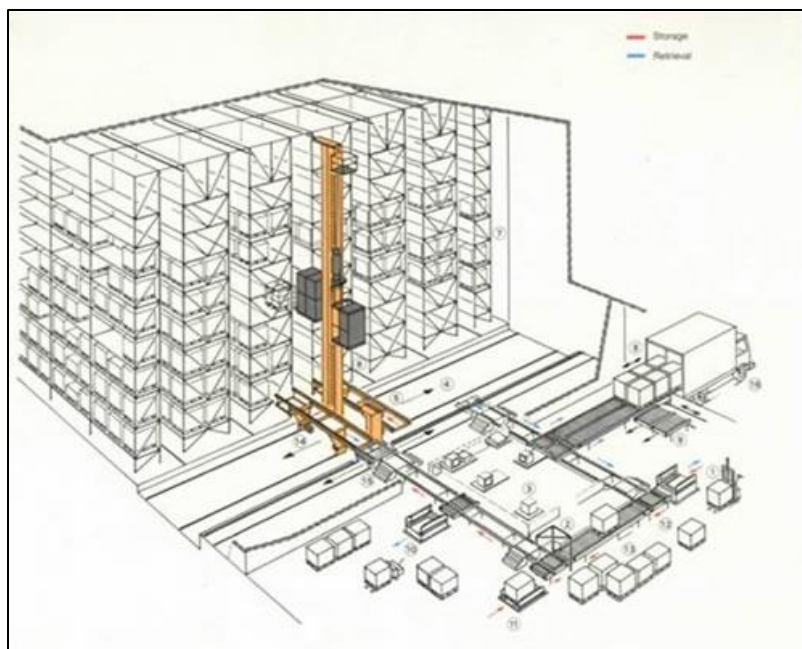


Figure 1. 8: AS/RS multi allées

1.3.3. Les AS/RS à racks glissants

Les AS/RS à racks glissants (Figure 1.9) sont aussi des variations des systèmes de stockage/déstockage automatisés à charge unitaire. Dans les AS/RS à racks glissants, les machines ne sont pas fixées à une allée spécifique. Les racks peuvent se déplacer pour ouvrir une allée entre deux racks adjacents, ce dernier est guidé par des rails. Cette technique permet à la machine S/R d'accéder à n'importe quel rack, de couvrir une plus grande zone de travail, et son débit est supérieur aux AS/RS multi allées (Sari 2003).

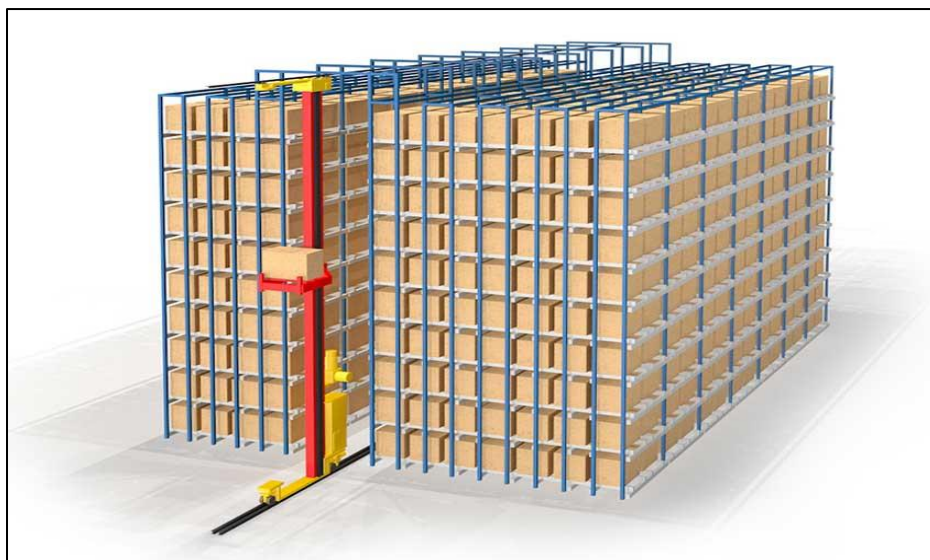


Figure 1. 9: AS/RS à racks glissants

1.3.4. Les AS/RS à mini charge

Les AS/RS à mini charge (Figure 1.10) manipulent généralement des charges plus petites que les systèmes à charge unitaire. Ils sont utilisés pour le stockage de petites charges telles que des outils individuels, des pièces contenues dans des boîtes, des plateaux ou des cartons. Vu que les charges sont beaucoup plus petites, le système utilise généralement une ventouse, des robots ou des tireurs. Il y a une machine S/R par allée. Le volume de stockage/déstockage est trop faible pour un système à charge unitaire à grande échelle et trop élevé pour un système manuel (Sari 2003). Les AS/RS à mini charge sont particulièrement conçus pour les opérations nécessitant des milliers de références (d'unités de stock) et un stockage important. Ils sont généralement utilisés pour assurer la sécurité des charges stockées et appliqués dans un espace limité.

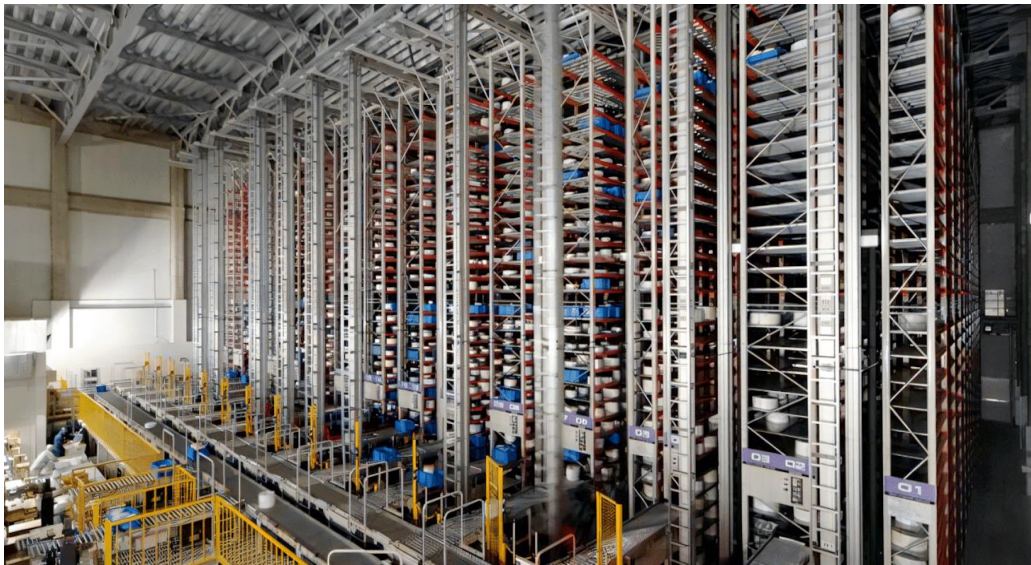


Figure 1. 10: AS/RS à mini charge

1.3.5. Les AS/RS à carrousel

Les AS/RS à carrousel (Figure 1.11) sont généralement utilisés pour les petites charges comme les AS/RS à mini charge, la différence principale est que les casiers du rack tournent automatiquement autour d'une allée jusqu'à l'extrémité du rack afin de stocker/déstocker les charges à l'aide d'un opérateur ou d'un robot. Les AS/RS à carrousel peuvent être équipés par un carrousel vertical où les casiers se déplacent verticalement, ou par un carrousel horizontal où les casiers se déplacent horizontalement.



Figure 1.11: AS/RS à carrousel

1.3.6. Les AS/RS à personne embarquée

Les AS/RS à personne embarquée (Figure 1.12) permettent de stocker un certain nombre de petites charges dans un casier. Dans ce système, un opérateur humain se déplace sur le chariot de la machine S/R pour stocker et déstocker les charges en même temps, ce qui augmente le débit du système. L'opérateur sélectionne les charges à déstocker et les emplacements des charges à stocker dans chaque casier. Les charges à déstocker sont placées sur la machine S/R à l'aide d'un module, qui les transporte jusqu'à la station de sortie.

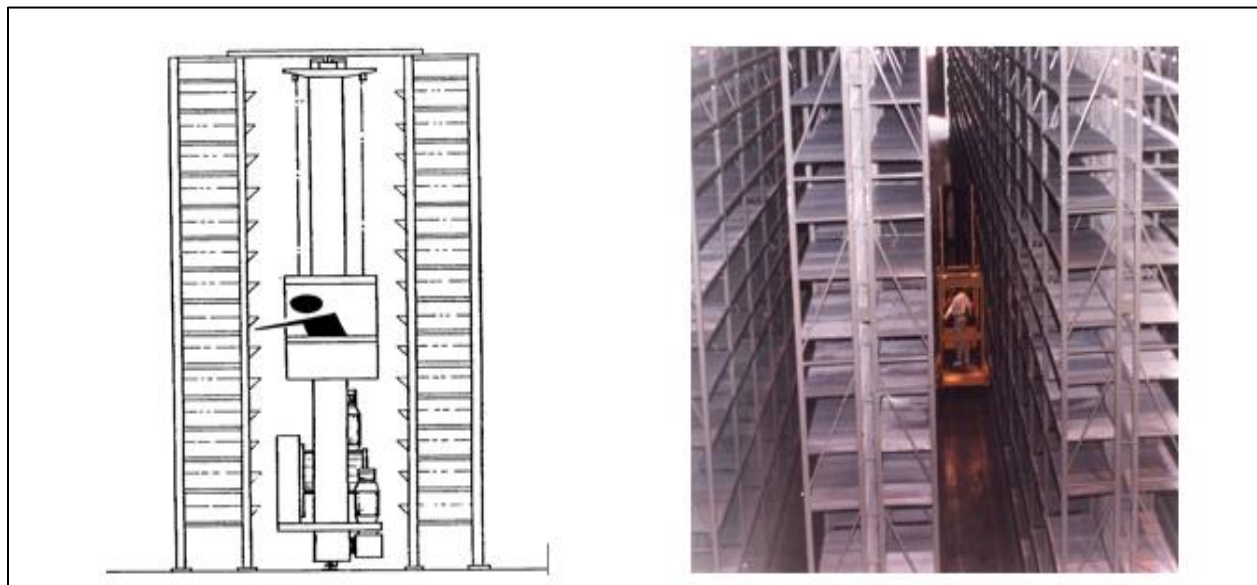


Figure 1.12: AS/RS à personne embarquée

1.3.7. Les AS/RS à étagères profondes

Ces systèmes (Figure 1.13) sont similaires aux systèmes AS/RS à charges unitaires, sauf que ces derniers ont des casiers qui peuvent contenir plusieurs charges simultanément (jusqu'à 10 charges). Ces AS/RS ont une densité de stockage élevée grâce à l'utilisation efficace de l'espace

disponible. Ainsi, le stockage a lieu d'un côté du casier tandis que le déstockage a lieu de l'autre côté. La machine S/R peut manipuler les plus profondes charges dans un casier grâce à des plateaux mobiles qui se déplacent de la machine S/R à l'entrée, ce qui permet un déplacement horizontal des charges (Sari 2003).

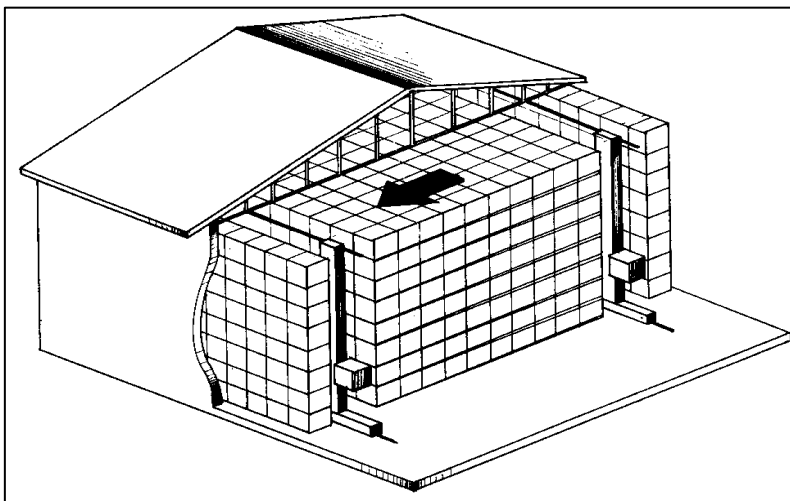


Figure 1. 13: AS/RS à étagères profondes

1.3.8. Les AS/RS à convoyeur gravitationnel

Un AS/RS à convoyeur gravitationnel (Figure 1.14) est un AS/RS multi profondeur, chaque casier est composé de plusieurs positions, et chaque position est conçue pour le stockage d'une seule charge. Cette configuration ne comporte qu'un seul rack et utilise deux machines. La première est responsable des opérations de stockage et la seconde pour les opérations de déstockage. Les deux machines opèrent sur deux côtés parallèles du rack. En effet, les opérations de stockage sont effectuées depuis un côté du rack (face de stockage) et les opérations de déstockage sont effectuées depuis le côté parallèle (face de déstockage). Le rack est incliné de la face de stockage vers la face de déstockage, de sorte que, lorsque la machine de stockage dépose une charge dans un casier, la charge glisse le long de ce dernier jusqu'à atteindre la première position libre. Les deux côtés du rack sont reliés par le convoyeur de restockage, incliné à l'inverse de l'inclinaison des casiers, son rôle est de transporter les charges à stocker dans le rack. Au cours d'une opération de stockage, la machine de stockage prend une charge soit à la station d'entrée, soit à l'extrémité finale du convoyeur de restockage, se déplace vers l'emplacement où la charge doit être stockée, la dépose et revient à sa position initiale. Pour déstocker une charge, la machine de déstockage commence par se déplacer de sa position initiale vers le casier dans lequel l'opération de déstockage sera effectuée, elle prend une par une les charges qui précèdent la charge souhaitée dans le casier et les transporte individuellement vers l'extrémité initiale du convoyeur de restockage, elle retourne ensuite au casier pour prendre la charge souhaitée et la transporter vers la station de sortie.

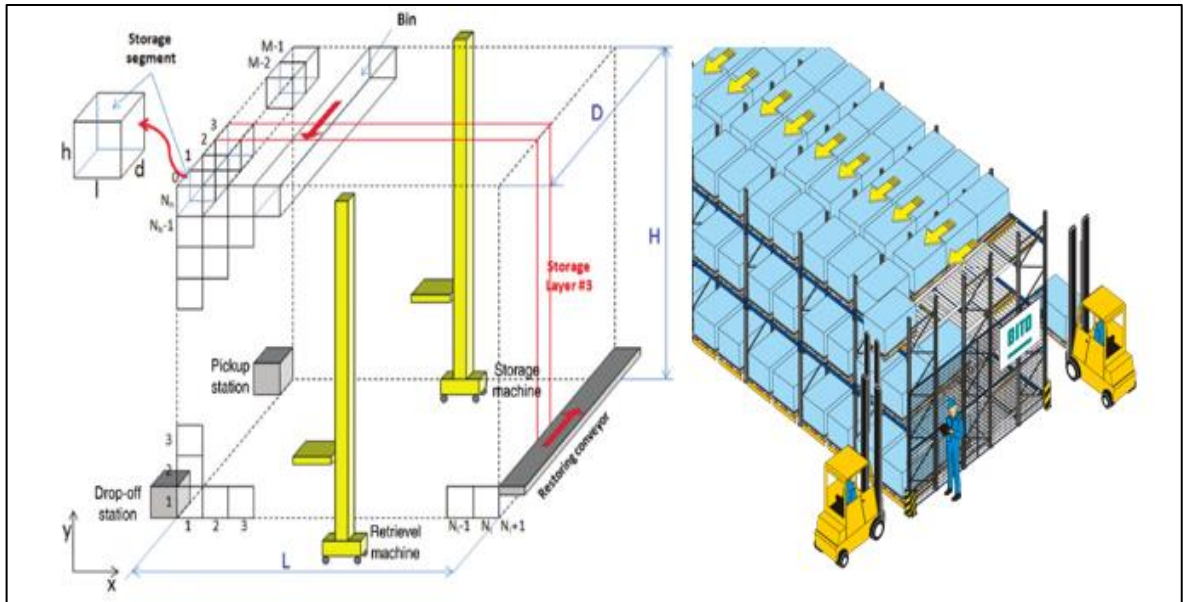


Figure 1. 14: AS/RS à convoyeur gravitationnel

1.3.9. Les AS/RS à convoyeur gravitationnel Bidirectionnel

Le bi-directional flow-rack (Figure 1.15) est l'une des dernières innovations dans la même famille des AS/RS compacts, en matière de structure de ces systèmes est le bi-directional flow-rack. Présenté dans la littérature pour la première fois par (Chen, Li, and Gupta 2015).

Le système en question, prend en compte les deux systèmes à convoyeurs gravitationnels classique et à une seule machine, parce que la disposition d'un ensemble de casiers superposés, mis côte à côte, et penchés par alternance vers l'une ou l'autre des faces du rack tout en utilisant deux machines différentes de part et d'autre (Figure 1.15). Cette approche architecturale, permet d'une part de garder l'avantage d'un casier en U tout en éliminant le virage en épingle, mécaniquement il est difficile à réaliser de manière performante un glissement délicat du produit, car le déplacement de l'article d'un casier vers celui qui lui est correspondant (changement de direction) est effectué par la machine -qui selon les cas est- secondaire.

Ce changement de l'orientation des casiers, et la suppression du convoyeur de restockage, en comparant avec un AS/RS à convoyeur gravitationnel classique, permet de meilleurs performances (temps de cycle) et du coût (acquisition et utilisation), mis en évidence par (Chen, Li, and Gupta 2015) par l'appui des modélisations continues du système en prenant en compte des équations du nombre moyen d'articles bloquants pour un déstockage. En comparaison avec le système à convoyeur gravitationnel à une seule machine, l'évaluation est plus délicate, puisqu'il s'agit de comparer deux quantités de natures différentes, qui sont coût et temps, mais dont la correspondance peut tout de même être approchée. Car l'avantage du nouveau système est qu'il

est plus rapide, mais son inconvénient est qu'il dispose de deux machines contrairement à son prédécesseur (flow-rack à une seule machine) (Hamzaoui 2020).

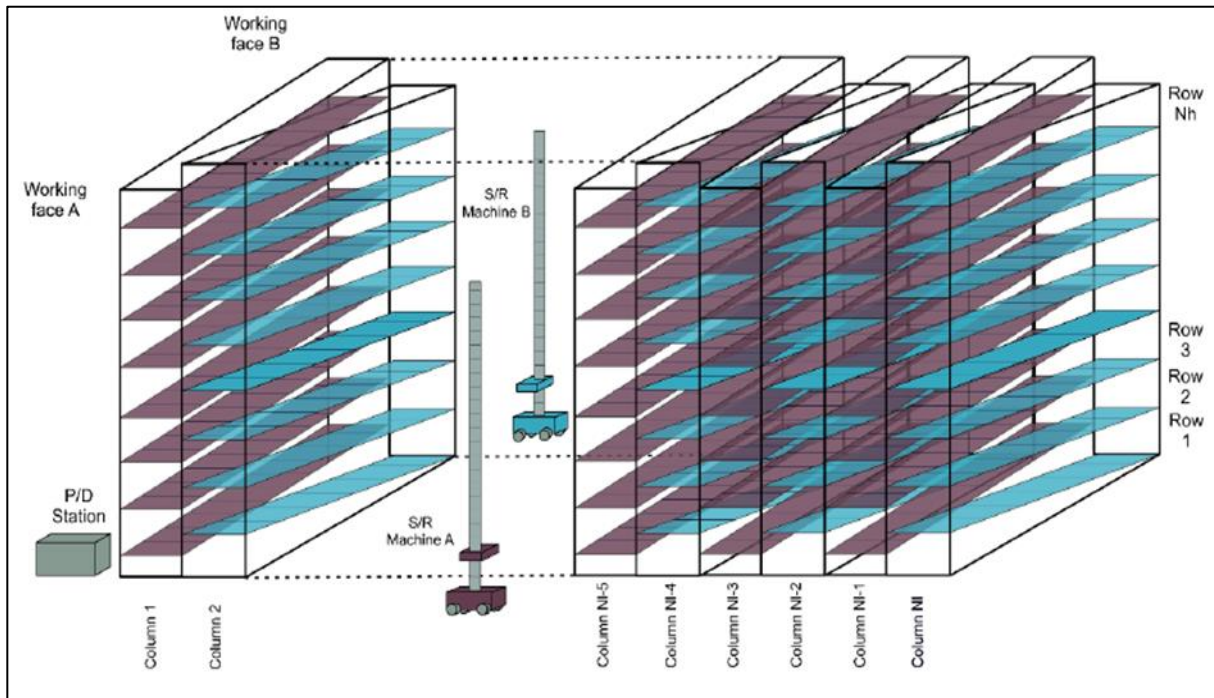


Figure 1. 15: AS/RS à convoyeur gravitationnel Bidirectionnel

1.3.10. Les AS/RS à convoyeur gravitationnel à une seule machine

Cette variante (Figure 1.16) de l'AS/RS à convoyeurs gravitationnels à deux machines, a été adoptée dans le domaine des systèmes de stockage automatisés par (Sari and Bessenouci, 2012). L'avantage principal de ce système, est dans la forme de ses casiers qui ont la forme d'un U, au lieu d'être parallélépipédique. En plus cette forme-là assure que chaque casier se constitue de deux bacs reliés à leur extrémité. De telle sorte que les opérations de stockage et de déstockage se font par le bac dédié. Cette structure particulière permet que le convoyeur gravitationnel puisse faire glisser les articles stockés à travers le bac de stockage et de changer de direction une fois arrivés au fond du casier, par la suite de glisser une nouvelle fois à travers le bac de déstockage jusqu'à rencontrer le produit stocké auparavant ou bien la fin du bac (c'est-à-dire la face du rack dans ce cas-là).

Cette ingéniosité d'architecture nous donne la possibilité de stockage d'une grande diversité d'articles parce qu'ils sont tous accessibles d'une manière ou d'une autre, de plus on n'a pas obligatoirement besoin d'une politique de stockage déterministe puisque la politique de stockage aléatoire permet l'utilisation de tout l'espace disponible au stockage tout en gardant l'équilibre des charges dans tout le volume du système. Autre propriété importante, c'est la réduction du nombre de machines S/R de deux à une seule en plus de la suppression du convoyeur de restockage, ce qui diminue considérablement le coût d'investissement initial et les coûts de fonctionnement et maintenance (Hamzaoui 2020).

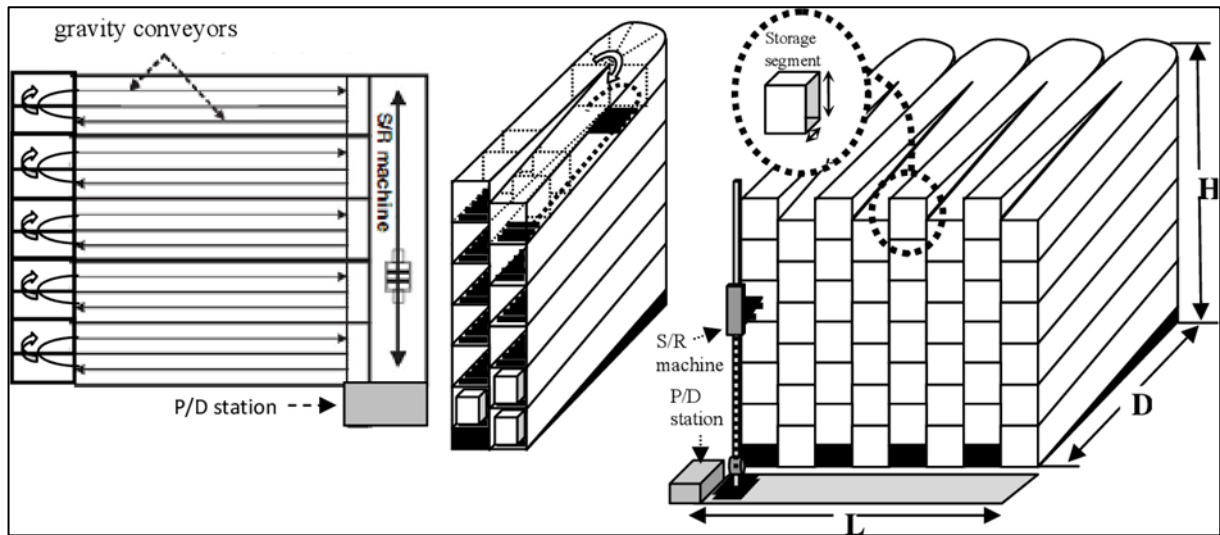


Figure 1. 16: AS/RS à convoyeur gravitationnel à une seule machine

1.4. Méthodologies de Stockage

Les systèmes de stockage sont classés selon la manière de stockage des produits. Ceci résulte en deux types principaux pour ces systèmes. Les systèmes de stockage dédiés où chaque produit dans le système a un espace, ou un emplacement à lui. Quand le produit peut se placer n'importe où d'une façon aléatoire, on parle donc des systèmes ouverts. Ces derniers sont équipés d'un système de commande qui possède une mémoire pour l'emplacement de chaque produit stocké dans le système, ainsi que les emplacements vides pour les futurs stockages.

1.4.1. Stockage dédié

Chaque produit dans ce type de système de stockage a un emplacement prédéterminé. Cette façon de stockage a plusieurs avantages (Sari 2003):

- **Simplicité de contrôle du système et de la gestion des stocks**, étant donné que chaque emplacement est destiné à un produit unique, il n'y aura pas de complications lors des opérations de stockage et de déstockage des produits.
- **Simplicité de la conception du système**, pour la même raison que chaque emplacement est destinée à un produit. Les caractéristiques de ces emplacements sont elles-mêmes prédéterminées (taille des casiers, des allées... etc.).

Quelques inconvénients qui touchent ce système de stockage ont été listés par (Sari 2003). Le fait que chaque produit va avoir des espaces déterminés, un grand nombre d'emplacements sera dédié pour chaque produit, ce qui peut causer des taux de remplissage faibles. Aussi, ce type de système est moins flexible, et les modifications peuvent être beaucoup plus complexes et coûteuses.

1.4.2. Stockage aléatoire

Ce type de système présente une solution aux inconvénients du système de stockage dédié. Dans les systèmes de stockage aléatoire ouverts, chaque produit est stocké dans un emplacement vide au hasard, mais avec le système de commande intégré, cet emplacement sera marqué comme occupé, avec des informations sur le produit qui l'occupe. Ce système demande un certain niveau d'expertise pour manipuler le système de commande.

Dans ce même type de systèmes de stockage, on trouve aussi les systèmes de stockage aléatoire par classe. La façon de stockage des produits se fait de la même manière que les systèmes de stockage aléatoire ouverts. La différence c'est que les produits ne sont pas stockés pour la même durée afin de réduire le temps de cycle; ce, à travers un système de classement:

- Les emplacements qui sont près de la station de livraison sont de la première classe.
- Les emplacements les plus loins de la station de livraison sont de la dernière classe.

Les produits sont classés à leur tour selon le temps qu'ils vont être stockés dans le système, et ceux qui restent le moins de temps de stockage seront placés dans les emplacements de la première classe.

En général, les classes sont limitées à trois (Figure 1.17): "A" représente la classe des produits ayant un taux de rotation élevé, "B" comprend la classe des articles ayant un taux de rotation moyen et "C" couvre le reste.

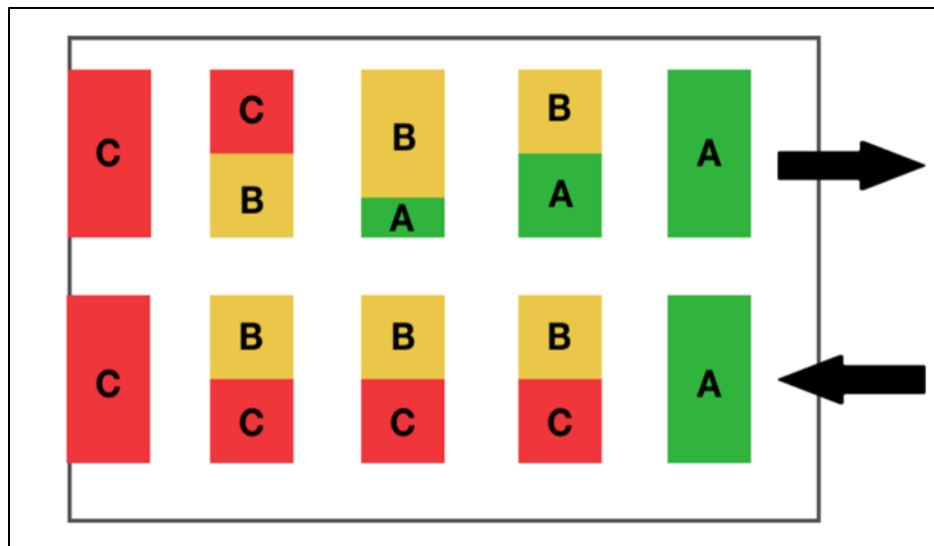


Figure 1. 17: Stockage par classe

1.5. Etat de l'Art

Les systèmes AS/RS ont été introduits dans les années 1960 pour un but de manipuler les lourdes charges de palettes. Soixante plus tard, les entreprises cherchent toujours à améliorer les performances de ces systèmes afin de traiter les marchandises, à expédier les produits plus rapidement aux clients et optimiser l'espace dans l'entrepôt, pour cela, deux catégories de

Chapitre 1: Les Systèmes Automatisés de Stockage/Déstockage à profondeur multiple

décisions doivent être prises, la première concerne la partie de conception, c'est-à-dire les décisions qui sont prises avant l'installation de système comme le choix de système AS/RS, le nombre d'allée, les dimensions du rack, le Nombre et l'emplacement des points d'E/S, Nombre et emplacement de machine S/R par allée..., et la deuxième concerne la partie d'exploitation (contrôle) c'est-à-dire les décisions qui sont prises pendant l'utilisation comme La méthodologie de stockage, le séquençement des produits, le type d'opération (simple cycle ou double)... Une liste des décisions pour chaque catégorie est présentée dans le tableau 01.

Tableau 1. 1 : Les décisions à prendre pour chaque classe de recherche

Les axes de recherches	Les classes des axes de recherches	Les décisions à prendre
La conception (des décisions prises avant l'installation du système)	le choix du système AS/RS	
	La configuration de système AS/RS	• Le nombre d'allées • Les dimensions du rack : largeur, hauteur, profondeur. • Les dimensions du casier : largeur, hauteur, profondeur. • La structure des racks. • Le nombre et l'emplacement des points d'E/S. • La capacité tampon aux points d'E/S. • Le nombre et l'emplacement de machine S/R par allée. • Le nombre de préparateurs de commandes par allée (le cas échéant).
Le contrôle (des décisions prises pendant l'utilisation du système)	La méthodologie de stockage	• La méthode d'affectation des stocks. • Le nombre de classes de stockage. • Le positionnement des classes de stockage.
	le séquençement	• Des contraintes de séquençement (par exemple : des dates limites). • Le type d'opération (commande simple ou double). • La méthode de séquençement.

	Batching (Mise en lot)	• La taille du lot (en fonction de la capacité ou du temps). • La règle de sélection pour l'affectation des ordres aux lots.
	Dwell point	• Le type de positionnement (statique ou dynamique). • La règle de positionnement (par exemple : Dernier emplacement, Point médian, etc...).

1.5.1. Les AS/RS à une seule profondeur

On a trouvé plusieurs travaux reliés aux AS/RS, leur conception et aux leur performance, notamment Firth (1988), Hatton (1990) et Mulcahy (1994) qui sont développé le travail précédent de Heskett, Glaskowsky, and Ivie (1973) en ajoutant des nouvelles caractéristiques décrivant les entrepôts et leur conception y compris la détermination des exigences, la conception des systèmes de manutention, l'élaboration de l'aménagement, la reconnaissance de l'entrepôt dans l'ensemble du réseau de distribution et la comparaison d'approches alternatives.

Mohsen and Hassan (2002) a créé une procédure facilitant le design et la configuration des entrepôts en considérant quelques caractéristiques clés comme: la modularité, l'adaptabilité, la compacité, l'accessibilité, la souplesse et la répartition des mouvements.

Après une étude approfondie sur plusieurs centres de distribution, Dallari, Marchet, and Melacini (2009) ont conçu une méthode qui aide les décideurs sur quel type de système d'entrepotage est le mieux adapté à leurs besoins.

Bozer and White (1984), Park et al. (2003), Kouloughli, Sari, and Sari (2010), Kouloughli and Sari (2015) et Yang et al. (2015) ont été intéressés à l'optimisation des dimensions du système AS/RS: c'est-à-dire trouver les meilleures dimensions qui assurent un fonctionnement optimal vis-à-vis le temps de cycle.

Bozer and White (1984) ont développé des expressions du temps de déplacement prévu pour analyser quatre règles statiques de positionnement du point de repos de la machine S/R. Egbelu (1991) a étudié le problème de positionnement du point de repos de la machine S/R pour minimiser le temps de cycle, en utilisant la programmation linéaire. Randhawa, McDowell, and Wang (1991) et Randhawa and Shroff (1995) ont construit des modèles de simulation pour évaluer l'effet du nombre et de l'emplacement des points d'E/S sur la performance du système. Les résultats montrent que le placement de la station d'E/S au milieu de l'allée donner un débit plus élevé que celui situé à l'extrémité de l'allée. Hale et al. (2015) ont effectué une recherche qui vise à trouver l'emplacement optimal de la station de repos de la machine S/R dans un AS/RS à multi allées. Huq et al. (2018) ont travaillé sur la localisation de la station de repos de la machine S/R afin que le temps de réponse soit minimisé.

Certains chercheurs concentrent sur la modélisation du temps de cycle Bozer and White (1984) ont développé des modèles analytiques pour le simple cycle et le double cycle pour les AS/RS à charge unitaire. Ces modèles sont basés sur un stockage et un déstockage aléatoire. Hwang and Lee (1990) développent des modèles de temps de cycle pour le cycle simple et le cycle double suivant une politique d'allocation de stockage aléatoire, en considérant un taux d'accélération et de décélération constant et une limitation de la vitesse maximale. De plus, Chang, Wen, and LIN (1995) ont construit des modèles de temps de cycle en considérant les profils de vitesse qui existent dans le monde réel. SARI (1998) a développé des expressions analytiques discrètes pour les AS/RS multi allées et les AS/RS à racks glissants.

Wen, Chang, and Chen (2001) ont amélioré les modèles de temps de cycle des travaux de Chang, Wen, and LIN (1995) en considérant la politique d'affectation de stockage par classe et rotation complète au lieu de la politique d'affectation de stockage aléatoire. Ghomri et al. (2008) ont développé des modèles du double cycle pour l'AS/RS multi-allées. Pour le système de carrousel, Hwang, Song, and Kim (2004) ont également présenté des modèles de temps de cycle tenant en compte les profils de vitesse du mouvement du corps du carrousel.

D'autre côté, le control des AS/RS a fait l'objet de plusieurs travaux de recherche. Hausman, Schwarz, and Graves (1976) ont calculé le temps de déplacement pour un système AS/RS à charges unitaires sous différentes politiques de stockage, notamment le stockage aléatoire, le stockage basé sur le retournement complet et le stockage basé sur les classes. Eynan and Rosenblatt (1994) proposent de diviser le rack en classes de stockage et calculer le temps de déplacement. Ils ont montré que les meilleures performances sont obtenues avec un petit nombre de classes. Pan and Wang (1996) ont développé le modèle de temps de déplacement à double cycle pour le rack carré sous une politique de stockage par classe. Une règle de séquence de déstockage du type FIFO a été utilisée. Eldemir, Graves, and Malmberg (2004) ont développé un modèle analytique pour AS/RS dans le but de déterminer le temps de cycle et l'espace requis pour un stockage par classe. Ils sont arrivés à la conclusion que le stockage par classe nécessite moins d'espace que le stockage aléatoire ou dédié.

Pour bien améliorer les performances des systèmes AS/RS, (Eben-Chaïme 1996) et (Eben-Chaïme and Pliskin 1997) disent qu'on en a besoin de moins de machines S/R si le système AS/RS fonctionne en régime à simple et à double cycle.

Elsayed and Lee (1996) ont pu minimiser le retard global des ordres de déstockage quand ces derniers sont aux dates connues; en utilisant une méthode qu'ils ont développée, qui forme et ordonne des doubles cycles en associant les ordres de stockage et de déstockage. D'autres travaux ont élaboré des techniques plus sophistiquées pour des cas beaucoup plus précis où l'incertitude joue le rôle du facteur déterminant: Chetty and Reddy (2003) proposent l'utilisation des algorithmes génétiques pour ce type de problèmes, alors que les travaux de (Seidmann 1988) et (Wang and Yih 1997) trouvent plus d'efficacité dans les réseaux neuronaux et l'intelligence artificielle (respectivement).

Han et al. (1987) ont proposé l'heuristique la plus proche NN pour trouver le séquençement des requêtes de déstockages, l'heuristique de la plus proche consiste à choisir à chaque fois le produit qui a un temps de déstockage minimal dans le lot et déstocker. Van Den Berg (1996) ont développé une approche d'ordonnancement des requêtes de stockage et de déstockage pour obtenir

un temps de cycle total minimal. Bertolini et al. (2019) ont proposé un algorithme métaheuristique qui optimise les performances d'un entrepôt automatisé pendant le déstockage; l'algorithme est basé sur la recherche taboue.

Pour obtenir des informations plus détaillées sur la conception et le contrôle des AS/RS, nous faisons référence aux articles (de Koster, Le-Duc, and Roodbergen 2007) et (Roodbergen and Vis 2009) Une vue d'ensemble récente de l'évaluation des performances et des modèles de temps de cycle pour les AS/RS peut être trouvée dans les revues de et (Hameed, 2019.).

1.5.2. Les AS/RS Multi profondeur

Depuis 2005, nombreux systèmes de stockage et de déstockage automatisés (AS/RS) multi profondeurs ont été introduites, tels que les AS/RS double profondeurs (T. Lerher et al. 2010), Les AS/RS à convoyeur gravitationnel (Sari, Saygin, and Ghouali 2005), les rack mobile, les systèmes basés sur des puzzle (Kota, Taylor, and Gue 2015), les AS/RS 3D compacts (de Koster, Le-Duc, and Yugang 2008), et L'AS/RS à convoyeurs gravitationnels bidirectionnels (BFR) (Hamzaoui and Sari 2020). La particularité de ces systèmes est que chaque casier possède la capacité de stocker plusieurs produits, mais elles sont plus difficiles à contrôler qu'un AS/RS à une seule profondeur. Dans ce qui suit, nous avons essayé de présenter les différents travaux concernant les AS/RS multi profondeur en tenant compte les deux catégories de décision : la conception et le contrôle.

1.5.2.1. Temps de cycle

Une revue bibliographique importante est explorée la manière de concevoir un système compact multi profondeur efficace. Des mesures de performance telles que le temps de déplacement par cycle d'opération S/R, le nombre de demandes S/R exécutées par unité de temps et le coût moyen par opération S/R ont été utilisées pour optimiser la conception du rack. Le temps de cycle de la machine S/R est considéré comme le facteur le plus important lors de l'amélioration et l'évaluation de la performance d'un système compact multi profondeur, qui dépend du rapport entre les différentes dimensions, des politiques de stockage, du modèle d'opération de la machine S/R, des politiques de point d'arrêt, du profil de vitesse de la machine S/R et de la profondeur du rack (mono- ou multi profondeur).

Les modèles de temps de cycle ont été développés pour la première fois par (Gudehus 1972) et plus tard par (Y. A. Bozer, Shiek, and White 1982) et (Yavuz A. Bozer and White 1984) pour des systèmes de stockage et de déstockage à profondeur unique.

Lerher et al. (2010) ont présenté des modèles analytiques de temps de déplacement pour le calcul des temps de cycle dans les systèmes de stockage et de déstockage automatisés à double profondeur et à charge unitaire (AS/RS). Ces modèles nous permettent de calculer le temps de cycle moyen du simple cycle et double. Les résultats des modèles analytiques de temps de cycle proposés sont en corrélation avec les résultats des modèles de simulation des AS/RS à double profondeur. Le même principe est appliqué au temps de cycle d'un système de stockage et de déstockage basés sur des navettes (SBS/RS) à double profondeur (Lerher 2016).

Xu et al. (2015) ont développé des modèles analytiques de cycle de type quadruple pour estimer la performance d'un certain système (les systèmes AS/RS à double profondeur et à double navette et les systèmes traditionnels AS/RS à une seule profondeur et à une seule navette), puis ont étendu le temps de déplacement proposé à un cas spécial où les emplacements de stockage vides doubles auront lieu et certaines idées de gestion sont données.

Après les modèles pour les AS/RS à un et double profondeurs, des modèles de temps de cycle pour les systèmes AS/RS compact multi profondeur ont été développés. En s'inspirant des travaux antérieurs de (Bozer and White 1984), de Koster, Le-Duc, and Yugang (2008) ont été les premiers à analyser les performances du système AS/RS tridimensionnel compact et de le dimensionner de manière optimale. Une expression de forme fermée a été développée pour le simple cycle et pour le double cycle, ils ont proposé une expression approximative.

Par la suite Yu and de Koster (2009) ont proposé un modèle analytique pour les systèmes (AS/RS) tridimensionnels compacts, ce modèle permet de déterminer les dimensions optimales du rack en minimisant le temps de cycle. Yu and de Koster (2009b) ont formulé un modèle de programmation non linéaire en nombres entiers mixtes pour déterminer les limites de la zone d'un système AS/RS compacts.

Hao, Yu, and Zhang (2015) ont présenté des expressions sous forme fermée sont développées pour la dérivation du temps de déplacement. Yang et al. (2017) ont présenté le temps de déstockage prévu pour les simples cycles en adoptant un modèle de vitesse exact et ils développent le modèle pour concevoir de manière optimale un rack AS/RS tridimensionnels compacts. Xu et al. (2018) Ont construit des modèles analytiques de temps de cycle à commande double pour Les systèmes AS/RS tridimensionnels compacts.

Sari, Saygin, and Ghouali (2005) ont été les premiers à examiner un système AS/RS à convoyeur gravitationnel, ils ont développé des expressions de temps de cycle en forme fermée pour ce système. Ces expressions sont basées sur une approche continue et sont comparées par simulation à des modèles exacts basés sur une approche discrète.

Par la suite Sari, Grasman, and Ghouali (2007) ont étudié la performance des configurations de systèmes AS/RS à convoyeur gravitationnel en tenant compte d'un large mélange de différents types de produits. Fan et al. (2015) ont étudié le temps de cycle de la machine S/R dans un AS/RS multi profondeur, basé sur les techniques de simulation. Les auteurs ont construit un modèle contenant 3600 types de plans puis simulé dans le but de mesurer le temps de déplacement de la machine S/R. l'analyse de régression multiple a été utilisée pour le traitement des données.

Ghomri and Sari (2017) ont développé un modèle analytique de temps moyen de déstockage dans un système AS/RS à convoyeur gravitationnel sous une politique de stockage aléatoires. Ce modèle permet d'évaluer les performances et la conception de l'AS/RS à convoyeur gravitationnel.

Ghomri and Cardin (2017) ont développé une formule empirique du rangé moyen de déstockage dans un système de stockage/ déstockage automatisé multi profondeur avec des heuristiques aléatoires pour le stockage et le déstockage. Ce qui facilite de déduire le temps moyen de déstockage. Ce calcul est basé sur des simulations multiples de différents modèles de systèmes AS/RS et sur une régression sur les données obtenues. Le cas particulier de ce système AS/RS à

convoyeur gravitationnel sera considéré pour illustrer cette contribution. Il sera ensuite possible d'utiliser cette formule pour d'autres systèmes multi profondeurs.

Lehmann and Hußmann, (2021) ont développé un modèle de temps de déplacement avec une approximation continue des racks de stockage d'un AS/RS multi profondeur en expression à forme fermée. Le calcul des probabilités de relocalisation dans ce modèle est basé sur une structure homogène d'allocation des produits de stockage qui conduit à une allocation symétrique des produits de stockage. Ce modèle détermine les temps de déplacement moyens pendant un stockage ou un déstockage à simple cycle, double cycle, la probabilité de relocalisation et le nombre de relocalisations.

Sari and Bessenouci, (2012) ont présenté une nouvelle variante du système de stockage et de déstockage automatisé (AS/RS) à convoyeur gravitationnel utilisant une seule machine pour les opérations de stockage et de déstockage à la place de deux machines. Des expressions analytiques sont également dérivées pour simple cycle et double cycle.

Chen, Li, and Gupta (2015) ont développé un modèle de temps de cycle pour les systèmes AS/RS à convoyeur gravitationnel bidirectionnel BFR. Dans le même système Hamzaoui and Sari (2020) ont développé des modèles continus dérivés et des modèles discrets permettant de déterminer le temps nécessaire pour réaliser les opérations de stockage et de déstockage dans AS/RS à convoyeur gravitationnel bidirectionnel.

Metahri and Hachemi (2017) ont proposé une comparaison des performances entre deux types de AS/RS (Free-Fall à convoyeur gravitationnel et à convoyeur gravitationnel classique), consiste à mettre en valeur les avantages de l'utilisation d'un système AS/RS à convoyeur gravitationnel de type FF dans les entrepôts. Les critères de comparaison utilisés dans cette étude sont : le temps de déplacement moyen pour le déstockage, le temps de déplacement total pour le déstockage, et le débit d'une demande client. Les résultats obtenus montrent que FF- AS/RS à convoyeur gravitationnel minimise de manière significative les critères de comparaison.

La première étude sur les performances d'un AVS/RS a été proposée par Malmborg (2002), qui a présenté des modèles mathématiques pour le temps de déplacement d'un SVA/RS en fonction du nombre d'allées, de véhicules et d'ascenseurs, de niveaux, en considérant une politique de stockage aléatoire et une configuration de niveau à niveau.

Manzini et al. (2016) ont développé un ensemble des modèles analytiques pour les systèmes AVS/RS. Dans le même système D'Antonio and Chiabert (2019) ont proposé un modèle analytique capable d'évaluer la performance d'un système AVS/RS multi-navettes et multi profondeur. Ce modèle vise à soutenir les phases de conception et de déploiement des AVS/RS en permettant d'évaluer le temps de cycle et le débit attendus.

D'Antonio and Chiabert (2019b) ont présenté de nouveaux modèles analytiques capables d'évaluer la performance des AVS/RS concernant les racks profonds. Ces modèles prennent en compte les critères réels liés au stockage et déstockage des articles, considèrent la capacité de la navette et du satellite à effectuer simultanément différentes tâches et évaluent l'écart-type du temps de cycle.

Une technologie similaire à l'AVS/RS est appelée système de stockage et de déstockage par navette (SBS/RS). Marolt, Kosanić, and Lerher (2021) mentionnent une simulation à événements discrets d'un AVS/RS multi profondeurs avec une navette captive à étages a été développée.

Eder (2020) a présenté une méthode permettant de déterminer les performances des (SBS/RS) avec des navettes captives à une allée et un stockage à profondeur multiple a été proposée. La base de cette méthode de calcul est un système de file d'attente ouvert en temps continu avec une capacité limitée. Une approche basée sur les probabilités est appliquée dans la méthode pour prendre en considération le stockage multiple, le temps de cycle a été déterminé par une approche de valeur spatiale. Les résultats de cette méthode montrent que si on augmente la profondeur de stockage à une certaine valeur, le débit augmente et le coût diminue.

Le même travail a été présenté dans Eder (2020b), sauf qu'il utilise une distribution uniforme des emplacements de stockage, les résultats du ce dernier montrent comment ce modèle de calcul peut être utilisé pour concevoir des SBS/RS qui répondent à des exigences prédéfinies.

Gue and Kim, (2007) Ont présenté et développé des modèles pour un système de stockage/déstockage basé sur le 15-puzzle dans lequel 15 tuiles numérotées glissent dans une grille de 4*4. Ces modèles permettent de comparer les conceptions basées sur les puzzles avec les conceptions traditionnelles basées sur les allées.

Kota, Taylor, and Gue (2015) ont développé Des expressions à forme fermée pour le temps de déstockage dans le cas où le nombre d'escortes libres, placées au hasard, est égal à un ou deux. L'objectif était d'accroître les résultats analytiques pour déterminer la performance du temps de déstockage.

1.5.2.2. Optimisation des dimensions des AS/RS compacts

Les modèles de temps de cycle basés sur optimisation des dimensions du rack multi profondeur ont été présentés par nombreux chercheurs depuis l'étude de de Koster, Le-Duc, and Yugang (2008) l'objectif était d'analyser les performances du système et le dimensionner de manière optimale.

Ce travail concerne les AS/RS tridimensionnels compacts de nouvelle conception. Une expression analytique a été proposée pour le simple cycle et le double cycle, En s'inspirant par Bozer and White (1984) pour les systèmes AS/RS à une seule profondeur aux systèmes AS/RS tridimensionnels compacts. Dans la continuité de leurs travaux de Yu and de Koster (2009a) ont proposé un modèle analytique pour les AS/RS tridimensionnels compacts, ce modèle permet de déterminer les dimensions optimales du rack en minimisant le temps de cycle, le temps de cycle moyen d'un simple cycle a été dérivé sous une politique de stockage basée sur le stockage par classe. Afin de étudié la politique de stockage par classe, Yu and de Koster (2009b) ont formulé un modèle de programmation non linéaire en nombres entiers mixtes pour déterminer les limites de la zone d'un système AS/RS compacts. Un algorithme de décomposition et un schéma de recherche tridimensionnel sont développés pour résoudre le modèle.

Les résultats montrent que les dimensions des zones et les tailles des zones influence de manière significative le temps de déplacement des machines S/R. Yang et al. (2017) ont étudié la

conception d'un rack optimal pour un AS/RS tridimensionnels compacts dans le cadre d'une politique de stockage basée sur la rotation complète. L'étude a été basée sur trois scénarios. Pour chaque scénario, ils précisent le temps de déstockage prévu pour le simple cycle en adoptant un modèle de vitesse exact et ils développent le modèle pour concevoir de manière optimale un rack AS/RS tridimensionnels compacts.

Ce modèle nous permet de déterminer plus précisément les dimensions optimales du rack pour divers scénarios. Pour le même système Hao, Yu, and Zhang (2015) ont étudié la conception des systèmes de stockage tridimensionnels compacts, où la machine d'E/S est située au milieu du rack de stockage. Des expressions sous forme fermée sont développées pour la dérivation du temps de déplacement et l'optimisation des dimensions du rack. Xu et al. (2018) ont construit des modèles analytiques de temps de cycle à commande double (DC) pour les systèmes AS/RS tridimensionnels compacts. Ces modèles permettent d'optimiser les dimensions du système.

Les résultats montrent que le double cycle est rapide par rapport au simple cycle en termes de temps de cycle d'une commande. Atz, Lantschner, and Günthner, (2012) ont décomposé les cycles de commande en petites fractions de temps de cycle séparées et en les synthétisant dans les différents cycles de commande spécifiques pour calculer le débit d'un système AS/RS à une seule profondeur et multi profondeur. Les composantes de temps représentent les mouvements typiques et les étapes de manipulation de la charge dans la machine S/R. les auteurs ont modélisé et calculé l'ensemble des différents types à l'aide de la simulation de Monte Carlo.

Ce modèle permet d'optimiser mathématiquement le dimensionnement des variantes de conception.

Pour l'AS/RS à convoyeur gravitationnel Ghomri and Sari (2015) ont modélisé mathématiquement le temps de déplacement moyen de la machine de déstockage en cas de stockage aléatoire. Ce modèle est en fonction des paramètres physiques du système tels que la longueur, la largeur, la profondeur, etc. Et peut être utile pour l'évaluation des performances.

Liu et al. (2014) ont présenté le temps de déplacement minimum attendu pour le simple cycle et la dimension optimale du rack d'un nouveau système AS/RS compact à une seule machine S/R. Hamzaoui and Sari (2015) cherchent à déterminer les dimensions optimales d'un AS/RS à convoyeur gravitationnel en utilisant une seule machine pour les opérations de stockage et de déstockage, l'optimisation s'est effectuée sur six expressions de temps de déplacement considérant deux positions de point d'arrêt, en utilisant les modèles développés de Sari and Bessenouci (2012).

Après ils ont évalués les dimensions en fonction des temps de déplacement minimaux attendus de la machine de stockage et de déstockage. Les résultats sont étudiés pour examiner l'influence des expressions de temps de déplacement sur les dimensions optimales ainsi que le facteur de forme du rack du système.

L'étude menée par Manzini et al. (2016) a été réalisée pour aider à la conception des AS/RS, un ensemble des modèles analytiques originaux sont développés afin de déterminer le nombre, la profondeur des couloirs et le temps de déplacement pour le simple cycle et à double cycle.

Zaerpour, Yu, and de Koster (2017) ont proposé un modèle non linéaire à nombres entiers mixtes pour optimiser les dimensions d'un système de stockage compacts à cubes vivants en minimisant le temps de déstockage. Des expressions en forme fermée sont obtenues pour le temps de déstockage minimum.

La plupart des études existantes ont supposé que les machines S/R se déplacent à une vitesse constante dans les directions horizontale et verticale, L'effet de l'accélération/décélération dans le temps de cycle a été pris en compte par les auteurs Yang et al. (2015), ils ont étendu les travaux de Koster, Le-Duc, and Yugang (2008) et ont proposé un modèle de conception optimale des racks de stockage pour un AS/RS compact multi profondeur, en tenant compte l'accélération et de la décélération de la machine S/R. Après ils dérivent le temps de déplacement attendu pour le simple cycle sous une stratégie de stockage aléatoire et calculent le rapport optimal entre les trois dimensions qui minimise le temps de trajet dans différents scénarios de profil de vitesse.

1.5.2.3. La position du point repos de la machine S/R et la station d'entrée/sortie (E/S) :

La position du point d'entrée/sortie (E/S) est fait l'objet de nombreux chercheurs dans les AS/RS compact car elle influence le temps de cycle.

Sur un système AS/RS tridimensionnel de Koster, Le-Duc, and Yugang (2008) ont considéré que le rack a une face de déstockage rectangulaire continue, où le point d'E/S est situé dans le coin inférieur gauche. Dans le même système Hao, Yu, and Zhang (2015) ont étudié La conception de systèmes de stockage tridimensionnels compacts, où la machine d'E/S est située au milieu du rack de stockage. Des expressions sous forme fermée sont développées pour la dérivation du temps de déplacement et l'optimisation des dimensions du rack. Les résultats montrent que le modèle produit un temps de déplacement plus court et donc un débit plus élevé par rapport à celui qui situe la machine d'E/S dans le coin inférieur gauche du rack de stockage.

Xu et al. (2020) ont séparé et élevé les points d'entrée et sortie en proposant quatre politiques (points d'E/S sur deux côtés, points d'E/S surélevés sur un côté, points d'E/S surélevés sur deux côté et points d'E/S surélevés en un point central). Après la comparaison, les résultats montrent que la structure carrée dans le temps est optimale dans la plupart des cas et que la politique de point d'entrée/sortie médian est la plus performante en termes de temps de déplacement attendu des machines.

Pour l'AS/RS à convoyeur gravitationnel Sari, Grasman, and Ghouali (2007) ont étudié la performance des configurations de systèmes en tenant compte d'un large mélange de différents types de produits. Plus précisément, des modèles de temps de déstockage attendus sont dérivés, ce qui permet d'analyser l'impact des stations d'E/S et des emplacements des convoyeurs de restockage basés sur un stockage et un déstockage aléatoires.

Des modèles sont construits pour évaluer l'effet du nombre et de l'emplacement des points de repos de la machine S/R sur les performances du système. Bessenouci, Sari, and Ghomri (2012) ont considéré deux positions de point d'arrêt pour déterminer la plus intéressante. Dans le même système Hamzaoui and Sari (2015) ont développé des modèles discrets exacts de temps de cycle pour AS/RS à convoyeur gravitationnel a une machine de stockage /déstockage a la place de deux

machines. Ces modèles ont étudié deux position de point d'arrêt de la machine S/R et le modèle global comprenant les opérations de simple cycle et double cycle.

1.5.2.4. *L'ordonnancement des transactions de déstockage pour les AS/RS :*

Les modes d'exploitation de l'AS/RS comprennent les simples cycles, doubles et multiples. Un grand nombre d'articles étudié le simple cycle et le double cycle. Certains d'entre eux se concentrent sur le problème de séquençage des AS/RS. Chen, Li, and Gupta (2016) ont développé un modèle de temps de cycle pour les systèmes AS/RS à convoyeur gravitationnel bidirectionnel BFR, ce qui nous permet d'obtenir une estimation du débit pour plusieurs configurations de BFR.

Une méthode de génération des opérations double cycle a été proposée pour BFR. Ils ont terminé ce travail par une comparaison des performances via la simulation entre BFR et SFR simple AS/RS à convoyeur gravitationnel. Chen and Gupta (2017) ont proposé une heuristique pour l'affectation du stockage et la génération des opérations double cycle dans un système automatisé de stockage et de déstockage (AS/RS) à convoyeur gravitationnel bidirectionnel (BFR), la politique de stockage à durée de séjour (DOS) a été utilisé.

L'heuristique comporte trois étapes : le groupement, la sélection et le séquençement (GMS).

Li and Chen (2019) ont proposé des heuristiques basées sur la méthode de Hungarian pour générer les opérations double cycle dans Les systèmes automatisés de stockage et de déstockage à convoyeur gravitationnel à une seule machine (BFR AS/RS). Les emplacements de stockage et de déstockage sont déterminés dans cette étude.

Certaines chercheurs se concentrent sur le problème de séquençement des opérations de déstockage pour les AS/RS. Hachemi and Alla (2008) ont proposé une méthode de control d'un système automatisé de stockage/déstockage (AS/RS) à convoyeur gravitationnel basé sur un modèle réseau de Petri Coloré. L'objectif de la méthode est de trouver les meilleurs emplacements dans le rack pour un lot de demande de clients qui minimisant le temps moyen de déstockage.

Dans le même système Bessenouci, Sari, and Ghomri (2012) ont utilisé la recherche tabou et le recuit simulé, deux algorithmes de la métaheuristique connus, pour le control d'un système AS/RS à convoyeur gravitationnel de telle sorte que le temps global de déstockage soit minimisé. Les résultats de ces algorithmes sont comparés aux heuristiques classiques et aux modèles analytiques trouvés dans la littérature de (Gaouar, Sari, and Ghouali 2005; Sari, Saygin, and Ghouali 2005).

Hachemi and Besombes (2013) ont étudié le problème du séquençage des demandes de stockage et de déstockage dans AS/RS à convoyeur gravitationnel. Pour ce problème les auteurs ont sélectionné tous les produits intéressants à prélever en déterminant une limite supérieure de la date d'expiration pour chaque produit demandé, puis trouvent les emplacements qui minimisent le temps moyen de déstockage du lot (demandes de stockage et de déstockage).

Pour les AS/RS tridimensionnels compacts Yu and De Koster (2012) ont proposé une nouvelle heuristique appelée la priorité en pourcentage aux demandes de déstockage avec la distance la plus courte (PPR-SL), pour séquencer une requête (ou un bloc) de demande de stockage et de déstockage dans un système de stockage automatisé multi profondeurs afin de minimiser le

makespan. Les résultats de PPR-SL sont comparés aux l'heuristique de premier arrivé, premier servi (FIFO) et l'heuristique du plus proche voisin (NN).

Dans un système basé sur des puzzles Bukchin and Raviv (2020) ont proposé un ensemble d'outils complémentaires pour résoudre les problèmes de déstockage des charges dans ce système. Une formulation de programmation linéaire en nombres entiers basé sur un graphe étendu dans le temps a été développée pour les problèmes avec un nombre relativement grand d'escortes (cellules), et une heuristique sur mesure basée sur l'approche programmation dynamique a été développée pour les problèmes avec un petit nombre d'escortes (cellules).

1.5.2.5. Les politiques de stockage :

La politique de stockage est également un facteur qui améliore la performance des AS/RS. Pour cette raison, plusieurs chercheurs ont produit de nombreux articles dans ce domaine. Gaouar, Sari, and Ghouali (2005) ont développé une heuristique de stockage/déstockage pour un système AS/RS à convoyeur gravitationnel dans le but de réduire le temps moyen de déstockage, ainsi améliorer ses performances. L'objectif de l'heuristique est de placer les produits identiques dans les mêmes casiers les plus proches de la station de livraison et/ou du point de repos de la machine de déstockage.

Cardin et al. (2012) ont présenté la méthode de stockage In-Deep Class Storage pour les AS/RS à convoyeur gravitationnel, le principe de cette méthode est de dédier les premières couches de chaque casier à la classe des articles les plus demandés. Après ils ont évalué les performances de système (temps de déstockage) par une simulation.

Pour le même système Metahri and Hachemi (2018) proposent un modèle analytique du temps de cycle pour AS/RS à convoyeur gravitationnel à chute libre suivant une politique d'affectation de stockage dédiée. Ce modèle basé sur une approche continue, et comparé par la simulation avec un modèle discret. Les résultats obtenus montrent que la déviation maximale entre le modèle continu et la simulation est inférieure à 5%, ce qui montre la précision du modèle pour estimer le temps de déstockage.

Xu et al. (2019) ont créés des modèles de temps de déplacement pour le système AS/RS multi profondeur, en utilisant la politique de stockage aléatoire et deux politiques de stockage basées sur la classe. Ils ont dérivé la formulation de la taille optimale du système, pour minimiser le temps de déplacement attendu de la machine S/R. la validation des modèles analytiques et les résultats montrent que l'erreur relative maximale entre le résultat analytique et le résultat de la simulation est de 4,3%.

Des simulations numériques sont effectuées pour trouver la taille optimale d'un système AS/RS multi profondeur et pour comparer la performance de ces politiques de stockage. Les politiques de stockage basées sur les classes qui ciblent à la fois la face de déstockage et la direction de la profondeur, ne peuvent pas être meilleures que pour celles qui ne ciblent que la face de déstockage et n'est pas forcément meilleure que celle qui ne concerne que la face de déstockage, mais elle peut être plus difficile à gérer dans le système.

Lehmann and Hußmann (2022) ont présenté quatre modèles analytiques de temps de déplacement pour les systèmes AS/RS multi profondeurs. Deux modèles gèrent les stratégies aléatoires, un modèle minimise la variance du remplissage des casiers de stockage et le quatrième maximise cette variance, après la vérification et la comparaison via la simulation d'évènement discret montre que la stratégie de variance minimale donne des bons résultats (nombre de relocalisations plus bas, temps de déplacement plus bas ...).

Dans le travail de Marolt, Kosanić, and Lerher (2021) une simulation à événements discrets d'un AVS/RS multi profondeurs avec une navette captive à étages a été développée. Les auteurs ont évalué dans le modèle de simulation, la durée de double cycle de neuf combinaisons différentes de stratégies de stockage et d'affectation de relocalisation. Les résultats de cette simulation montrent que les systèmes AVS/RS plus profonds sont plus performants que les systèmes moins profonds.

1.6. Conclusion

Dans ce premier chapitre, nous avons présenté les systèmes automatisés de stockage et de déstockage sous leurs deux formes : profondeur unique et profondeur multiple. L'étude qui est menée dans cette thèse porte sur les systèmes de stockage automatisés à profondeur multiple. Un état de l'art a aussi été présenté dans ce premier chapitre. Dans les chapitres suivants, nous allons présenter les contributions de cette thèse. Qui consiste en l'adaptation de plusieurs méthodes pour l'optimisation du temps moyen de déstockage d'un lot de produits.

Chapitre 2 : Minimisation du temps de déstockage en utilisant l'apprentissage automatique

Chapitre 2

Minimisation du temps de déstockage en utilisant l'apprentissage automatique

2.1. Introduction

Dans ce chapitre nous allons expliquer comment des techniques de l'apprentissage automatique ont été exploitées pour minimiser le temps moyen de déstockage dans un AS/RS profond. La dernière décennie a vu une grande évolution dans le domaine de l'intelligence artificielle en général, et particulièrement dans les domaines des machines learning. Cette dernière est une méthode d'optimisation non formelle, qui donne des résultats très satisfaisants.

Nous allons montrer ci-après comment nous avons adapté plusieurs techniques de l'apprentissage automatique pour l'optimisation de performances de systèmes automatisés de stockage/déstockage profond.

2.2. Machine Learning

Pour comprendre ce qu'est l'apprentissage automatique en tant que terme utilisé dans les journaux et les articles, nous devons considérer ses deux parties "Machine" et "Apprentissage" individuellement :

- Définir le processus d'apprentissage peut s'avérer une tâche très difficile, tout comme l'est la définition de l'intelligence, qui est le produit final de l'apprentissage. La définition la plus simple que l'on puisse trouver pour l'apprentissage dans son sens le plus large serait "l'acquisition de connaissances ou de compétences grâce à l'étude, l'expérience ou l'enseignement" ("Definition of Harvard | Dictionary.Com") ;
- Alors qu'une machine pourrait être définie comme un système mécanique, un appareil, une structure, un dispositif. Qui transforme une source d'énergie en une action mécanique ou autre. Les ordinateurs sont également considérés comme des machines au même titre que les capteurs... Ces systèmes ont des entrées, des sorties et des fonctions qui fonctionnent de manière cohérente afin de produire l'effet désiré (la ou les fonctions principales) de la machine.

L'apprentissage automatique (Figure 2.1), pris dans son ensemble, est l'un des principaux sous-ensembles de l'intelligence artificielle et vise à donner la capacité d'apprendre aux systèmes informatiques en utilisant des données prédéfinies, des exemples et même l'expérience qui aident ces systèmes à "prédire" et à "prendre des décisions". L'ensemble du processus d'apprentissage est répétitif, tout comme pour les humains et les animaux ; avec chaque nouvel élément d'information correct, la machine devient plus apte à prédire avec de meilleures performances ; par exemple, nous disons qu'un logiciel de reconnaissance de texte doté d'une intelligence artificielle est en train d'apprendre lorsqu'il devient de plus en plus rapide à reconnaître des textes. La différence principale et innovante entre les machines (et logiciels) dotés d'une intelligence artificielle et les solutions conventionnelles est que ces dernières utilisent des algorithmes

préprogrammés pour effectuer des tâches, alors que les premières sont autonomes, tout comme un être humain.

L'apprentissage automatique est très utilisé lorsque les problèmes ou les tâches exigent une manière non conventionnelle de les résoudre, comme la reconnaissance, le diagnostic, la planification, le contrôle des robots, la prédiction, etc (Nilsson 1998).

Avec l'augmentation des recherches et des solutions révolutionnaires dans ce domaine, l'apprentissage automatique est devenu l'un des outils les plus puissants dans les différents domaines où il peut être appliqué (informatique, médecine, affaires, sécurité et cybersécurité, ...); il convient de noter que plus les ordinateurs progressent (des GPU, des CPU et des APU plus puissants sortent tous les deux mois), plus les performances des algorithmes de ML augmentent de manière exponentielle ; en conséquence, les nouveaux systèmes qui reposent sur l'apprentissage automatique peuvent aujourd'hui être encore plus performants que les humains dans des tâches où ils n'étaient pas très utiles par le passé(Royal Society (Great Britain) 2017).

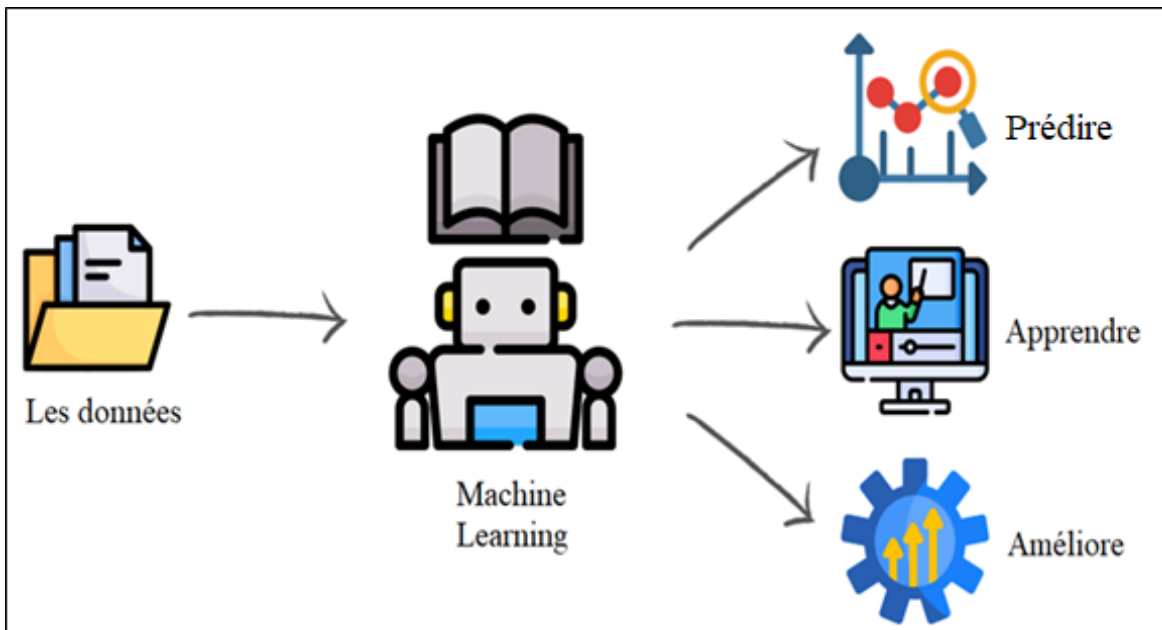


Figure 2. 1: L'apprentissage automatique

De manière très similaire à ce que les gens apprennent de nouvelles choses, soit de manière formelle en allant dans une institution spécialisée, soit en utilisant les différents moyens disponibles pour l'auto-formation (cours et documentations en ligne...), les machines ont les mêmes options en fonction des algorithmes qu'elles utilisent pour apprendre ; Par conséquent, les algorithmes d'apprentissage automatique peuvent être divisés en trois grandes catégories, chacune ayant son propre objectif :

2.2.1. Apprentissage supervisé

Pour mieux comprendre l'apprentissage supervisé (Figure 2.2), on peut utiliser l'analogie suivante : de la même manière que les enfants reçoivent un enseignement à l'école primaire, surtout

dans les premières années, où les enseignants doivent être présents à chaque niveau de l'éducation des enfants afin de leur donner des exemples, de s'assurer qu'ils obtiennent les bonnes réponses et de valider ces réponses, les enfants vont ensuite mémoriser et apprendre à trouver les bonnes réponses par eux-mêmes ; lorsqu'un algorithme d'apprentissage automatique doit être supervisé et ses résultats validés par des humains, on parle alors d'"apprentissage supervisé" ; mais qu'est-ce que cela signifie et comment cela fonctionne-t-il techniquement ?

Il est impératif de savoir ce que sont les ensembles de données avant de plonger plus profondément dans les processus d'apprentissage automatique. En termes simples, les ensembles de données sont des "collections de données, d'exemples, d'informations..." utilisés pour entraîner les modèles, qui sont généralement des fichiers contenant les résultats précédents, ces derniers sont le produit de l'application d'un algorithme ML sur l'ensemble de données d'entraînement (dataset). En utilisant ces modèles, un algorithme ML peut améliorer ses performances, en fournissant de meilleures prédictions et en prenant des décisions plus précises. Dans le cas de l'apprentissage supervisé, ces ensembles de données sont associés aux réponses "correctes" et aux réponses, représentant techniquement le "comportement/résultat" souhaité par le superviseur.

Les superviseurs prépareront des réponses ciblées (target) qui peuvent être de n'importe quel type (qualitatif ou quantitatif ; comme des nombres, des poids, des couleurs, des étiquettes, des classes, ... (Mueller and Massaron 2016)) et les présenteront comme des ensembles de données au système qu'ils veulent "enseigner" ; le système utilisera ensuite ces exemples étiquetés pour apprendre comment les différents points de données sont connectés et structurés ; à ce stade, nous disons qu'un modèle a été formé ; ce modèle sera ensuite utilisé par le système pour prédire les résultats de nouvelles entrées données (Royal Society (Great Britain) 2017);

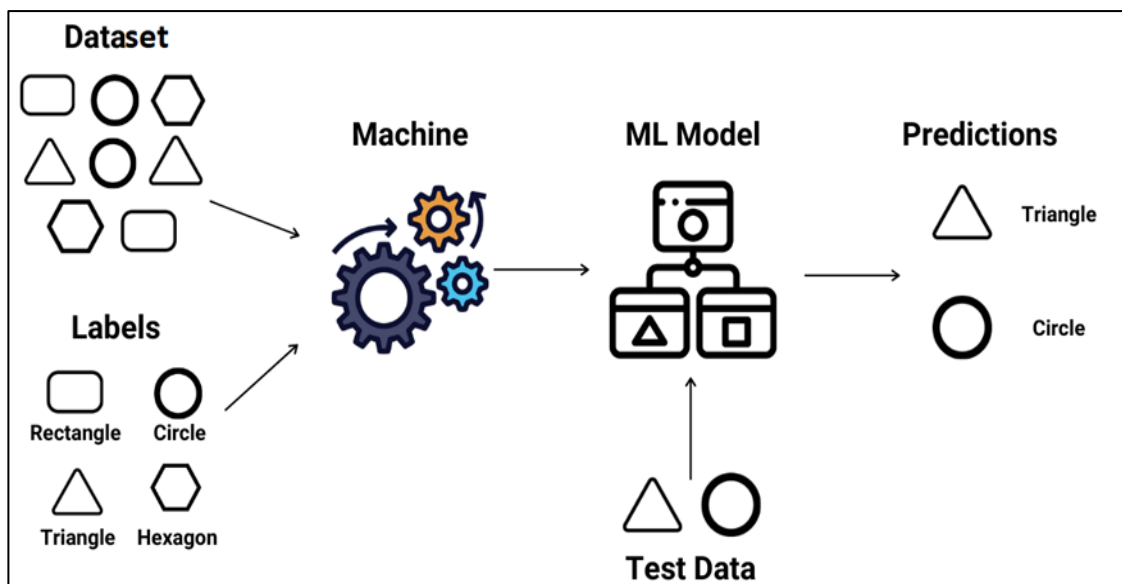


Figure 2. 2: L'apprentissage non supervisé

2.2.1.1. La classification

Comme son nom l'indique, la classification consiste à classer des ensembles d'informations dans des catégories ; les catégories sont généralement déterminées par les attributs que ces points de données ont en commun. Par exemple, nous devons classer des objets selon leur couleur ; dans

ce cas, les couleurs représentent les catégories et peuvent être blanc, noir, rouge, bleu, etc. Le nombre minimal de classes requis est de 2, sans limite maximale (Bishop 2006).

Le type de données dans ce cas est discret (Royal Society (Great Britain) 2017). Cette information est importante et doit être prise en compte avant d'entraîner un modèle et de le mettre au travail dans des applications réelles comme la reconnaissance d'objets/de caractéristiques, les applications médicales, la reconnaissance de visages, etc.

A. L'arbre de décision

Un arbre de décision est une structure graphique qui ressemble à un organigramme et comporte des nœuds et des branches. Il est utilisé pour cartographier des informations et constitue un outil très important dans de nombreux domaines, notamment les affaires, les mathématiques, les opérations de sécurité, l'IA... Les différents nœuds représentent des "tests" (par exemple, nous observons si le résultat d'un lancer de pièce est pile ou face) et le résultat du test est représenté par la ou les branches qui suivent.

Chaque résultat peut également être appelé une classe (alias : nœud feuille), résultant d'un test qui le précède ; par exemple, le nœud C2 dans la figure suivante (Nilsson 1998) est un résultat de C1 (Figure 2.3), et il a deux sorties qui représentent la classe G3 et G2.

Si les modèles d'entrée sont testés pour une et une seule caractéristique (comme la couleur), alors ils sont appelés tests univariés ; alors que plusieurs caractéristiques (poids, formes, intensité de la couleur, etc.) signifient que les tests sont multivariés. Ces caractéristiques peuvent être quantitatives ou qualitatives. En outre, selon le nombre de résultats, la complexité de l'arbre et des algorithmes qui l'utilisent peut augmenter, car il peut y avoir aussi peu que deux résultats sans aucune limite supérieure. Si tous les tests n'ont que deux résultats, l'arbre de décision est appelé "arbre binaire" (Nilsson 1998).

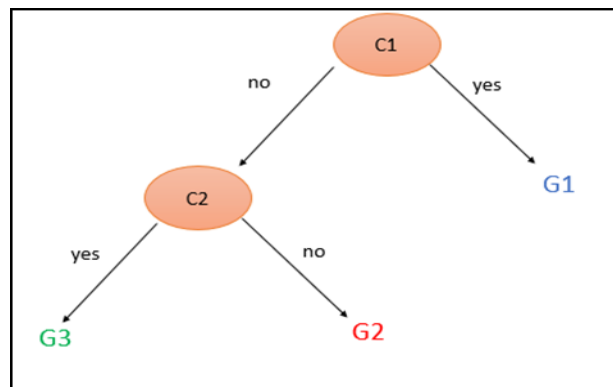


Figure 2. 3: Exemple d'arbre de décision

B. La régression logistique

La régression logistique (Figure 2.4) est l'un des algorithmes d'apprentissage automatique les plus populaires, qui fait partie des techniques d'apprentissage supervisé. Elle est utilisée pour prédire un résultat binaire (catégorique ou discret), c'est-à-dire sous forme d'Oui/Non,

Réussite/Echec, Vie/Décès, etc., en fonction d'un ensemble de données continues ou discrètes (Osisanwo et al. 2017).

La régression logistique est utilisée pour résoudre les problèmes de classification. Elle prédit la sortie d'une variable dépendante, mais au lieu de donner la valeur exacte comme oui ou non, elle donne une probabilité qui se situe entre 0 et 1. Cela s'écrit comme suit :

$$P(Y=1|X) \text{ ou } P(Y=0|X)$$

Y : représente la variable dépendante.

X : représente la variable indépendante.

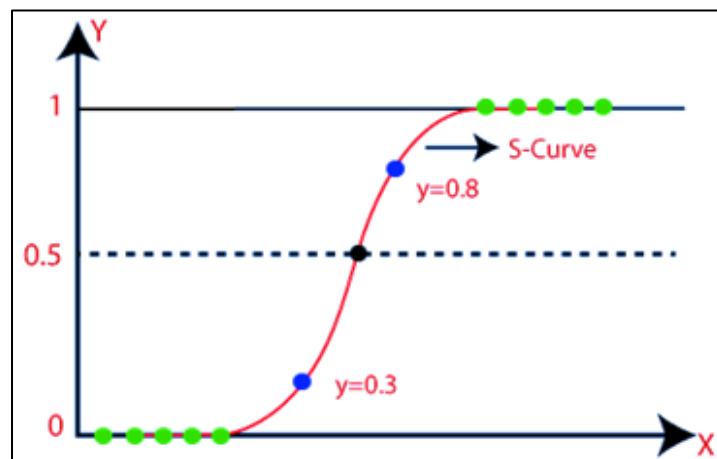


Figure 2. 4: La régression logistique

La courbe (figure 2.4) de la fonction logistique est sous la forme de 'S', qui prédit une sortie dans l'intervalle [0,1].

C. K-nearest neighbors

L'algorithme K-NN est l'un des algorithmes d'apprentissage supervisé. Il traite à la fois les problèmes de régression et de classification, mais il est principalement utilisé pour les problèmes de classification.

L'algorithme K-NN stocke toutes les données de la phase d'apprentissage (formation) afin de trouver les k plus proches dans les données futures. Il représente chaque point de données dans un espace (Figure 2.5) à n dimensions, définies par n caractéristiques. Lorsqu'une nouvelle donnée apparaît, Il calcule la distance entre les points classés et le nouveau point pour placer la donnée dans la catégorie de leur plus proche point voisin (Osisanwo et al. 2017).

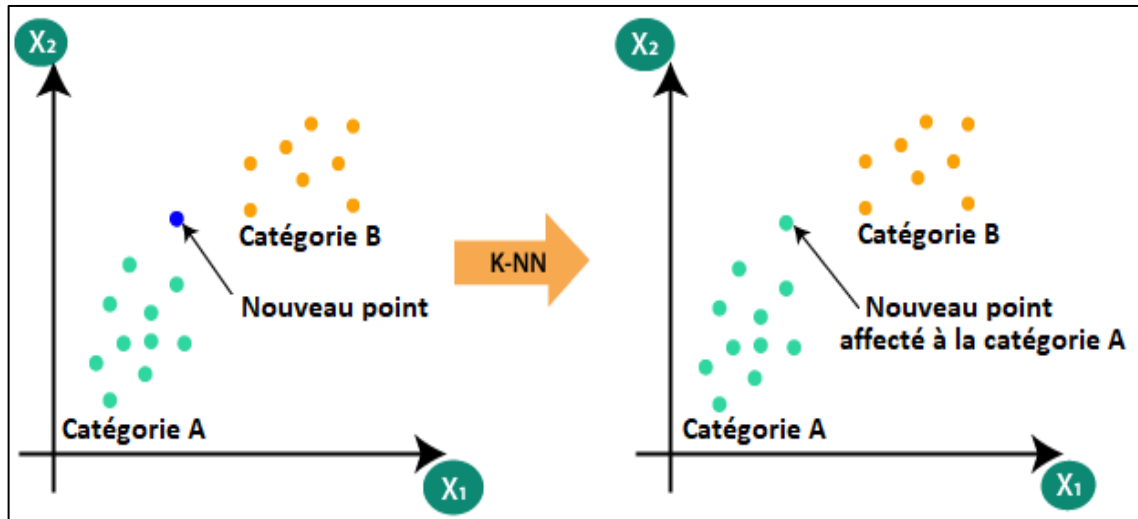


Figure 2. 5: K-nearest neighbors

D. Support Vector Machine

Support Vector Machine (SVM) consiste à classer les données en fonction de leur position dans l'espace par rapport à une ligne ou frontière (Figure 2.6) entre la catégorie positive et la catégorie négative, afin que les nouveaux points de données puissent être facilement placés dans la bonne catégorie à future. Cette frontière de décision est appelée un hyperplan qui maximise la distance entre les points de données de différentes catégories. Le SVM est utilisé pour prédire un résultat binaire (catégorique ou discret), basé sur un ensemble de données continues ou discrètes. Le SVM est destiné aux problèmes de classification.

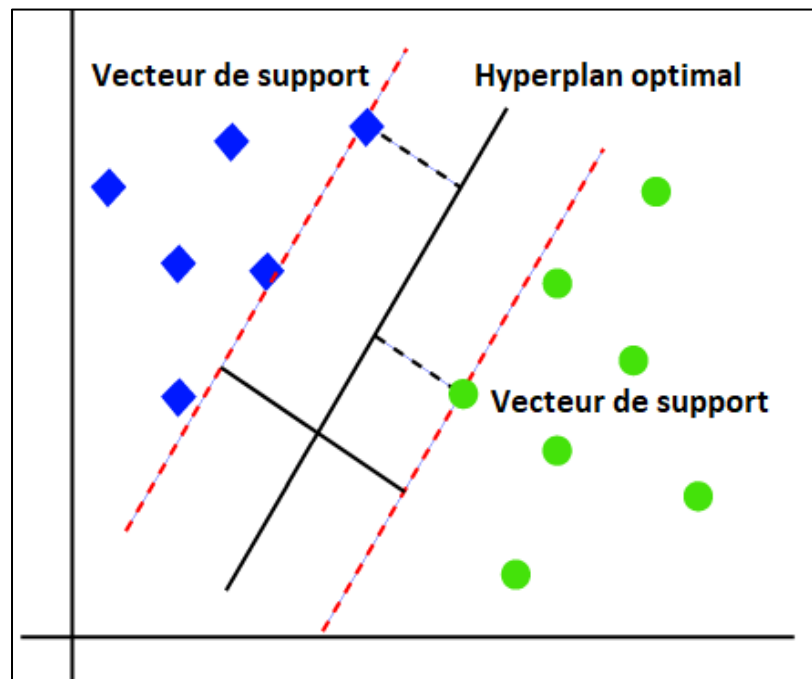


Figure 2. 6: Support Vector Machine (SVM)

E. Naïve Bayes

L'algorithme de Naïve Bayes est un algorithme de classification basé sur le théorème de Bayes, qui calcule la probabilité conditionnelle qu'un point de données appartienne ou non à une certaine catégorie. L'algorithme de Naïve Bayes permet de construire des modèles d'apprentissage automatique rapides, qui peuvent faire des prédictions rapides. Le principal avantage de l'algorithme de Naïve Bayes est que la plupart des algorithmes d'apprentissage automatique reposent sur une grande quantité de données d'apprentissage, mais l'algorithme de Naïve Bayes est relativement performant même si la taille des données d'apprentissage est faible (Osisanwo et al. 2017).

F. La forêt aléatoire

La forêt aléatoire est l'un des algorithmes d'apprentissage supervisé destiné aux problèmes de classification. La forêt aléatoire est une collection d'arbres de décision, chaque arbre dans la forêt est entraîné dans la phase d'apprentissage par un échantillon aléatoire de l'ensemble de données de la formation, puis elle prend le vote majoritaire des arbres.

Tableau 2. 1: résumé des algorithmes de la classification

Algorithme	Problème de base	Prédicteurs	Puissance	Inter prétabilité	Régression aussi
K-NN	Multi classe ou binaire	Numérique	Moyenne	Bonne	Non
La régression logistique	Binaire	Numérique	Faible	Bonne	Non
SVM	Binaire	Numérique ou catégorique	Haute	Moyenne	Non
Naïve Bayes	Multi classe ou binaire	Catégorique	Moyenne	Bonne	Non
Random forest	Multi classe ou binaire	Numérique ou catégorique	Haute	Bonne	Oui
L'arbre de décision	Multi classe ou binaire	Numérique ou catégorique	Haute	Bonne	Oui

2.2.1.2. La régression

Alors que la classification est utilisée pour les données essentiellement discrètes, les données d'entrée qui peuvent être classées comme "continues" sont traitées par l'analyse de régression. Ce type d'analyse est effectivement utilisé pour prévoir les actions et les changements du marché, les ventes et les revenus, etc.

A. La régression linéaire :

Parmi les trois méthodes de régression connues, la régression linéaire est considérée comme la moins complexe, lorsque la variable en question est de nature continue (Royal Society (Great Britain) 2017). Voici quelques points clés concernant ce type de régression : la partie " linéaire " provient de la relation que la variable de décision entretient avec les variables indépendantes ; il existe une règle contraignante selon laquelle les variables indépendantes ne doivent pas être arbitraires et doivent être mesurées de manière irréprochable (aucune tolérance d'erreur) (Sinha 2013) ; la valeur résiduelle (qui est la ligne verticale reliant chaque point à la ligne de régression) doit être constante et ne doit pas être corrélée à travers toutes les observations, et ces valeurs doivent s'identifier comme une distribution normale.

En fonction du nombre de variables indépendantes, nous pouvons diviser la régression linéaire en deux catégories principales, qui sont :

Régression linéaire simple

Lorsqu'une variable est dépendante et une autre indépendante (figure 2.7);

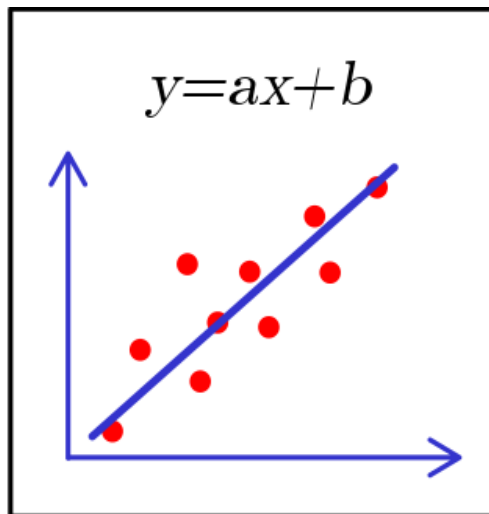


Figure 2. 7 : Régression linéaire simple

Régression linéaire multiple

Lorsqu'une variable est dépendante et qu'il y a plus d'une variable indépendante ;

B. La régression non-linéaire :

Si l'équation n'est pas linéaire, alors la méthode de régression linéaire classique ne permettrait pas de prédire de nouvelles valeurs et estimations ; dans ce cas, les méthodes et modèles de régression polynomiale sont utilisés pour ajuster les courbes de l'équation non linéaire en déterminant les fonctions polynomiales de la variable indépendante, une par une (Sinha 2013).

Les équations dans ce type de régression ont des variables indépendantes à une puissance de 2 ou plus, par exemple $y = a + b \cdot x^2$ est une équation polynomiale, et pour cette raison, elle ne peut pas être représentée par une ligne mais plutôt par une courbe (Figure 2.8) (Sinha 2013).

Lors de l'ajustement d'une équation polynomiale, il est considéré comme une mauvaise pratique d'aller trop loin avec cette méthode. L'objectif est de trouver la meilleure façon d'expliquer le phénomène en question à l'aide de statistiques tout en minimisant les erreurs ; Plus le degré du polynôme est élevé, plus il est difficile de trouver la meilleure courbe pour ajuster son équation, il est donc préférable d'essayer de comprendre la nature du problème avant d'appliquer la méthode de façon arbitraire.

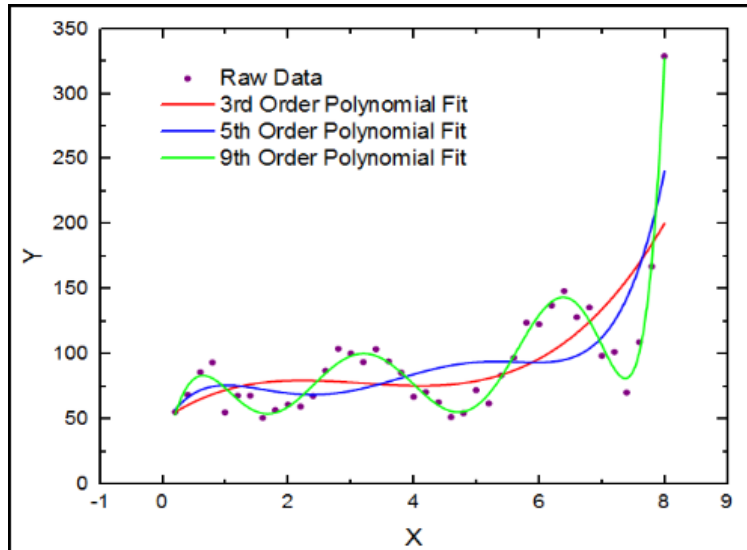


Figure 2. 8: La régression non-linéaire

2.2.2. Apprentissage non supervisé

Contrairement à ceux qui s'appuient sur l'apprentissage supervisé, les algorithmes basés sur l'apprentissage non supervisé utilisent des modèles formés par des ensembles de données contenant des exemples qui ne sont associés à aucune réponse (Figure 2.9) ; cela signifie que, dans ce cas, les systèmes devront apprendre sans s'appuyer sur des labels et déterminer si les différents éléments d'information d'un ensemble de données suivent des schémas spécifiques sans s'appuyer sur un jugement humain (Mueller and Massaron 2016). Un aspect intéressant de ce type d'algorithmes est que, le plus souvent, ils trouvent des liens et des relations entre les ensembles de données fournis qui s'avèrent très utiles pour former de meilleurs modèles pour les algorithmes d'apprentissage automatique non supervisé.

Plutôt que d'apprendre de manière explicite, les systèmes utilisant des algorithmes d'apprentissage non supervisé essaient de "comprendre" les modèles à mesure qu'ils reçoivent de nouvelles données de l'ensemble de données. Pour ce faire, l'algorithme tente de trouver des caractéristiques ou des classes que les points de données peuvent avoir en commun (similarités) en les assignant à des groupes, plutôt que d'analyser chaque point individuellement (Royal Society (Great Britain) 2017).

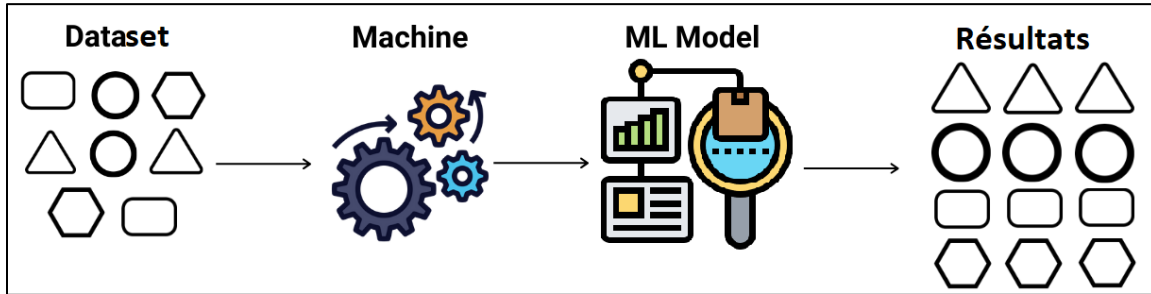


Figure 2. 9: L'apprentissage non supervisé

2.2.3. Apprentissage par renforcement

Ce type d'apprentissage est très similaire à l'apprentissage non supervisé, où l'algorithme reçoit des ensembles de données et des exemples sans label pour déterminer les bons résultats. La différence avec l'apprentissage par renforcement est qu'un retour (Figure 2.10) d'information peut être donné chaque fois que l'algorithme fournit une réponse. L'algorithme apprend par essais et erreurs, et s'efforce d'améliorer la qualité de ses réponses en visant un meilleur retour d'information (Mueller and Massaron 2016) ; c'est très similaire à un être humain qui apprend par expérience. Dans ce cas, l'algorithme essaie de maximiser la fonction objectif et reçoit une pénalité chaque fois qu'il ne parvient pas à donner une bonne réponse (par le biais d'un retour positif/négatif) ; la meilleure analogie est peut-être un jeu où le joueur gagne en obtenant le score le plus élevé : afin d'atteindre ce but, le joueur peut choisir l'approche par essai et erreur pour trouver la meilleure façon de battre le jeu (Royal Society (Great Britain) 2017).

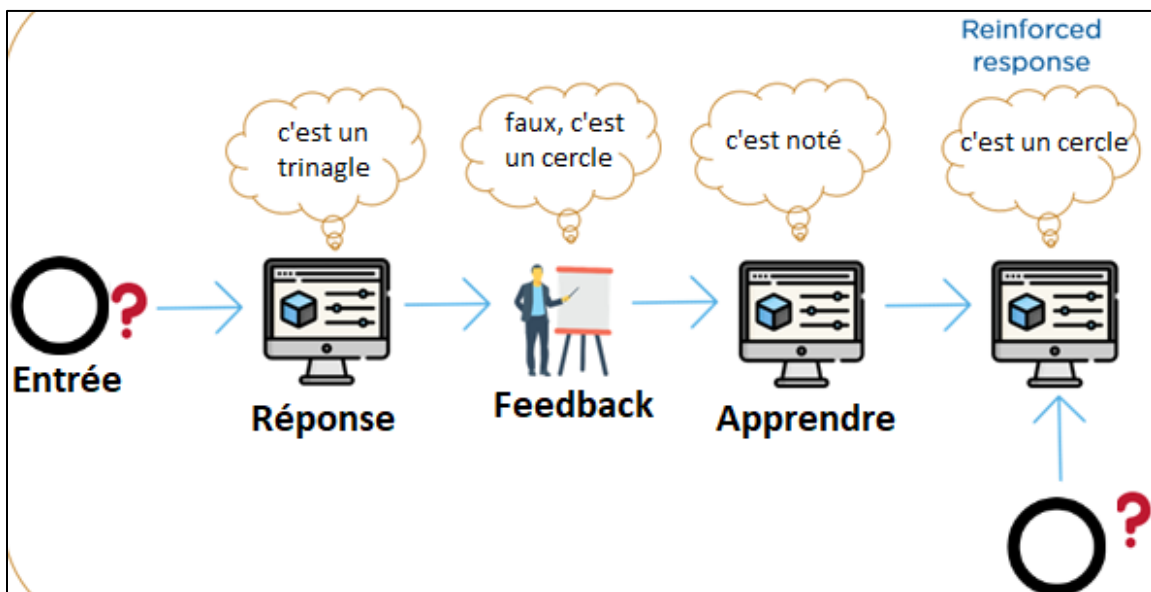


Figure 2. 10 : Apprentissage par renforcement

2.3. AS/RS à convoyeur gravitationnel

Un AS/RS à convoyeur gravitationnel est un AS/RS multi profondeur, c'est-à-dire que chaque casier est utilisé pour le stockage de plus d'une charge et est équipé d'un convoyeur gravitationnel. Il y a plusieurs positions dans un casier, chacune étant dédiée au stockage d'une seule charge. Un AS/RS à convoyeur gravitationnel (Figure 2.11) est composé d'un seul rack compact. Ce dernier a deux côtés : celui de devant est appelé le côté de déstockage et celui de derrière est appelé le côté de stockage. Ce type d'AS/RS contient deux machines S/R. La première opère du côté du stockage et exécute toutes les opérations de stockage ; la seconde machine S/R exécute toutes les opérations de déstockage du côté du déstockage. Les deux machines S/R se déplacent simultanément dans les directions horizontale et verticale, ce qui est connu sous le nom de mouvement de Chebyshev. Les deux côtés du rack sont reliés par un convoyeur de restockage.

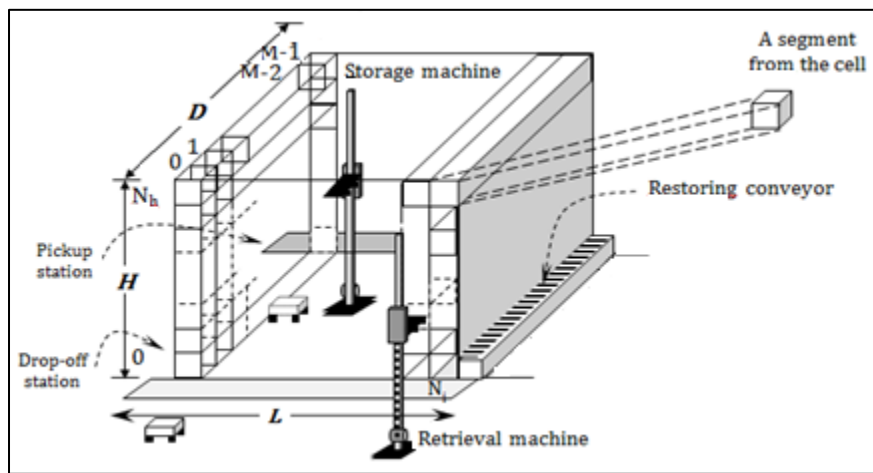


Figure 2. 11: AS/RS à convoyeur gravitationnel

La station d'entrée est située dans le coin inférieur droit du côté stockage, qui est également la position initiale de la machine de stockage. Dans le coin inférieur gauche de ce côté se trouve l'extrémité du convoyeur de restockage. La position de la station de sortie se trouve dans le coin inférieur gauche du côté de déstockage et c'est également la position initiale de la machine de déstockage. Le début du convoyeur de restockage se trouve dans le coin inférieur droit du côté de la sortie.

Comme le montre la Figure 2.11, le rack possède une largeur L , une hauteur H et une profondeur D . Il comporte N_l casiers horizontaux et N_h casiers verticaux. Chaque casier est constitué de M emplacements de stockage (couches) qui sont numérotés de 0 (face de déstockage) à $M-1$ (face de stockage). Les machines de stockage et de déstockage possèdent la même vitesse horizontale V_h , et la même vitesse verticale V_v .

Le temps de déplacement horizontal de la station de dépôt/livraison à la colonne la plus lointaine :

$$t_h = L/V_h \quad (1)$$

L : est la largeur du rack.

Le temps de déplacement vertical de la station de dépôt/livraison à la ligne la plus lointaine :

$$t_v = H/V_v \quad (2)$$

H : est l'hauteur du rack.

Le temps de déplacement horizontal d'un casier au suivant :

$$t'_h = \frac{l}{V_h} = t_h/N_l \quad (3)$$

l : est la distance entre deux casiers adjacents (horizontalement).

Le temps de déplacement vertical d'un casier au suivant :

$$t'_v = \frac{h}{V_v} = t_v/N_h \quad (4)$$

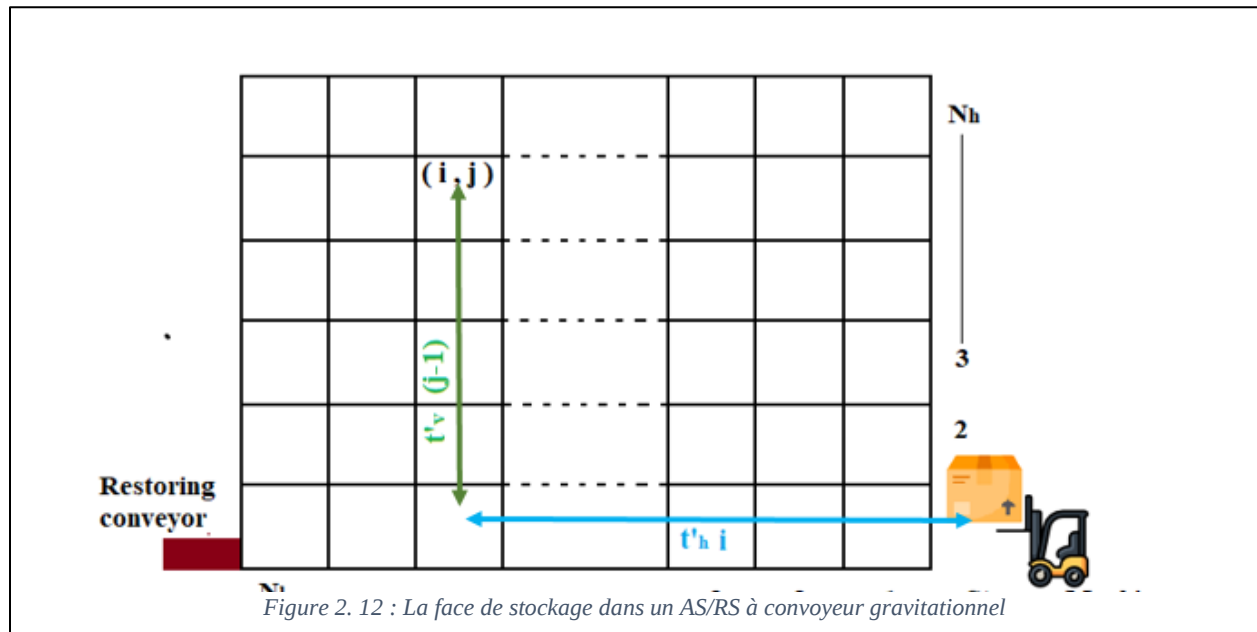
h : est la distance entre deux casiers adjacents (verticalement).

Dans ce qui suit, nous allons présenter de manière formelle les expressions analytiques des temps de stockage et de déstockage

2.3.1. Stockage

Supposons que l'on souhaite stocker une charge dans le casier de coordonnées (i, j). Pour ce faire, la machine de stockage (Figure 2.12) va effectuer un déplacement de sa position de repos (La station d'entrée) vers le casier (i, j) et revenir à sa position initiale, cette opération nécessite une durée :

$$ST = 2 \max(t'_h i, t'_v (j - 1)) \quad (5)$$



2.3.2. Déstockage

Pour déstocker une charge partir de la $k^{ième}$ position (Figure 2.13) du casier de coordonnées (i, j) , la machine de déstockage commence par se déplacer de sa position initiale vers le casier (i, j) , il prend une par une les charges qui précèdent la charge souhaitée dans le casier et les transporte individuellement vers l'extrémité initiale du convoyeur de restockage, il retourne ensuite au casier pour prendre la charge souhaitée et la transporter vers la station de sortie. Le temps nécessaire pour effectuer cette opération est le suivant :

m : est le nombre du charges qui précèdent la charge souhaitée.

$$RT = 2 \max(t'_h i, t'_v(j-1)) + 2(k-1) \max(t'_h ((N_l + 1) - i), t'_v(j-1)) \quad (6)$$

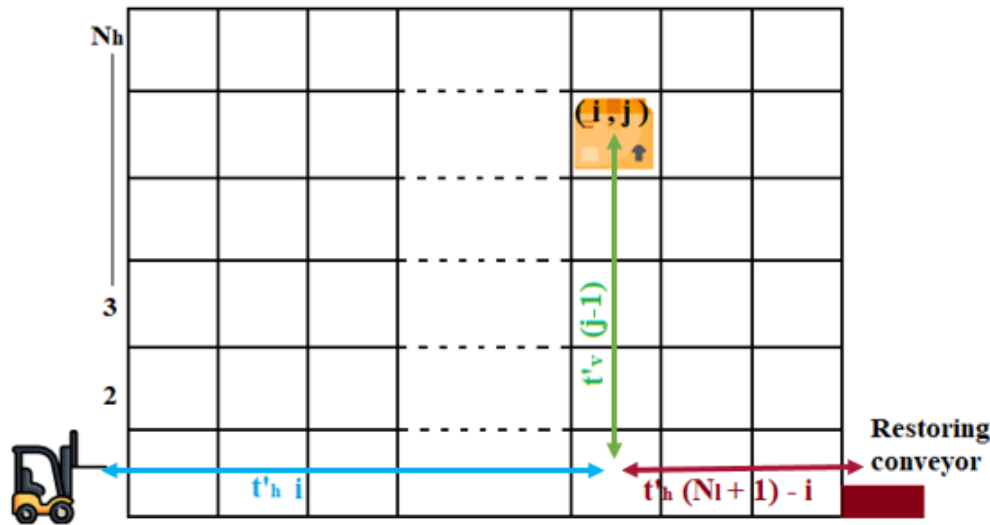


Figure 2. 13: La face de stockage dans un AS/RS à convoyeur gravitationnel

2.4. Mise en œuvre de l'apprentissage machine

L'apprentissage automatique est l'une des principales branches de l'intelligence artificielle (IA) et son principal objectif est de donner la capacité d'"apprendre" aux systèmes informatiques. Il est particulièrement utilisé lorsque les problèmes à résoudre requièrent une solution non conventionnelle, par exemple la planification avancée, la reconnaissance vocale, la prédiction, ... (Nilsson 1998).

Il existe trois principaux types de ML en fonction des algorithmes et des objectifs : l'apprentissage supervisé, l'apprentissage non supervisé et l'apprentissage renforcé. Nous avons choisi d'utiliser les techniques d'apprentissage supervisé en raison de la nature du problème que nous traitons et de ses données, c'est-à-dire que pour obtenir de meilleurs modèles de ML, nous avons fourni au modèle des données labellisées (Mueller and Massaron 2016).

Le problème que nous étudions ici est statique. Il considère en entrée une configuration d'un rack d'AS/RS à convoyeur gravitationnel et une charge à placer dans une des positions vides du rack. Le rack est caractérisé par : ses dimensions, un taux de charge et le nombre de différents types de charges. Le résultat attendu par l'algorithme ML est la position la plus appropriée pour stocker la nouvelle charge de façon à ce que le temps de déstockage soit minimal.

Pour résoudre ce problème, nous avons procédé (Figure 2.14) comme suit :

- La construction de la base des données : Nous avons considéré plusieurs configurations d'AS/RS à convoyeur gravitationnel. Ces configurations se distinguent par : leurs dimensions. Leur taux de charge (taux de remplissage) et le nombre de différents types de charges stockées dans le rack ainsi que leurs proportions. Nous avons utilisé les données de différentes dimensions, contenant 2 à 6 types de charges, le nombre de couches a été fixé à 10, le nombre de casiers horizontalement est {2, 3, 4}, le nombre de casiers verticalement est {2, 3, 4} et le rack a été remplie de manière aléatoire avec un taux de remplissage initial du rack est de 75%.
- Définit les données d'entraînement : Chaque configuration constitue une donnée exploitée dans la phase d'apprentissage. Nous expliquerons plus loin comment nous avons codé ces données.
- Sélection de l'algorithme : Pour ce problème, nous avons besoin d'utiliser deux algorithmes de la régression (la régression linéaire multi variable et la régression polynomiale multi variable) pour estimer le temps de déstockage et un seul algorithme de classification (nous avons appliqué plusieurs algorithmes qui nous donnent les mêmes résultats) qui divise l'ensemble des positions en trois classes : Les positions vides, les positions qui contiennent les charges qui seront directement déstockées dans l'ordre et les positions contenant les charges qui nécessitent deux étapes de déstockage, la première de restockage, la seconde de déstockage..
- Evaluation avec les données de test : les algorithmes entraînés sont évalués en fonction de leur précision dans cette étape. Si la précision est égale à 1 ou proche de 1, l'algorithme est prêt à utiliser, dans le cas contraire, l'algorithme a besoin de plus de données ou d'être modifié.
- La dernière étape est la prédiction qui consiste à considérer chaque position, candidate pour recevoir la nouvelle charge. Et pour chacune de ces positions, estimer le temps nécessaire pour déstockage la charge. L'estimation du temps de déstockage se fait par la régression linéaire et la régression polynomiale.

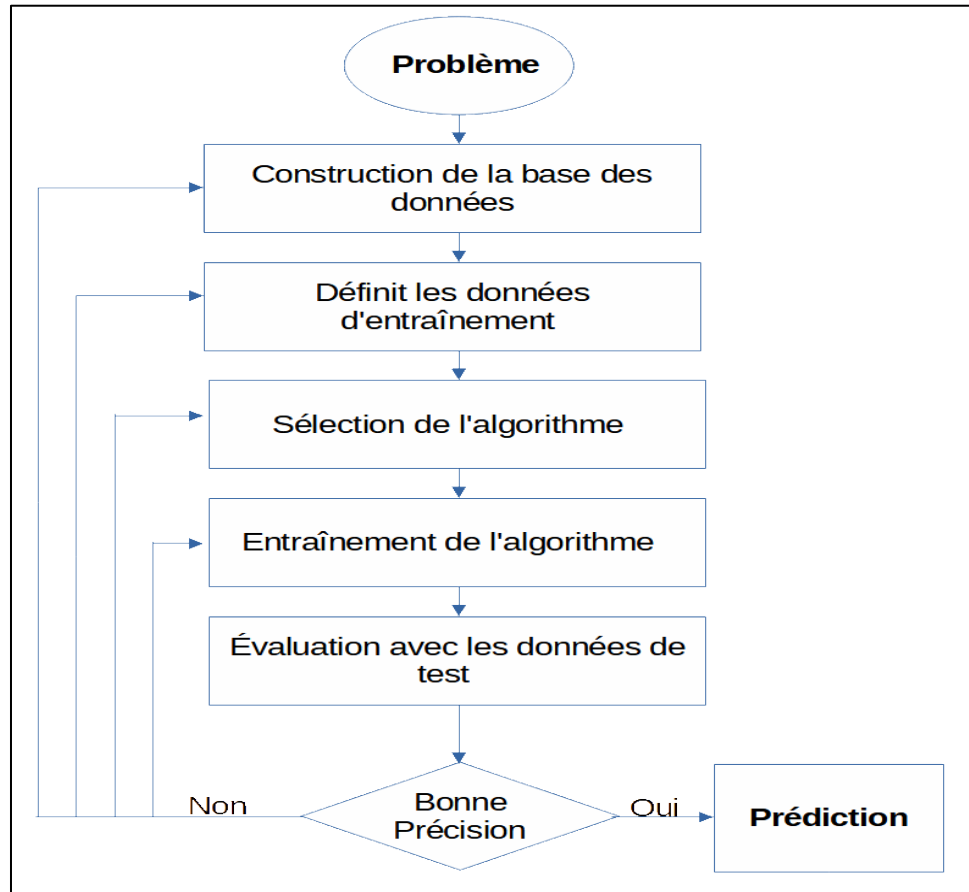


Figure 2. 14: la méthode de résolution

2.4.1. Construction de la base de données

Les données d'entrée du problème traité ici sont des configurations de rack. Chaque donnée est un tableau, avec autant de rangées que la capacité du rack : N_h , N_v et M avec N_h : le nombre de casiers par ligne, N_v : le nombre de casiers par colonne et M : la capacité du casier.

Ainsi, chaque ligne du tableau de données représente un emplacement dans le rack. Cet emplacement a plusieurs propriétés : sa position dans le rack (ligne, colonne et profondeur), si elle est vide ou non, et sinon, le type de charge qu'elle contient et le temps nécessaire pour déstocker la charge qu'elle contient. Sur cette base, le nombre de colonnes de la table de données est de 7, de telle sorte que :

- 1) (Type), représente le type de charge stockée dans l'emplacement, (0) représente un emplacement vide. (1), (2), (3), ... montre que l'emplacement contient une charge de type 1, 2, 3, ... respectivement.
- 2) (x), représente la couche de l'emplacement dans le casier, $1 \leq x \leq M$.
- 3) (y), représente la position du casier contenant l'emplacement horizontalement, $1 \leq y \leq N_h$.

- 4) (z), représente la position du casier contenant l'emplacement verticalement, $1 \leq y \leq N_v$.
- 5) ($Label$), représente le temps nécessaire pour déstocker la charge stockée dans cet emplacement, si elle existe.
- 6) (G), représente le groupe de l'emplacement, nous allons expliquer plus en détail cette notion de groupe dans la suite.
- 7) (nc), représente la position de l'extrémité initiale du convoyeur de restockage dans la face de destockage.

Pendant la phase d'entraînement, nous avons introduit des données de différentes dimensions, contenant 2 à 6 types de charges, le nombre de couches a été fixé à 10, le nombre de casiers horizontalement est $\{2, 3, 4\}$, le nombre de casiers verticalement est $\{2, 3, 4\}$, la vitesse de déplacement horizontal et vertical de la machine S/R est de 1 et le taux de charge initial du rack est de 75%.

Les données (Tableau 2.2) ont été mises dans un tableau Excel, pour être importées dans le programme Jupyter de Python.

Les emplacements candidats pour recevoir la nouvelle charge sont tous vides. Cependant, toutes les emplacements vides ne sont pas candidats. En fait, seules les emplacements, derrière un emplacement plein et dans le début du casier, sont candidats. Si un casier de capacité dix contient déjà cinq charges, alors seule le sixième emplacement peut recevoir immédiatement une nouvelle charge. Parmi tous ces emplacements candidats, il existe plusieurs groupes. Il y a les emplacements qui n'ont pas de charge avant ils dans le casier, c'est-à-dire que le casier est vide. Cela facilite considérablement l'accès à la charge lors du déstockage. Les premières positions des casiers sont donc le groupe le plus favorable pour le stockage de la nouvelle charge.

Tableau 2. 2 : Echantion d'un configuration du rack.

	<i>Type</i>	<i>x</i>	<i>y</i>	<i>z</i>	<i>nc</i>	<i>Label</i>	<i>G</i>
0	1	1	1	1	3	2	2
1	1	2	1	1	3	2	2
2	1	3	1	1	3	2	2
3	2	4	1	1	3	20	2
4	1	5	1	1	3	8	2
5	2	6	1	1	3	26	2
6	2	7	1	1	3	26	2
7	1	8	1	1	3	20	2
8	3	9	1	1	3	20	2
9	0	10	1	1	3	0	1

2.4.1.1. *Python Jupyter :*

L'application open-source Jupyter Notebook permet la création et le partage de code, d'équations, de visualisations et de textes narratifs. Ce logiciel peut être utilisé pour le nettoyage et la transformation des données, la simulation numérique, la modélisation statistique, la visualisation des données, l'apprentissage automatique et bien plus encore (Kluyver et al. 2016).

2.4.1.2. *Calcul du temps de déstockage (label)*

Dans le temps de déstockage, nous ne considérons que le nombre de charges différentes avant la charge actuelle et pour cela nous avons créé un vecteur m qui contient le nombre de charges différentes avant la charge actuelle (la charge que nous voulons déstocker ou stocker), puis nous l'ajoutons à nos données. Nous calculons le temps de déstockage comme suit :

$$\begin{aligned} label = 2 * \max(t_h z, t_v(y - 1)) + 2 * (m - 1) \\ * \max(t_h((nc + 1) - z, t_v(y - 1))) \end{aligned} \quad (7)$$

2.4.2. La phase d'apprentissage

L'apprentissage supervisé comporte deux sous-sujets principaux, qui sont la régression et la classification. Dans cette dernière, nous classons les données sous des catégories [nous aurons au moins deux catégories pour que cela fonctionne (Bishop 2006)] ; les résultats dans ce cas sont discrets (Royal Society (Great Britain) 2017). Quant à la régression, le modèle ajuste la courbe de l'équation (linéaire et non linéaire) afin d'aider à prédire et à estimer de nouvelles valeurs (Sinha 2013) ; les résultats dans ce cas sont continus (Royal Society (Great Britain) 2017). Chacune de ces deux sous-parties possède des algorithmes et des modèles différents :

- 1 La classification: Support vector machine (SVM), decision trees, forest trees, Naïve Bayes, k- nearest -neighbour (KNN), logistic regression, ...
- 2 La regression: Régression linéaire simple, régression linéaire multiple, régression polynomiale, régression polynomiale multiple, ...

Nous avons utilisé les méthodes de classification ainsi que les méthodes de régression pour résoudre notre problème.

2.4.2.1. *L'application des méthodes de classification*

Dans cette partie, le modèle va apprendre à classer notre ensemble de données en 3 groupes :

- Groupe 1 (G1) : contient les emplacements vides.
- Groupe 2 (G2) : contient les charges qui seront directement déstockées dans l'ordre.
- Groupe 3 (G3) : où une opération en deux étapes est effectuée, la première de réstockage, la seconde de déstockage.

Voici comment nous avons obtenu les groupes :

- 1 La matrice des entrées et la sortie (G) : Nous avons sélectionné les entrées (Tableau 2.3) et le vecteur (G) qui représente la sortie de la matrice des entrées afin d'avoir une méthode de classification à partir de nos données.

Tableau 2. 3: Echantion de La matrice des entrées pour la classification

	type	x	y	z	label	M
0	1	1	1	1	1.5	0
1	1	2	1	1	1.5	0
2	1	3	1	1	1.5	0
3	2	4	1	1	13.5	3
4	1	5	1	1	5.5	1
5	2	6	1	1	17.5	4
6	2	7	1	1	17.5	4
7	1	8	1	1	13.5	3
8	1	9	1	1	13.5	3
9	0	10	1	1	0.0	0
10	2	1	2	1	2.5	0

Le code Python ci-dessous nous permet de diviser la matrice en : une matrice d'entrée xd1 et un vecteur de sortie yt1.

```
xd1 = d.drop(['G', 'label'], axis = 1)
```

```
yt1 = d['G']
```

- 2 Pour estimer les données d'entrées (dataset), nous avons divisé nos données en deux parties de manière aléatoire, 80% pour l'entraînement, et les autres 20% pour le test. Pour ce faire, nous utilisons le code Python ci-dessous :

Sklearn : est une bibliothèque Python libre utilisée pour l'apprentissage automatique(ML), qui donne accès à des versions efficaces d'un grand nombre d'algorithmes ML. Elle a de nombreux développeurs, notamment dans le monde académique par des instituts d'enseignement supérieur et de recherche.

```
from sklearn.model_selection import train_test_split  
x_trains,x_test,y_trains,y_test=train_test_split(xd1,yt1,test_size = 0.8 )
```

- 3 Nous avons appliqué plus d'une méthode de classification (SVM, KNN, arbre de décision et Naïve Bayes) afin de créer les classes et avons obtenu les mêmes résultats. Le code suivant est celui de l'arbre de décision, qui donne ses résultats après de nombreuses itérations et observations :

```
clf = tree.DecisionTreeClassifier()
clf = clf.fit(x_trains, y_trains)
```

Pour finaliser cette partie, nous avons divisé nos données à partir de la classification en trois groupes.

```
xd2 = d['G'= 2]
xd2 = xd2.drop(['G', 'label'], axis = 1)
yt2=xd2['label']
xd3 = d['G'= 3]
xd3 = xd3.drop(['G', 'label'], axis = 1)
yt3=xd3['label']
```

2.4.2.2. *L'application des méthodes de régression*

Dans cette partie nous avons utilisé la régression linéaire multi-variable et la régression polynomiale multi-variable.

A. La régression linéaire multi-variable

Dans cette partie, le modèle va apprendre à calculer le temps de déstockage DIRECT du groupe 2 (obtenu à partir des résultats de la classification).

- 1 La matrice des entrées et la sortie (label) du groupe 2 : Nous avons sélectionné les entrées (Tableau 2.4) et le vecteur (label) qui représente la sortie de la matrice des entrées du groupe 2 pour avoir une méthode de calcul à partir de nos données.

Tableau 2. 4: Echantion de la matrice des entrées pour la Régression linéaire multi-variable

	type	x	y	z	M
0	1	1	1	1	0
1	1	2	1	1	0
2	1	3	1	1	0
10	2	1	2	1	0
20	2	1	1	2	0
21	2	2	1	2	0

- 2 Pour estimer les données d'entrées (dataset), nous avons divisé nos données en deux parties de manière aléatoire, 80% pour l'entraînement, et les autres 20% pour le test. Pour ce faire, nous utilisons le code Python ci-dessous :

```
from sklearn.model_selection import train_test_split
x1_trains,x1_test,y1_trains,y1_test=train_test_split(xd2,yt2,test_size = 0.8 )
```

- 3 Régression linéaire multi-variable : Puisque la nature des résultats est continue (Royal Society (Great Britain) 2017), nous avons décidé d'utiliser la Régression pour prédire les temps de déstockage, et sachant que notre modèle utilise plus d'une variable pour la prédiction (x, y, z, M, type), La Régression multi-variable était la meilleure façon de résoudre l'équation (8), et en suivant une méthode étape par étape pour créer un meilleur modèle en commençant par la forme la plus simple de Régression linéaire à des formes plus complexes (non linéaires), nous avons choisi notre modèle final qui est basé sur la Régression linéaire multi-variable. Donnée par :

$$\beta_0 + \beta_1x + \beta_2y + \beta_3z + \beta_4M + \beta_5type = Label \quad (8)$$

Avec $\beta_0, \beta_1, \dots, \beta_5$: représentent les coefficients de corrélation de la Régression linéaire multi-variable.

```
from sklearn import linear_model
from sklearn.metrics import confusion_matrix
regr = linear_model.LinearRegression()
regr.fit(x1_traind,y1_traind)
print('Intercept: \n', regr.intercept_)
print('Coefficients:\n',regr.coef_)E1=regr.predict(d.drop(['label'],axis=1))
```

B. La régression polynomiale multi-variable

Dans cette partie, le modèle apprendra à calculer le temps de déstockage (réstockage et déstockage) à partir des données du groupe 3 obtenues à partir des résultats de la classification.

- 1 La matrice des entrées et la sortie (label) du groupe 3 : Nous avons sélectionné les entrées (Tableau 2.5) et le vecteur (label) qui représente la sortie de la matrice des entrées du groupe 3 pour avoir une méthode de calcul à partir de nos données.

Tableau 2. 5: Echantion La matrice des entrées pour La Régression polynomiale multi-variable

	y	z	label	M
3	1	1	12.0	3
4	1	1	4.0	1
5	1	1	16.0	4
6	1	1	16.0	4
7	1	1	12.0	3
8	1	1	12.0	3
11	2	1	6.0	1
12	2	1	6.0	1

- 2 Pour estimer les données d'entrée (dataset), nous avons divisé nos données en deux parties de manière aléatoire, 80% pour l'entraînement, et les autres 20% pour le test. Pour ce faire, nous utilisons le code Python ci-dessous :

```
from sklearn.model_selection import train_test_split
x2_trains,x2_test,y2_trains,y2_test=train_test_split(xd3,yt3,test_size = 0.8 )
```

- 3 La Régression polynomiale multi-variable : Nous avons utilisé la même méthode que dans la partie précédente pour décider de l'algorithme à utiliser pour cette partie. La seule différence est que la régression linéaire ne nous donne pas de bons résultats, c'est pourquoi nous avons entraîné et testé notre modèle en utilisant la régression polynomiale multi variable (2e degré) l'équations (9) et obtenu de bien meilleurs résultats.

$$\begin{aligned} \beta_0 + \beta_1 x + \beta_2 xy + \beta_3 xz + \beta_4 xM + \beta_5 x^2 + \beta_6 y + \beta_7 yz + \beta_8 yM \\ + \beta_9 y^2 + \beta_{10} z + \beta_{11} xzM + \beta_{12} z^2 + \beta_{13} M + \beta_{14} M^2 \\ = Label \end{aligned} \quad (9)$$

Avec $\beta_0, \beta_1, \dots, \beta_{14}$: représentent les coefficients de corrélation de la Régression polynomiale multi-variable.

```
from sklearn.preprocessing import PolynomialFeatures
poly = PolynomialFeatures(2)
xpoly = poly.fit_transform(x2_trains)
poly.fit(xpoly,y2_trains)
```

2.4.3. La phase de test

Dans cette phase on va évaluer les trois modèles par 20% des données (les données de test).

2.4.3.1. L'arbre de décision

Après avoir entraîné notre modèle (l'arbre de décision), nous avons effectué un test d'évaluation.

```
from sklearn.metrics import accuracy_score
accuracy_score(y_test, clf.predict(x_test))
```

Nous avons obtenu une Précision = 1. (100% correct).

2.4.3.2. La régression linéaire multi variable

Après avoir entraîné notre modèle (régression linéaire multi variable), ce dernier nous a donné les coefficients suivants (Figure 2.15).

```
Intercept:
[-0.5]
Coefficients:
[[7.23131496e-16 1.99840144e-15 1.00000000e+00 1.00000000e+00
 0.00000000e+00]]
```

Figure 2. 15 : Les coefficients de la régression linéaire multi variable

Ce modèle a été évalué avec le code Python ci-dessous:

```
ypred = regr.predict(x1_test)
from sklearn.metrics import r2_score
r2_score(y1_test, ypred)
```

A partir de l'équation suivante :

$$\begin{aligned} R^2 &= 1 - \frac{U}{V} \\ U &= \sum \left((y_{test} - y_{pred})^2 \right) \\ V &= \sum \left((y_{test} - y_{test \cdot \text{mean}()})^2 \right) \end{aligned} \quad (10)$$

R^2 : est l'erreur du modèle divisée par l'erreur d'un modèle de base qui prédit la moyenne de la variable à prédire en permanence. Plus le modèle est performant, plus le score R^2 est élevé, et plus le score R^2 est élevé, plus les prédictions sont précises, jusqu'à un maximum de 100 %. Il n'y a pas de score minimum, mais un modèle simple prédisant toujours la valeur moyenne obtient un score R^2 de 0 %.

Nous avons obtenu une Précision $R^2 = 1$. (100% correct) à partir de l'équation (10).

2.4.3.3. La régression polynomiale multi variable

Après avoir entraîné notre modèle (régression polynomiale multi variable), ce dernier nous a donné les coefficients suivants (Figure 2.16).

```
Intercept:
3.197442310920451e-14
Coefficients:
[ 0.00000000e+00 -1.67366121e-14  1.99493200e-15  4.00000000e+00
 5.32907052e-15 -4.46170878e-15  2.00000000e+00 -9.87751547e-15
-2.00000000e+00 -1.52655666e-16]
```

Figure 2. 16 : Les coefficients de la régression polynomiale multi variable

Ce modèle a été évalué avec le code Python ci-dessous:

```
ypred2 = regr2.predict(poly.fit_transform(x_test))
r2_score(y_test ,ypred2 )
```

Nous avons obtenu une Précision $R^2 = 1$. (100% correct) à partir de l'équation (10).

2.4.4. La méthode de stockage

Dans le but de stocker une charge dans un système AS/RS à convoyeur gravitationnel, le modèle que nous avons entraîné trouvera les emplacements vides dans les casiers par le modèle de classification, puis il calculera le temps de déstockage de chaque emplacement vide par les deux modèles de la régression, et enfin il choisira l'emplacement qui a le temps de déstockage minimum. Notre modèle est capable de trouver l'emplacement dont le temps de déstockage est le plus court, quelle que soit la dimension de l'AS/RS à convoyeur gravitationnel, car nous entraînons le modèle avec différentes dimensions d'AS/RS.

2.5. Resultats

Afin d'évaluer notre modèle, nous l'avons comparé à d'autres méthodes :

- Le stockage aléatoire (R) : dans cette méthode, l'emplacement de stockage est choisi de manière aléatoire.
- Stockage dans l'emplacement minimal (Smin) : Dans cette méthode, nous stockons dans l'emplacement le plus proche de la station de sortie (cet emplacement a le temps de déstockage minimal par rapport aux autres emplacements vides).

Nous avons appliqué la méthode à différents AS/RS à convoyeur gravitationnel (chaque AS/RS a des dimensions différentes que les autres), les positions initiales de la machine S/R et du convoyeur de restockage ne sont pas changées, le nombre d'emplacements (couches) dans le casier est de dix pour tous les AS/RS, le nombre de produits différents dans les AS/RS est de 5, la vitesse de déplacement horizontal et vertical de la machine de Déstockage est de 1 et le taux de

remplissage initial du rack est de 75% (nous avons rempli les AS/RS selon la méthode que nous utiliserons dans la simulation).

Dans la simulation, nous avons appliqué les étapes suivantes :

- 1 Choisir un produit d'un certain type de manière aléatoire.
- 2 Sélectionner le produit le plus proche (qui a le temps de déstockage minimal).
- 3 Re-stocker les produits s'il y a des produits qui précèdent le type que nous voulons déstocker (selon la méthode que nous utilisons).
- 4 déstocker le produit désiré et compter le temps de déstockage.
- 5 Stocker un produit du même type que le produit déstocké (selon la méthode utilisée).

Après un certain nombre de simulations, nous avons obtenu les résultats présentés dans le Tableau 2.6. La première ligne de ce tableau indique les dimensions du différents AS/RS à convoyeur gravitationnel (le premier chiffre présente le nombre du casier horizontalement N_h et le dernier chiffre présente le nombre du casier verticalement N_v) et les autres lignes indiquent les résultats du stockage aléatoire (R), de notre modèle et du stockage à l'emplacement minimal (Smin). Chaque valeur du tableau indique le temps moyen de déstockage d'un produit, en fonction des dimensions du différents AS/RS à convoyeur gravitationnel (colonne), et en fonction de la méthode concernée (ligne). La colonne AVE indique le temps moyen de déstockage pour chaque méthode.

Tableau 2. 6 Le temps de déstockage moyen d'un produit pour chaque méthode

méthode	[5, 5]	[10, 10]	[10, 15]	[15, 10]	AVE
R	5.28	6.32	7.1	6.42	6.277
Le modèle proposé	3.28	3.08	3.22	3.31	3.222
Smin	4.94	4.75	4.82	4.84	4.837

Ainsi que nous pouvons le constater, l'approche proposée donne de très bons résultats (Figure 2.17), non seulement par rapport à la méthode aléatoire (R) mais aussi par rapport à la méthode (Smin) qui sélectionne la position à partir de laquelle le temps de déstockage est minimal. Ceci est dû au fait que notre méthode proposée ne prend pas seulement en considération la distance en temps entre le casier et la station de sortie. Elle prend également en considération si des charges du même type précèdent cette position.

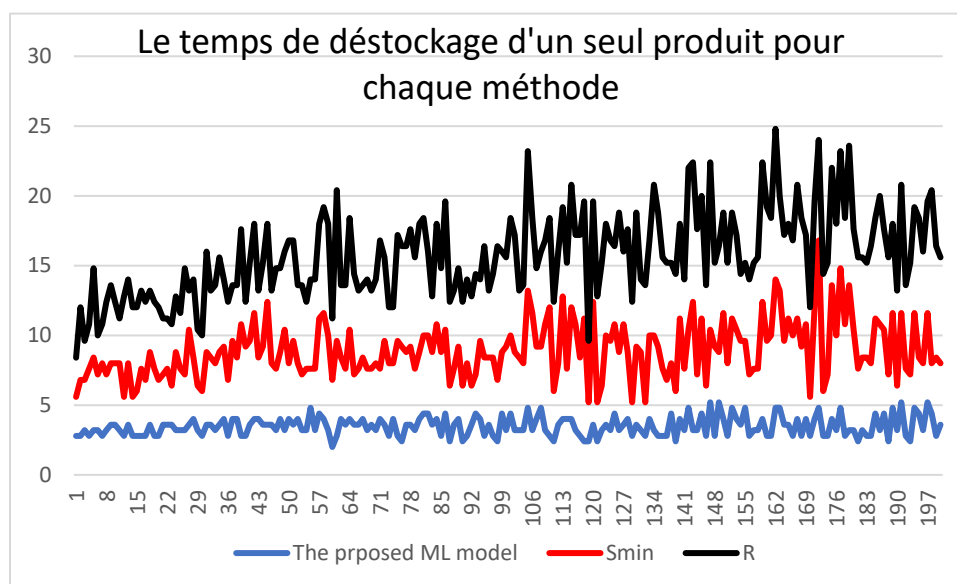


Figure 2. 17 : le temps de déstockage pour chaque méthode

2.6. Conclusion

Dans ce chapitre nous avons d'abord brièvement présenté des notions de base relatives à l'apprentissage automatique. Nous avons ensuite montré comment nous avons adapté des techniques d'apprentissage automatique à la minimisation du temps moyen de déstockage à partir d'un AS/RS profond. Dans le chapitre suivant nous nous intéresserons à la résolution du même problème mais en développant une heuristique dédiée. Nous adapterons aussi deux métaheuristiques à la résolution du même problème.

Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

Chapitre 3

Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

3.1. Introduction

Dans ce chapitre, nous présentons les contributions principales de cette thèse. Il s'agit de l'adaptation de plusieurs techniques de l'intelligence artificielle pour le problème étudié. Rappelons que ce dernier consiste en la minimisation du temps de déstockage d'un lot de produits à partir d'un AS/RS profond. Pour la résolution de ce problème, nous avons utilisé plusieurs techniques de l'intelligence artificielle que nous allons d'abord présenter dans la suite, avant de montrer comment nous les avons adapté au problème et les résultats que nous avons obtenus.

3.2. Les Métaheuristiques:

Les métaheuristiques sont des algorithmes d'optimisation visant à résoudre des problèmes complexes et/ou de grandes tailles, dans un temps raisonnable. Le terme métaheuristique a été introduit pour la première fois par (Glover 1989) qui définit comme une stratégie de recherche itérative qui explore l'espace de recherche de manière efficace afin de déterminer des solutions optimales, et peut être utilisée de manière générale, sans faire appel à un problème spécifique. Les algorithmes de la métaheuristique comprennent la recherche locale, les algorithmes génétiques (GA), le recuit simulé et l'analyse de données, la colonie de fourmis (ACO). La métaheuristique est composée de deux types des algorithmes, les algorithmes basés sur une solution ou une population (plusieurs solutions comme GA).

3.2.1. Recuit simulé

Le recuit simulé (SA) est l'un des algorithmes métaheuristiques les plus anciens, qui a été introduit par (Kirkpatrick, Gelatt Jr, and Vecchi 1983) et (Černý 1985), pour la recherche de problèmes d'optimisation combinatoire. La méthode du recuit simulé s'inspire de la thermodynamique d'un système de particules. Le recuit est une procédure utilisée en métallurgie pour obtenir un alliage sans défaut.

Le SA est l'un des premiers algorithmes à disposer d'une stratégie permettant d'explorer l'espace de recherche. L'idée de base est d'autoriser les mouvements qui conduisent à une solution de moindre qualité que la meilleure solution actuelle. Ces mouvements sont utilisés pour éviter d'être piégé dans des optimums locaux. La température initiale et le taux de refroidissement sont des critères essentiels dans cet algorithme car la température intervient dans la probabilité d'accepter des voisins de moindre qualité que la solution courante.

Le recuit simulé commence par générer une solution initiale, de manière aléatoire ou heuristique, et par initialiser la température initiale T , la vitesse de refroidissement et le critère d'arrêt.

Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

L'algorithme va alors répéter de manière continue ce qui suit jusqu'à ce que le critère d'arrêt soit atteint :

Une solution s' du voisinage $N(s)$ de la solution s est choisie aléatoirement.

- Si la solution s' du voisinage diminue la valeur de la fonction objectif, le SA va l'accepter toujours $s = s'$.
- Si la solution s' du voisinage augmente la valeur de la fonction objectif $f(s') > f(s)$, l'SA va examiner la nouvelle solution s' par la probabilité d'acceptation, Généralement cette probabilité suit la probabilité de Boltzmann :

$$e^{(f(s_i) - f(s_j))/T_k} \quad (11)$$

Pendant toute la durée d'exécution de l'algorithme, la température initiale T est réduite à chaque étape en fonction de la vitesse de refroidissement jusqu'à atteindre le critère d'arrêt.

L'algorithme de recuit simulé est présenté ci-dessus :

1. *Select an initial state $s_i \in S$;*
2. *Select an initial temperature $T > 0$;*
3. *Select an sample size ;*
4. *Repeat*
5. *Generate neighbors of s_i ;*
6. *Repeat*
7. *Select state s_j , a neighbor of s_i ;*
8. *Calculate $f(s_i)$ and $f(s_j)$;*
9. *If $f(s_j) \leq f(s_i)$ then;*
10. *Set $s_j = s_i$;*
11. *Else if $e^{(f(s_i) - f(s_j))/T_k} \leq \text{random}(0,1)$ then;*
12. *$s_i = s_j$;*
13. *Update T_k*

3.2.1.1. Avantages

- le SA converge vers la solution optimale d'un problème (lorsque le nombre d'itérations tend vers l'infini).
- Une mise en œuvre facile de l'algorithme.
- Donne souvent de bonnes solutions si on le compare aux algorithmes de recherche classiques.
- utilisé généralement dans la plupart des problèmes d'optimisation.

3.2.1.2. Inconvénient

- bien qu'il soit prouvé qu'il converge vers l'optimum, il converge en un temps infini.
- La détermination de la température initiale est difficile.
- L'impossibilité de savoir que la solution trouvée est optimale.

3.2.2. Recherche taboue

La recherche Tabou (TS) est un des algorithmes métaheuristiques les plus couramment utilisées, qui a été introduite par Glover (1989), pour résoudre divers problèmes d'optimisation combinatoire.

Le concept principal de la TS consiste à interdire la recherche de se déplacer vers des solutions déjà visités dans l'espace de recherche, au moins pour les quelques prochaines itérations. Ces mouvements "tabous" permettent d'éviter d'être piégé dans des optimums locaux en autorisant l'algorithme à accepter de manière temporaire de nouvelles solutions de moindre qualité que la meilleure solution actuelle afin d'éviter les chemins déjà explorés.

TS utilise une mémoire à court terme, appelée liste tabou, qui garde la trace des solutions les plus dernièrement visitées et interdit tout mouvement vers ces solution, ce qui permet d'éviter d'être piégé dans des optimums locaux. Le voisinage de la solution courante dans TS est limité à la solution qui n'appartient pas à la liste tabou. Ces solutions sont classées comme des ensembles autorisés. A chaque itération, la meilleure solution est choisie comme nouvelle solution. Cette solution est ajoutée à la liste tabou et la plus ancienne solution de la liste est supprimée en utilisant La technique FIFO (First In First Out). L'algorithme va répéter de manière continue jusqu'à ce que les ensembles autorisés soient vides ou si la condition d'arrêt soit atteinte. La longueur de la liste de tabou qui contrôle la mémoire du processus de recherche. Une grande longueur de la liste de tabou forcera la recherche sur de grandes régions, Au contraire, une petite longueur de la liste tabou forcera la recherche sur de petites régions.

L'algorithme de recherche tabou est présenté ci-dessus :

1. *Select an initial state $si \in S$;*
2. *Initialize the tabu lists L ;*
3. *Repeat*
4. *Generate a neighbors of $N(si, k)$;*
5. *Choose the best of them $bs \in S$;*
6. *If $f(bs) \leq f(si)$ then*
7. *$si = bs$*
8. *Update L_k*

3.2.2.1. Avantages

- Offre des économies de temps de résolution pour des programmes de grosse taille.
- Une mise en œuvre facile de l'algorithme.
- Donne souvent de bonnes solutions si on le compare aux algorithmes de recherche classiques
- utilisé généralement dans la plupart des problèmes d'optimisation.

3.2.2.2. Inconvénients

- L'impossibilité de savoir que la solution trouvée est optimale.
- Nécessite des ressources importantes si la liste des tabous est trop longue.

3.3. La méthode NN

L'heuristique du plus proche (NN) est présentée pour la première fois par Han et al. (1987) dans le but de trouver le séquençement des produits dans la requête de stockage ou de déstockage en fonction de leurs distances par rapport à la station d'entrée/sortie (c'est-à-dire en fonction du temps de déplacement de la machine de stockage/ déstockage à partir de la station d'entrée/sortie jusqu'à l'emplacement souhaité). L'heuristique du plus proche consiste à choisir à chaque fois le produit qui a un temps de stockage/ déstockage minimal dans le lot et de le stocker/déstocker et ainsi de suite pour le reste des produits dans la requête. NN est l'une des heuristiques les plus connues, et elle est utilisée dans plusieurs travaux (par exemple, ((Gue and Kim, 2007) ; (Yu and De Koster 2012))).

3.4. Description du problème

Dans ce travail, nous nous intéressons à l'opération de déstockage dans le même système de chapitre précédent qui est l'AS/RS à convoyeur gravitationnel. Nous détaillerons dans la suite les déplacements élémentaires de la machine de déstockage lorsqu'il effectue une opération de déstockage à partir de la $k^{\text{ème}}$ position du casier à l'intersection de la $i^{\text{ème}}$ ligne et de la $j^{\text{ème}}$ colonne. Nous allons nommer un casier par ses numéros de ligne et de colonne, en partant du casier le plus proche de la station de sortie. Ainsi, le casier de coordonnée (i, j) se trouve à l'intersection de la ligne i et de la colonne j dans le rack. Le casier de coordonnée (1, 1) est le casier adjacent à la station de sortie. De plus, les coordonnées (i, j, k) vont labelliser le $k^{\text{ème}}$ emplacement du casier (i, j). Avec $k = 1$ pour les emplacements qui sont sur la couche la plus proche de la face de déstockage. Le déstockage d'une charge de l'emplacement (i, j, k) implique deux types de déplacements élémentaires :

- Le déplacement entre la station de la machine de déstockage et le casier (i,j). le temps de ce déplacement est le même à l'aller et au retour. Elle est formulée comme suit :

$$T_1: (i, j) \rightarrow T_1(i, j) = \max\{jt_h; (i - 1)t_v\}. \quad (12)$$

- Le déplacement entre le casier (i, j) et l'extrémité initiale du convoyeur de restockage. Ici aussi, le temps de déplacement est le même à l'aller et au retour et est formulée comme suit :

$$T_2: (i, j) \rightarrow T_2(i, j) = \max\{(N_h + 1 - j)t_h; (i - 1)t_v\}. \quad (13)$$

Au cours d'une opération de déstockage à partir de la position (i, j, k), la machine de déstockage effectue un déplacement de son station de repos vers le casier (i, j), 2 (k - 1) déplacements entre le casier (i, j) et le convoyeur de restockage et un dernier déplacement du casier

Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

(i, j) et de sa station de repos. Ainsi, le temps de déplacement de déstockage T depuis la position (i, j, k) $\in \{1; \dots; N_h\} \times \{1; \dots; N_v\} \times \{1; \dots; M\}$ est formulé comme suit :

$$T(i, j, k) = 2T_1(i, j) + 2(k - 1)T_2(i, j). \quad (14)$$

Notre objectif est de minimiser le temps de déstockage, donné par (14). Pour cela, nous commençons par expérimenter de façon formelle le problème étudié.

Nous caractérisons l'AS/RS à convoyeur gravitationnel par ses dimensions : N_h , N_v et M et nous modélisons son état à un instant donné par une matrice S de N lignes et 4 colonnes (rappelons que N est le nombre total d'emplacement dans le rack, $N = N_h * N_v * M$). Chaque ligne de la matrice S, représente un emplacement dans le rack et son état. Elle contient 4 informations : la ligne i, la colonne j, la profondeur k, et l'état de l'emplacement α ; avec α est égal à 0 lorsque l'emplacement est vide et le type de la charge qu'elle contient lorsque l'emplacement n'est pas vide.

Exemple 1. La matrice S ci-dessous représente l'état du rack de dimensions $N_h * N_v * M$ (égal à $2*2*3$). Il y a deux types de charges dans ce système : les charges de type 1 sont dans les emplacements (1,1,1), (1,2,1), (2,1,1), (2,1,2), (2,1,3), les charges de type 2 sont dans les emplacements (1,1,2), (1,1,3), (1,2,2), (2,2,1), et les emplacements (1,2,3), (2,2,2), (2,2,3) sont vides.

$$S = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 2 & 2 \\ 1 & 1 & 3 & 2 \\ 1 & 2 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 0 \\ 2 & 1 & 1 & 1 \\ 2 & 1 & 2 & 1 \\ 2 & 1 & 3 & 1 \\ 2 & 2 & 1 & 2 \\ 2 & 2 & 2 & 0 \\ 2 & 2 & 3 & 0 \end{pmatrix} \begin{pmatrix} 2t_h & = & 2 \\ 6t_h & = & 6 \\ 10t_h & = & 10 \\ 4t_h & = & 4 \\ 6t_h & = & 6 \\ 8t_h & = & 8 \\ 2 \max\{t_h; t_v\} & = & 2 \\ 2 (\max\{t_h; t_v\} + \max\{2 t_h; t_v\}) & = & 6 \\ 2 (\max\{t_h; t_v\} + 2 \max\{2 t_h; t_v\}) & = & 10 \\ 2 \max\{2 t_h; t_v\} & = & 4 \\ 2 (\max\{2 t_h; t_v\} + \max\{t_h; t_v\}) & = & 6 \\ 2(2 (\max\{2 t_h; t_v\} + \max\{t_h; t_v\})) & = & 10 \end{pmatrix}$$

Dans cet exemple, nous pouvons calculer pour chaque ligne de la matrice S, le temps nécessaire pour déstocker la charge contenue dans l'emplacement correspondant (i, j, k). Ces valeurs du temps de déstockage T(i, j, k) sont représentées dans la droite de la matrice S ci-dessus. En considérant les valeurs numériques des déplacements élémentaires $t_h = t_v = 1$.

Notons que le temps de déstockage de l'emplacement (i, j, k) ne dépend pas du paramètre α .

Dans le paragraphe suivant, nous allons étudier le problème de la localisation d'un lot de charges et le séquençement des opérations de déstockage de ces charges (les charges du lot), de telle sorte que le temps de déstockage du lot soit minimisé.

3.5. Modélisation mathématique et résolution du problème d'optimisation étudié

Comme mentionné précédemment, le problème qui nous intéresse est la minimisation du temps de déstockage. Plus précisément, nous sommes intéressés par la minimisation du temps de déstockage d'un lot de charges. En effet, supposons que nous voulons déstocker un lot de q charges du rack. Ce lot contient q_l charges de type l ($l = \overline{1, L}$). L étant le nombre de types différents de charges stockées dans le rack et $Q = \sum_l^L q_l$. De plus, nous mettons

$$q_l = \begin{pmatrix} q_1 \\ q_2 \\ \vdots \\ q_L \end{pmatrix}$$

Nous cherchons ici à exprimer le problème de localisation des charges Q dans le rack et à planifier leur déstockage de telle sorte que le temps de déstockage global du lot soit minimisé.

L'algorithme le plus basique pour résoudre ce problème est l'algorithme NN (nearest load), qui consiste à sélectionner pour le déstockage des charges dont le temps de déstockage est minimal dans l'état actuel du rack. Cependant, l'idée de l'algorithme NN est très simple ; son application n'est pas très efficace car les charges sont déstockées une par une et l'état du rack est modifié après chaque opération de déstockage, ce qui permet de trouver les casiers dans lesquelles le déstockage est plus intéressant.

3.5.1. Modélisation mathématique

Une formulation mathématique du problème étudié est présentée ci-dessous :

$$\min 2 \sum_{i=1}^{N_h} \sum_{j=1}^{N_v} X_{i,j,1} T_1(i, j) + \sum_{i=1}^{N_h} \sum_{j=1}^{N_v} \sum_{k=2}^M X_{i,j,k} (2T_1(i, j) + 2T_2(i, j)(k-1 - \max_{1 \leq s \leq k-1} \{0 ; s * X_{i,j,s}\})) \quad (16)$$

Sous contraintes :

$$\sum_{i=1}^{N_h} \sum_{j=1}^{N_v} \sum_{k=1}^M Y_{i,j,k,l} X_{i,j,k} = q_l, \quad \forall l \in \{1, \dots, L\}, \quad (17)$$

avec X est une variable de décision binaire définie par :

$$X_{i,j,k} = \begin{cases} 1 & \text{si la charge dans la position } i, j, k \text{ est sélectionnée pour être déstocker,} \\ 0 & \text{sinon,} \end{cases} \quad (18)$$

et Y est un paramètre binaire qui définit l'état du rack, tel que :

$$Y_{i,j,k,l} = \begin{cases} 1 & \text{si la charge de type } l \text{ est stockée dans la position } i, j, k, \\ 0 & \text{sinon} \end{cases} \quad (19)$$

Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

La résolution de ce modèle est possible pour de petites instances. Pour ce faire, nous avons utilisé le logiciel Gurobi.

Le solveur Gurobi est une bibliothèque logicielle d'optimisation mathématique permettant de résoudre des problèmes d'optimisation linéaire et quadratique en nombres entiers mixtes dans python.

Les objets Gurobi les plus couramment utilisés sont les modèles. Ce dernier est composé d'une fonction objectif (linéaire ou quadratique) sur ces variables en utilisant la commande "Model.setObjective", d'un ensemble de variables de décision en utilisant les commandes "Var" ou "MVar", et de contraintes sur ces variables en utilisant les commandes "Constr", "MConstr", "GenConstr", "SOS", "QConstr", "MQConstr", ou "MGenConstr". Les variables sont associées à une borne inférieure, une borne supérieure et un type (continu, binaire, etc.). Les contraintes (inertielles ou quadratiques) sont associées à une direction (inférieure ou égale à, supérieure ou égale à, ou égale à) et à une valeur de droite (Karbstein, 2022).

Par exemple, nous considérons la configuration présentée dans l'exemple 1, à partir de laquelle nous considérons le déstockage du lot contenant une charge de type 1 et une charge de type 2, c'est-à-dire $q = (1,2)^T$. La solution obtenue par le logiciel Gurobi est : $X_{1,1,1} = X_{1,1,2} = 1$ et $X_{i,j,k} = 0$ pour tous les $(i,j,k) \in \{1; \dots; N_h\} \times \{1; \dots; N_v\} \times \{1; \dots; M\} \setminus \{(1,1,1); (1,1,2)\}$, cette solution est obtenue en quelques millisecondes (Figure 3.1).

```
Optimal solution found (tolerance 1.00e-04)
Best objective 2.600000000000e+01, best bound 2.600000000000e+01, gap 0.0000%
Variable          x
-----
y[0,0,0]          1
y[0,0,1]          1
```

Figure 3. 1: La solution obtenue par le solveur gurobi

Les solveurs tels que Gurobi (Jablonský 2015) (Anand, Aggarwal, and Kumar 2017) donnent une solution optimale au problème étudié pour de petites instances telles que l'exemple considéré, malheureusement les limites de cette méthode sont vite atteintes d'où la nécessité d'utiliser des méthodes moins formelles telles que les heuristiques et métaheuristiques. Nous proposons dans ce qui suit un algorithme pour résoudre ce problème étudié.

Si le lot à déstocker est constitué de charges qui se trouvent chacune dans un casier différent, alors l'ordre dans lequel les opérations de déstockage sont effectuées ne change pas le temps de déstockage du lot et le choix de l'emplacement des charges composant le lot dans le rack selon la méthode NN garantit un temps de déstockage optimal. Cependant, si certaines des charges qui composent le lot sont situées dans le même casier, il est évident que l'algorithme NN n'est plus optimal. C'est pour cette situation que nous proposons une heuristique. L'idée principale de l'heuristique est d'identifier les groupes de charges de type demandé dans le lot, et les charges de groupe sont trouvées dans le même casier. Ensuite, parmi les groupes détectés, on sélectionne ceux qui améliorent le temps de déstockage.

3.6. Proposition d'une heuristique de résolution

L'algorithme a comme entrée les données suivantes :

P est un vecteur de dimension q qui contient la position du type demandé qui a le temps de déstockage minimal ;

MR est une matrice de dimension $N_h \times N_h \times M$. L'élément $MR(i, j, k)$ est égal au type de la charge stockée dans la position (i, j, k) et est égal à zéro si la position (i, j, k) est vide ;

Lot est un vecteur de dimension q où chaque élément représente un type de produit à déstocker ;

L'algorithme proposé comporte trois étapes. La première étape consiste à calculer la matrice MG qui est la matrice de séquence. L'élément $MG(i, j, k)$ est l'ordre de déstockage des produits trouvés dans la requête de déstockage Lot et dans le même casier. La deuxième étape consiste à calculer les matrices MTG et MTG_{min} qui contiennent respectivement les temps de déstockage des groupes et les temps de déstockage des groupes utilisant les temps des produits les plus proches P_r . L'élément $MTG(i, j, k)$ est le temps de déstockage du groupe des coordonnées (i, j, k) et l'élément $MTG_{min}(i, j, k)$ le temps de déstockage du groupe (i, j, k) en utilisant les temps de déstockage des produits les plus proches P_r . Et finalement la troisième étape qui a un but de comparer les résultats des matrices MTG et MTG_{min} , et trouver les meilleurs groupes bg_t .

Nous avons résumé l'heuristique proposée dans l'organigramme (Figure 3.2) suivant :

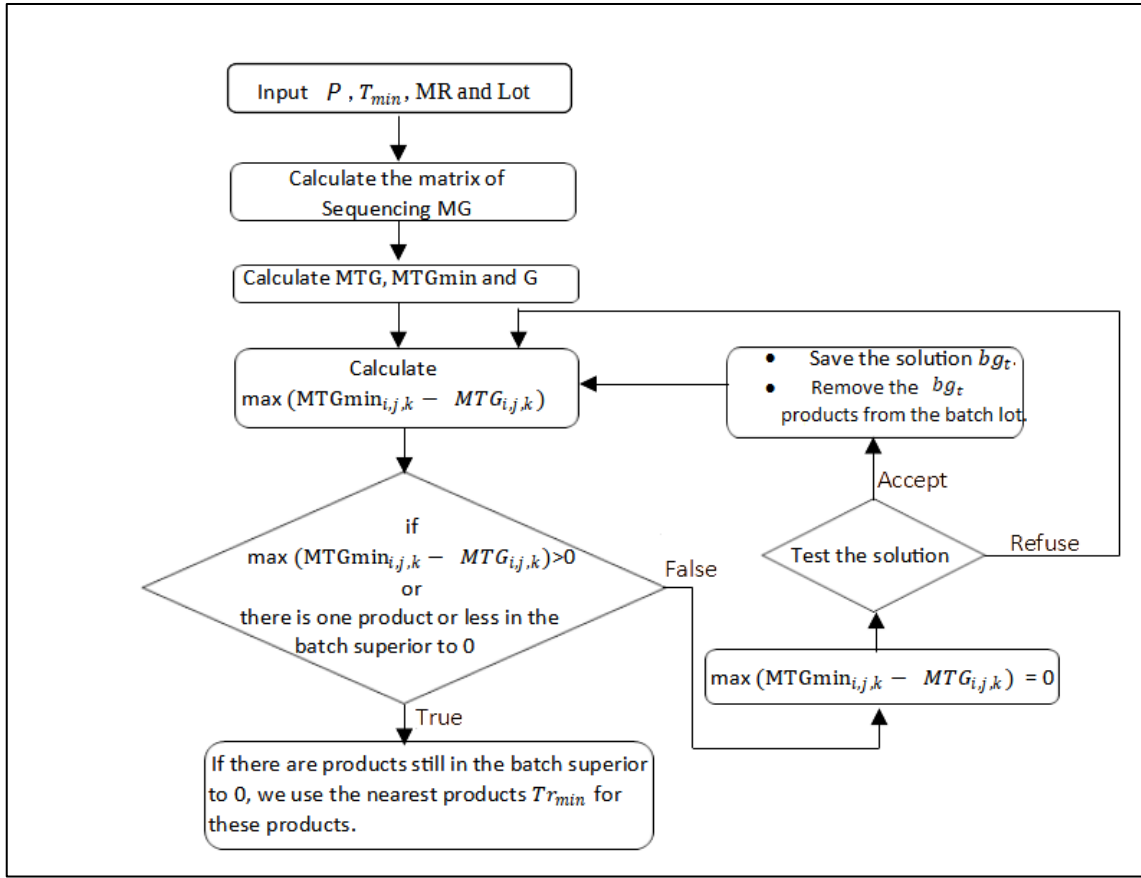


Figure 3. 2 : L'Organigramme de l'heuristique proposée

Algorithm 1 Part 1

```

Input: P, Tr, MR, Lot
1. for i ← 1, Nv do
2.   for j ← 1, Nh do
3.     Z ← 0
4.     for k ← 1, M do
5.       for r ← 1, q do
6.         if MR(i, j, k) = Lot(r) then
7.           Z ← Z + 1  Z est l'ordre des charges trouvées dans la requête de
recherche (lot) et dans le même casier.
8.           MG(i, j, k) = Z
9.         else
10.          MG(i, j, k) = 0
11.        end if
12.      end for
13.    end for
    
```

Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

```

14.   end for
15. end for

```

La première étape de notre méthode est de créer une matrice MG qui a les mêmes dimensions du rack, chaque ligne dans cette matrice représente le séquençement des produits trouvés dans la requête de déstockage (lot) et dans le même casier, chaque élément dans la matrice représente l'ordre de déstockage des produits trouvés dans la requête de déstockage (lot) et dans le même casier. Concernant l'ordre :

- sera supérieur à 0 (>0) : dans le cas où ce produit se trouve dans la requête de déstockage (lot) et dans le même casier (NB : le choix des emplacements sera toujours aux produits qui ont un temps de déstockage minimum par rapport aux produits de même type dans le casier).
- égale à 0 ($=0$) : lorsque ce produit appartient de type non demandé dans le lot, ou ce produit appartient d'un des types choisis déjà en respectant la quantité des produits de même type dans le lot (exemple : si on a 3 produits de même type dans un lot, et on a 4 produits de ce dernier type dans un casier, les 3 premiers produits seront séquençés et le 4ème sera égal à zéro)

Grâce à MG on peut trouver tous les groupes g_s ($g_s \in G$), les emplacements choisis dans les groupes doivent être dans le même casier et avec le même ordre de stockage dans le casier, la taille minimale du groupe g_s est égale à 2.

4	1	5	3	2	6	4
6	3	6	2	1	0	0
1	6	5	4	3	5	0
2	2	5	2	0	0	0

Figure 3. 3: AS/RS contient 4 casiers

La Figure 3.3 présente un rack contenant 4 casiers ($J = 4$) l'un sur l'autre, chaque casier contient 7 positions ($K=7$), si nous voulons déstocker un $lot = [1 \ 2 \ 1 \ 5]$, dans ce rack la matrice possède deux dimensions car $i=1$ (le nombre de casier verticalement égale 1), donc la matrice MR égale :

$$MR = \begin{pmatrix} 2 & 2 & 5 & 2 & 0 & 0 & 0 \\ 1 & 6 & 5 & 4 & 3 & 5 & 0 \\ 6 & 3 & 6 & 2 & 1 & 0 & 0 \\ 4 & 1 & 5 & 3 & 2 & 6 & 4 \end{pmatrix}$$

La matrice MG égale :

$$MG = \begin{pmatrix} 1 & 0 & 2 & 0 & 0 & 0 & 0 \\ 1 & 0 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 & 3 & 0 & 0 \end{pmatrix}$$

On a trouvé les groupes suivants $G = [g_0, g_1, g_2, g_3, g_4]$, nous avons 5 groupes, un groupe pour chaque casiers 1, 2 et 3 sauf les deux derniers groupes (g_3 et g_4) se trouve dans le quatrième casier, le 4eme groupe (g_3) contient les emplacements des produits qui ont les ordres 1 et 2 dans la matrice MG , le 5eme groupe (g_4) contient les emplacements des produits qui ont les ordres 1, 2 et 3 dans la matrice.

Algorithm 1 Part 2

```

16.  s = 0
17.  for i ← 1,  $N_v$  do
18.    for j ← 1,  $N_h$  do
19.      for k ← 1, M do
20.        créer le groupe  $g_s$  et sauvegarder ce groupe dans G
21.        s = s + 1 ;
22.        if ( $MG(i, j, k) > 1$ ) then
23.           $MTG(i, j, k) = MG(i, j, k) * \max\left(\text{th}'(i + 1), \text{tv}'(j)\right) + ((k + 1) -$ 
             $MG(i, j, k)) * \max(\text{th}'(N_L - i), \text{tv}'(j))$ ;
24.           $MTG_{min}(i, j, k) =$  Calculer la matrice en utilisant le temps des produits
            les plus proches  $T_r$  ;
25.        End if
26.      end for
27.    end for
28.  end for

```

Dans la deuxième étape on va créer deux matrices, une première matrice MTG qui contient les temps de déstockages des groupes on utilisant l'équation 20 pour chaque élément dans la matrice MTG (la matrice MTG a les mêmes dimensions de MG et MR).

$$MTG(i, j, k) = MG(i, j, k) * \max\left(\text{th}'(i + 1), \text{tv}'(j)\right) + ((k + 1) - MG(i, j, k)) * \max(\text{th}'(N_L - i), \text{tv}'(j)) \quad (20)$$

Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

Et une deuxième matrice MTG_{min} qui contient les temps de déstockages des groupes en utilisant les temps de déstockages des produits les plus proches T_r , si il y a des produits de même type dans le lot on utilise le temps de déstockage de produit le plus loin pour ce type dans T .

Pour l' exemple précédent, la matrice MTG_{min} est égale :

$$MTG_{min} = \begin{pmatrix} 0 & 0 & (T_1+T_3) & 0 & 0 & 0 & 0 \\ 0 & 0 & (T_2+T_3) & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & (T_1+T_2) & 0 & 0 \\ 0 & 0 & (T_2+T_3) & 0 & (T_2+T_3+T_1) & 0 & 0 \end{pmatrix} \quad 15$$

Concernant la dernière étape (la comparaison), on va comparer les résultats des deux matrices $MTG(i, j, k) < MTG_{min}(i, j, k)$. Dans le cas ou $MTG(i, j, k)$ sera inférieur à $MTG_{min}(i, j, k)$. Le g_s est classé dans les meilleurs groupes bg_t ($bg_t \in BG$).

A chaque fois nous choisissons un bg_t dans BG qui a la plus grande valeur de $(MTG_{min}(i, j, k) - MTG(i, j, k))$ on doit le tester, c' est-à-dire voire si les produits de bg_t se trouvent dans le lot ou pas ;

Si oui on va l' enregistrer directement et enlèvent ces derniers produits de bg_t du lot donc égal à 0 (parmi les produits qu' on doit les enlever, ceux qui ont le même type dans le lot on doit donc enlever les plus loins égal à 0)

Si non on va la refuser.

Après chaque teste on est obligé de supprimer le meilleur groupe bg_t testé, pour que ne pas répéter la procédure au même groupe.

La recherche sera arrêter soi après le test de tous les meilleurs groupes de BG, soi lorsque sa reste un seule >0 ou l'absence de produit dans le *lot*.

Finalement, si il y a des produits restés dans le lot supérieur à 0, on utilise celles les proches pour ces produits T_r .

Dans le cas d'absence de meilleur groupe, les produits les plus proches $0 \leq (R-1) \leq T_r$ donnent le temps de déstockage optimal.

Algorithm 1 Part 3

```

29. t = 0;
30. for i ← 1, Nv do

```

```

31.   for  $j \leftarrow 1, N_h$  do
32.       for  $k \leftarrow 1, M$  do
33.           if  $(MTG(i, j, k) < MTG_{min}(i, j, k))$  then
34.                $bg_t = g_s$  (Enregistrer le groupe  $g_s$  comme meilleur groupe  $bg_t$  dans
                    BG);
35.                $t = t + 1$  ;
36.           End if
37.       end for
38.   end for
39. end for
40. finalement on choisit les meilleures solutions dans BG s'il y a des produits dans le lot
    qui n'ont pas de meilleurs groupes  $bg_t$  on utilise le produit le plus proche  $T_r$ ;
41. if (nous n'avons pas de meilleurs groupes à BG) then
42.     la meilleure solution est le produits les plus proches  $\sum_0^R T_r$ ;
43. end if

```

3.7. Résultats

Pour évaluer notre méthode, nous considérons un AS/RS a convoyeur gravitationnel qui a un nombre de casiers horizontaux N_h est égal à 20 casiers, le nombre de casiers verticaux N_v est égal à 20 casiers, le nombre d'emplacement M dans le casier est égal à 10 emplacements, La Vitesse de déplacement horizontal t_h et vertical t_v de la machine de déstockage est égal à 1, Le taux de remplissage initial du rack est égal à 75%, ainsi que nous avons variés le nombre de produits et la taille du lot.

Nous avons exécuté plusieurs simulations pour comparer notre heuristique avec les autres méthodes (SA, TS et NN). Les résultats résumés des simulations sont présentés dans les tableaux 3.1, 3.2, 3.3, 3.4, 3.5 et 3.6, chaque tableau indique les résultats obtenu pour une seule taille de lot, les tailles du lot utilisées sont 2, 3, 5, 8, 10 et 15 respectivement.

La première colonne dans ces tableaux indique le nombre de produits différents et les autres colonnes indiquent les résultats de la plus proche (NN), notre heuristique, recuit simulé (SA) et recherche tabou (TS). Chaque valeur dans ces tableaux présente le temps moyen de déstockage d'une pièce pour une taille de lot déterminée, en fonction avec le nombre de produits différents dans l'AS/RS (ligne), et en fonction avec la méthode concernée (colonne).

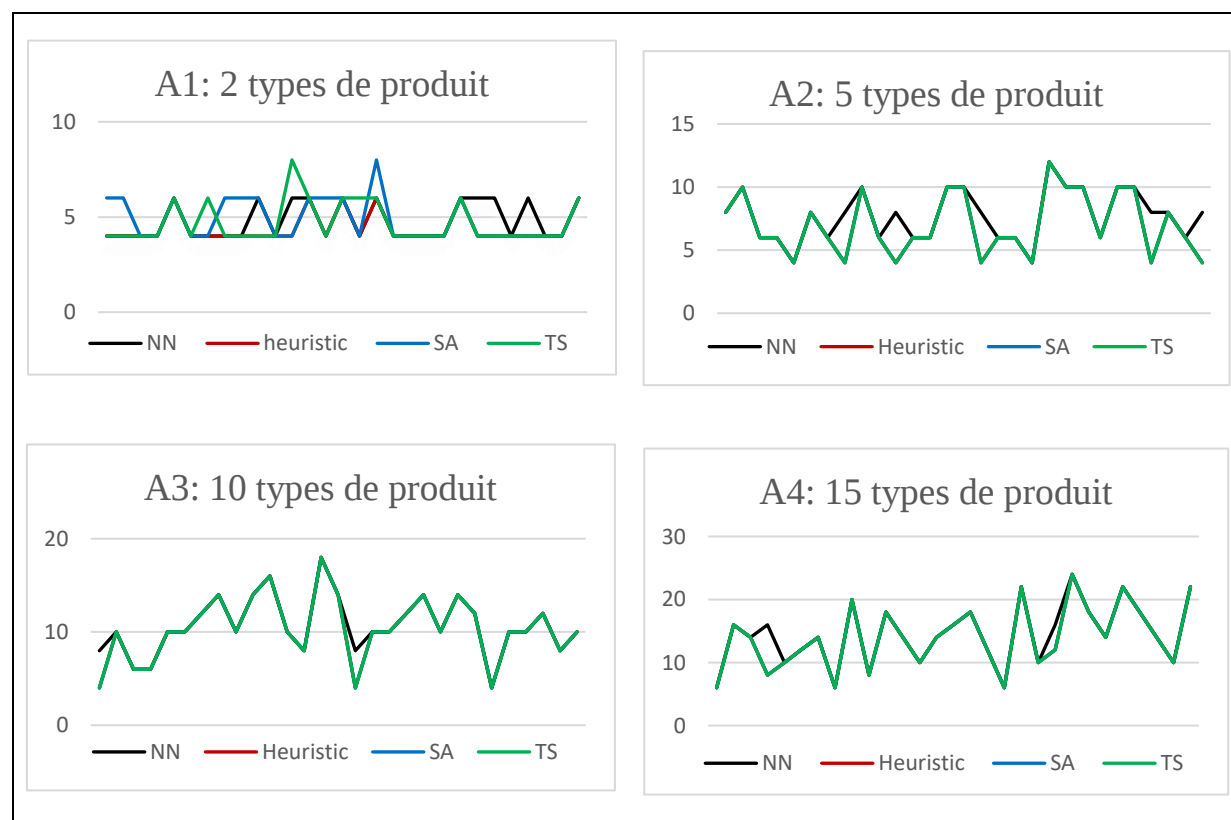
La méthode de voisinage utilisée dans les métaheuristiques (le recuit simulé (SA) et recherche tabou (TS)) consiste à sélectionner aléatoirement 3 emplacements dans la requête des demandes de déstockage (lot), puis à remplacer ces emplacements par d'autres emplacements dans le rack du même type. Enfin, les emplacements du lot sont ordonnés en fonction du temps de déstockage, c'est-à-dire que le produit dont le temps de déstockage est minimum sera déstocké en premier.

NB : Dans les tailles de lot 2, 3, 5, 8 et 10 nous avons utilisé pour le recuit simulé une température initiale $T_i = 28$, une température finale $T_f = 0.001$ avec un coefficient de refroidissement $\alpha = 0.9994$. Et pour recherche taboue nous avons utilisé une liste taboue de taille $L = 15$ et le nombre d'itérations $N_i = 13000$.

3.7.1. Temps moyen de déstockage/ Taille de lot = 2

Tableau 3. 1: le temps moyen de déstockage d'un produit pour une taille de lot égale à 2.

Nombre de types	NN	Heuristique	SA	TS
2	2,43	2,25	2,41	2,41
5	3,84	3,52	3,52	3,53
10	5,33	5,11	5,11	5,11
15	7,26	6,95	6,97	7,02
20	8,2	8,1	8,1	8,1
25	9,27	9,06	9,06	9,08



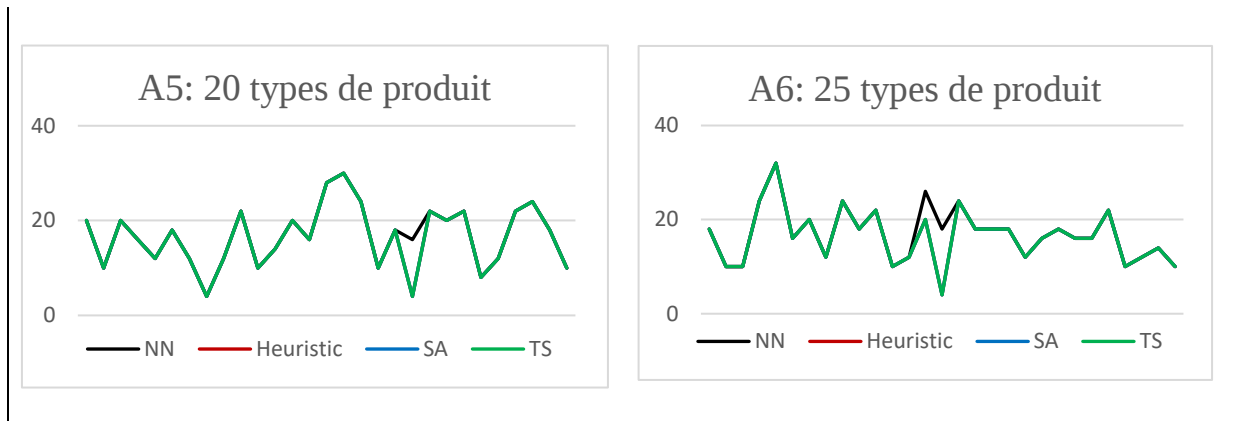


Figure 3. 4 : Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 2

Nous avons remarqué que dans la taille du lot égale à 2 :

- Notre méthode donne de bons résultats dans le cas où le nombre de produits différents est égale à 2 (A1 dans la Figure 3.4) par rapport aux autres méthodes (NN, SA et TS). Et dans les cas où le nombre de produits différents est moyen ou important (A2, A3, A4, A5 et A6 dans la Figure 3.4), les résultats de notre méthode et les résultats des deux méthodes de la métaheuristiques (SA et TS) sont très proches.
- La méthode NN donne de mauvais résultats (Figure 3.5) par rapport aux notre méthode et les deux méthodes de la métaheuristiques (SA et TS) dans tous les cas (le nombre de produits différents est faible, moyen et important).

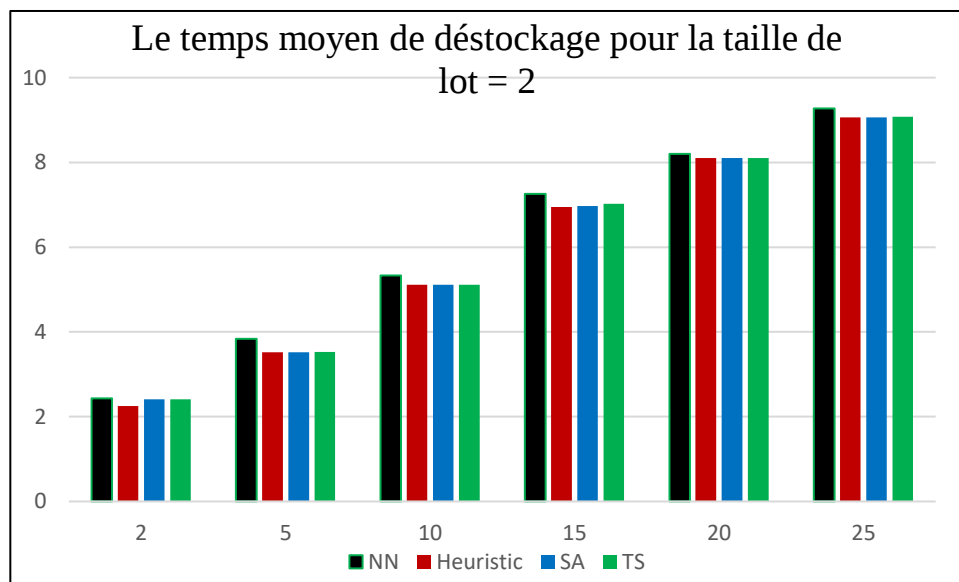


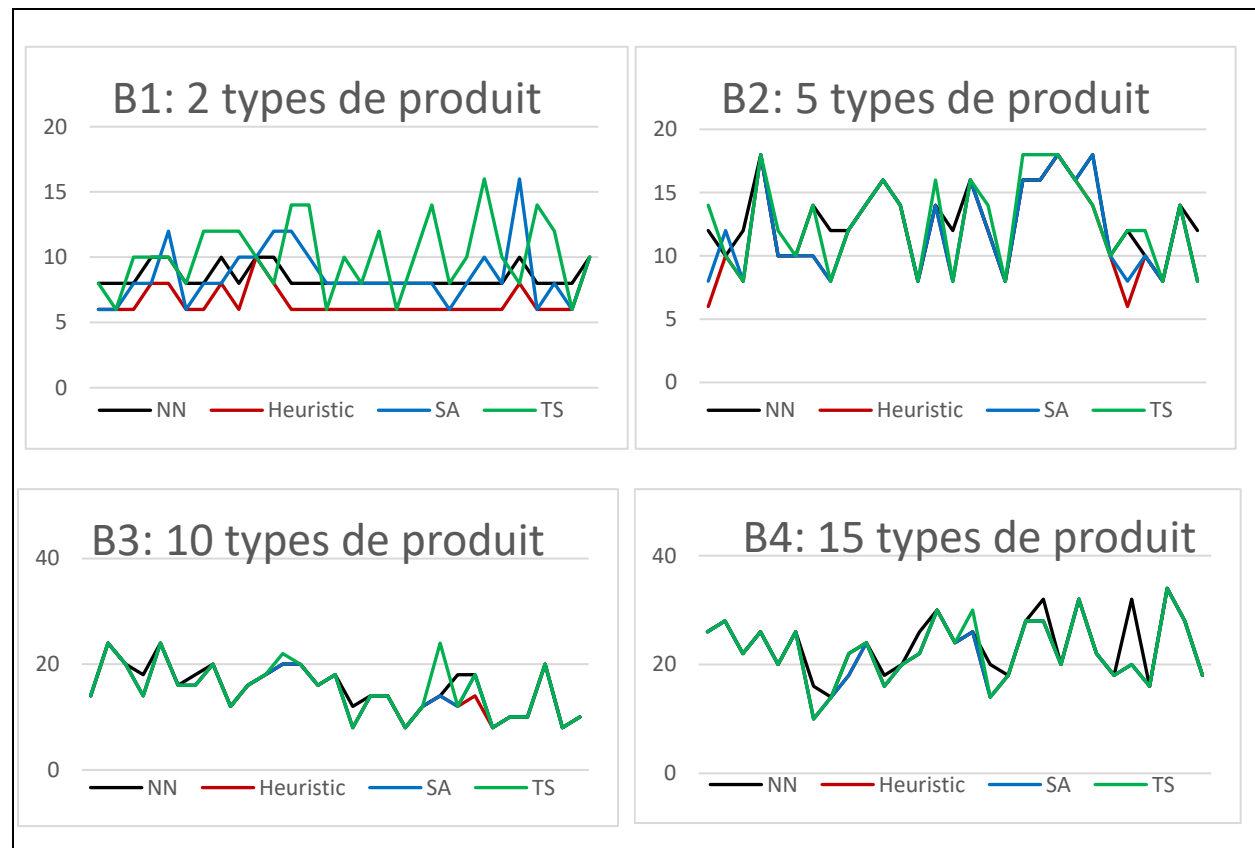
Figure 3. 5 : Le temps de déstockage pour la taille de lot = 2

3.7.2. Temps moyen de déstockage/ Taille de lot = 3

Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

Tableau 3. 2 le temps moyen de déstockage d'un produit pour une taille de lot égale à 3.

Nombre de types	NN	Heuristique	SA	TS
2	2,813333	2,233333	2,88	3,113333
5	4,046667	3,473333	3,58	3,866667
10	5,38	5,06	5,153333	5,186667
15	7,326667	6,966667	7,013333	7,066667
20	7,966667	7,693333	7,7	7,7
25	9,233333	8,98	8,993333	9,013333



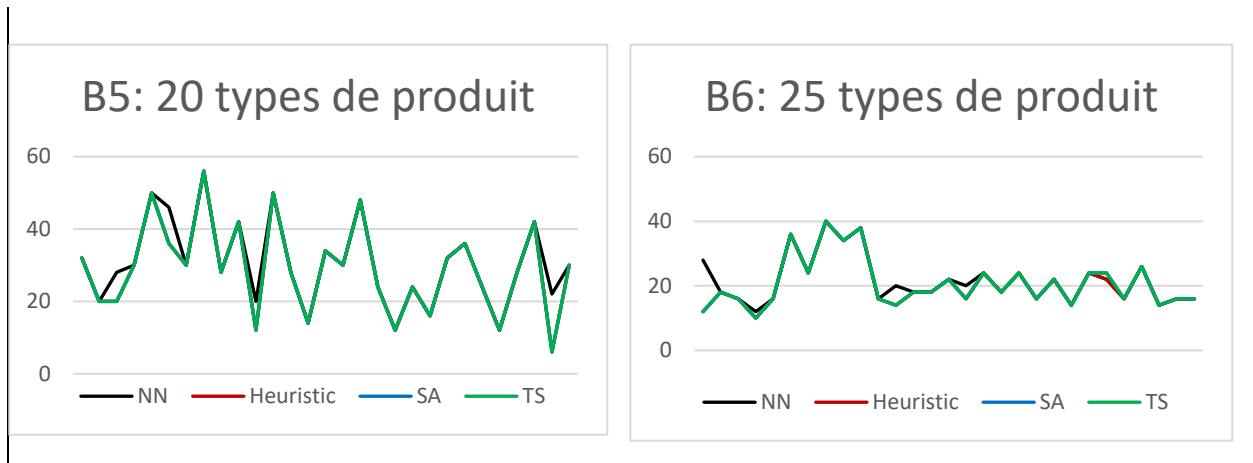


Figure 3. 6 : Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 3

Nous avons remarqué que dans la taille du lot égale à 3 :

- Notre méthode donne de bons résultats dans le cas où le nombre de produits différents est faible ou moyen (B1, B2, B3 et B4 dans la Figure 3.6) par rapport aux deux méthodes de la métaheuristiques (SA et TS). Et dans les cas où le nombre de produits différents est important (B5 et B6 dans la Figure 3.6), les résultats du notre méthode et les résultats des deux méthodes de la métaheuristiques (SA et TS) sont très proches.
- Notre méthode donne de bons résultats dans tous les cas (Figure 3.7) par rapport à la méthode NN.
- La méthode NN donne de bons résultats (Figure 3.6) par rapport aux méthodes SA et TS dans le cas où le nombre de produits différents est égale à 2 (B1 dans la Figure 3.6). Et dans les cas où le nombre de produits différents est moyen et important (B2, B3, B4, B5 et B6

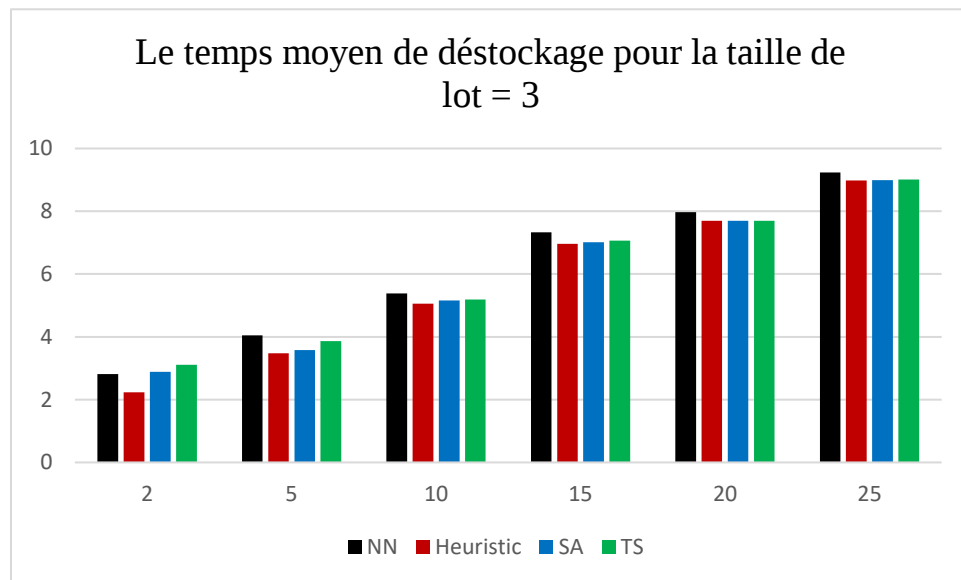


Figure 3. 7 : Le temps de déstockage pour la taille de lot = 3

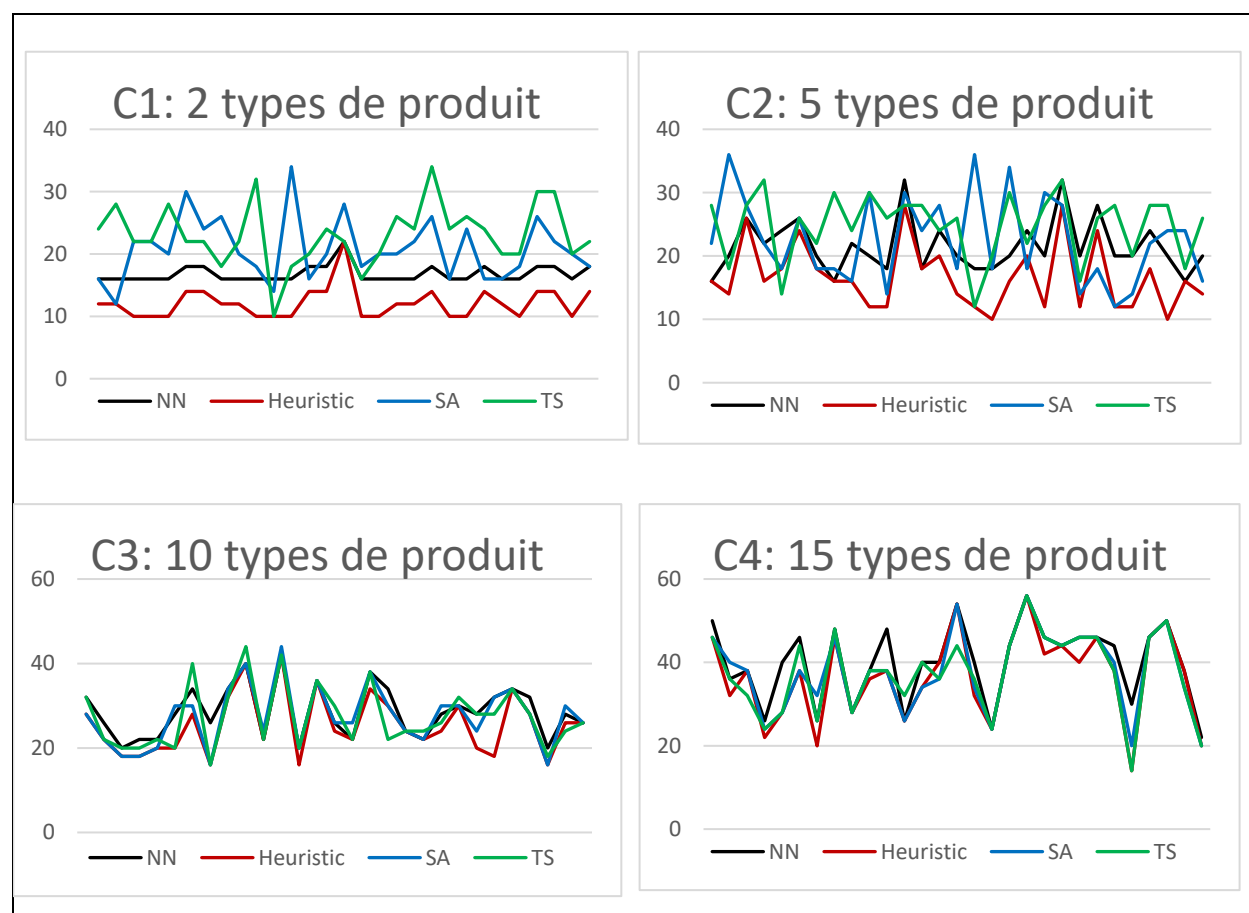
Chapitre 3 : Minimisation du temps de déstockage du lot (une requête de demande de stockage et de déstockage)

dans la Figure 3.6), les deux méthodes de la métaheuristiques (SA et TS) donnent de bons résultats par rapport à la méthode NN.

3.7.3. Temps moyen de déstockage/ Taille de lot = 5

Tableau 3. 3 le temps moyen de déstockage d'un produit pour une taille de lot égale à 5.

Nombre de types	NN	Heuristique	SA	TS
2	3,392	2,488	4,272	4,444
5	4,552	3,608	4,672	5,072
10	5,804	5,164	5,656	5,624
15	7,736	7,128	7,328	7,408
20	8,248	7,68	7,664	7,828
25	9,8	9,26	9,288	9,46



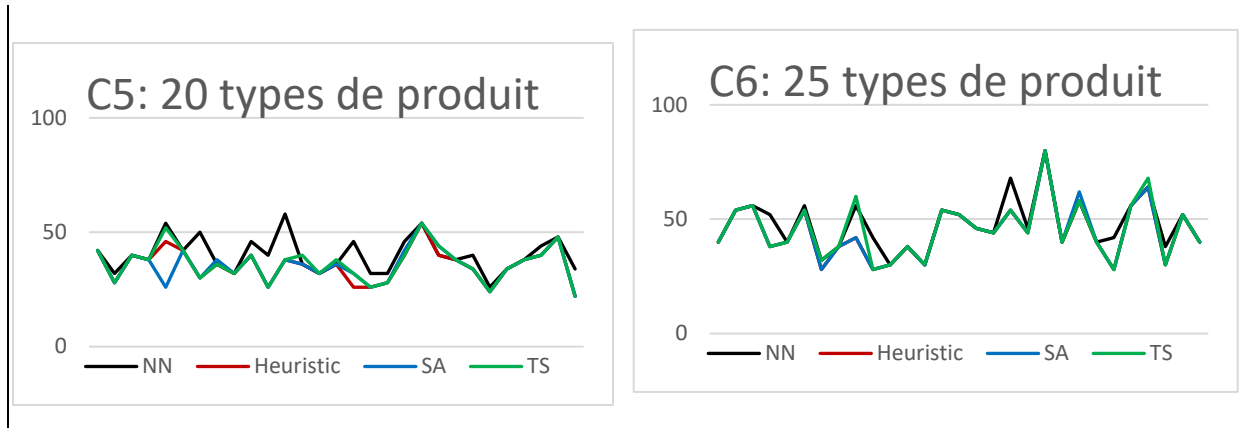


Figure 3. 8 : Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 5

Nous avons remarqué que dans la taille du lot égale à 5 :

- Notre méthode donne de bons résultats dans le cas où le nombre de produits différents est faible ou moyen (C1, C2, C3 et C4 dans la Figure 3.8) par rapport à la méthode SA. Et dans les cas où le nombre de produits différents est important (C5 et C6 dans la Figure 3.8), les résultats de notre méthode et les résultats de la méthode SA sont très proches.
- Notre méthode donne de bons résultats dans tous les cas (Figure 3.9) par rapport à la méthode NN et TS.
- La méthode NN donne de bons résultats par rapport aux méthodes SA et TS dans le cas où le nombre de produits différents est faible (C1 et C2 dans la Figure 3.8). Et dans les cas où le nombre de produits différents est moyen et important (C3, C4, C5 et C6 dans la Figure 3.8), les deux méthodes de métaheuristiques (SA et TS) donnent de bons résultats par rapport à la méthode NN.

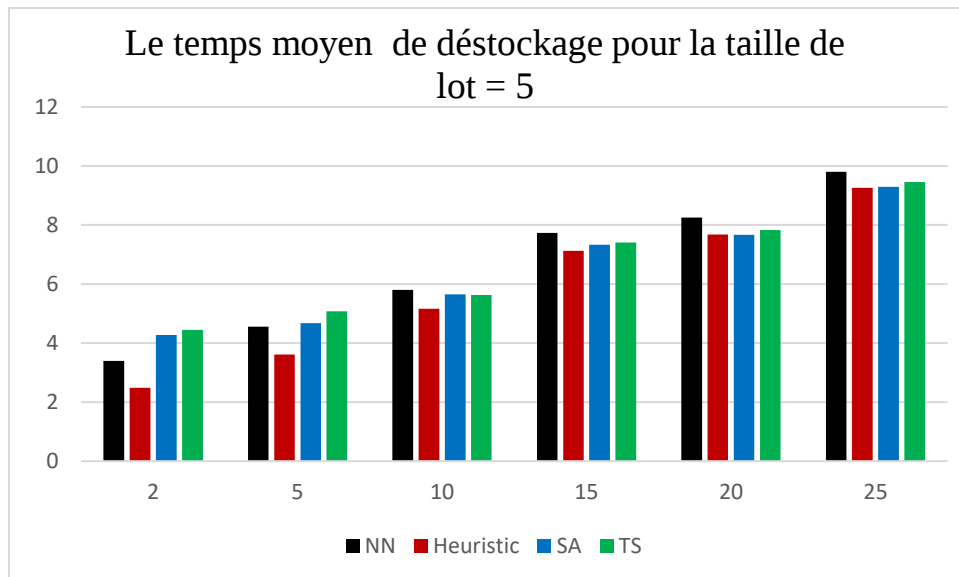
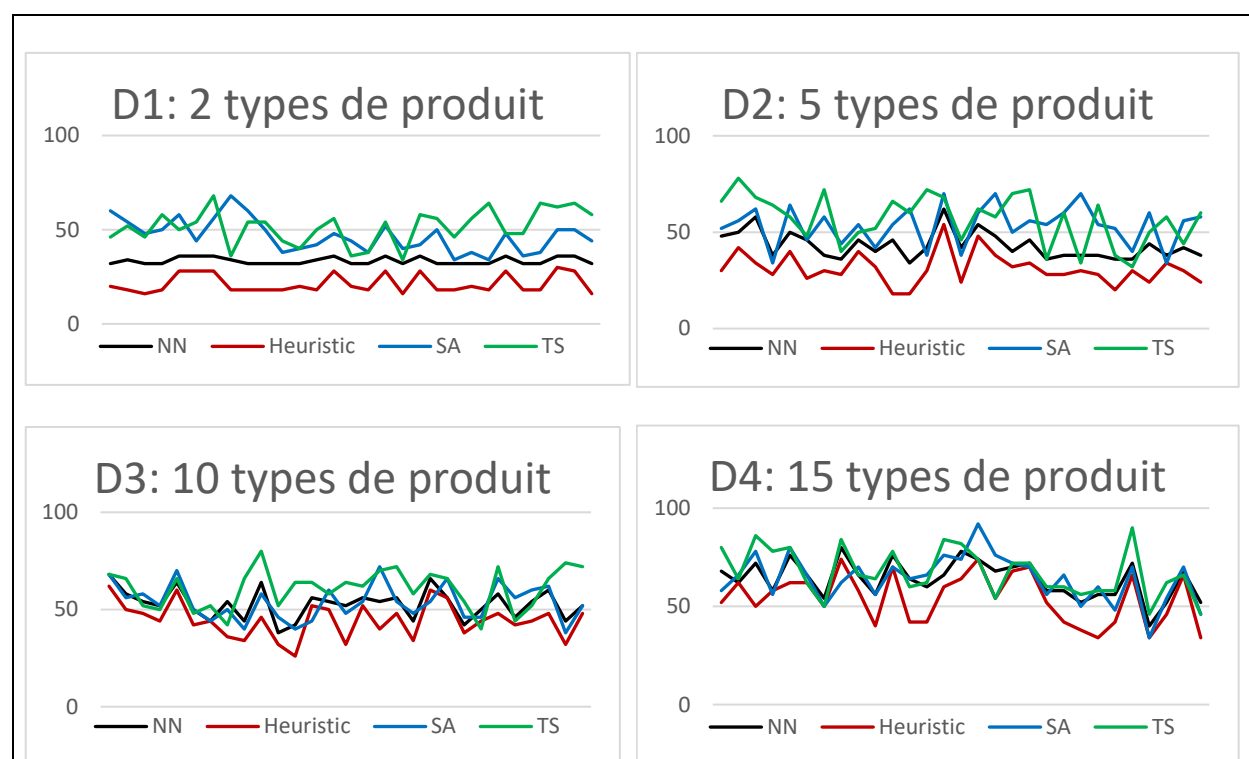


Figure 3. 9 : Le temps de déstockage pour la taille de lot = 5

3.7.4. Temps moyen de déstockage/ Taille de lot = 8

Tableau 3. 4 : le temps moyen de déstockage d'un produit pour une taille de lot égale à 8.

Nombre de types	NN	Heuristic	SA	TS
2	4,1475	2,5475	5,8125	6,2
5	5,2125	3,8275	6,52	6,725
10	6,2325	5,1925	6,6575	7,0225
15	8,0975	6,9225	8,1575	8,235
20	8,5	7,475	8,21	8,52
25	10,0125	9,205	9,575	9,75



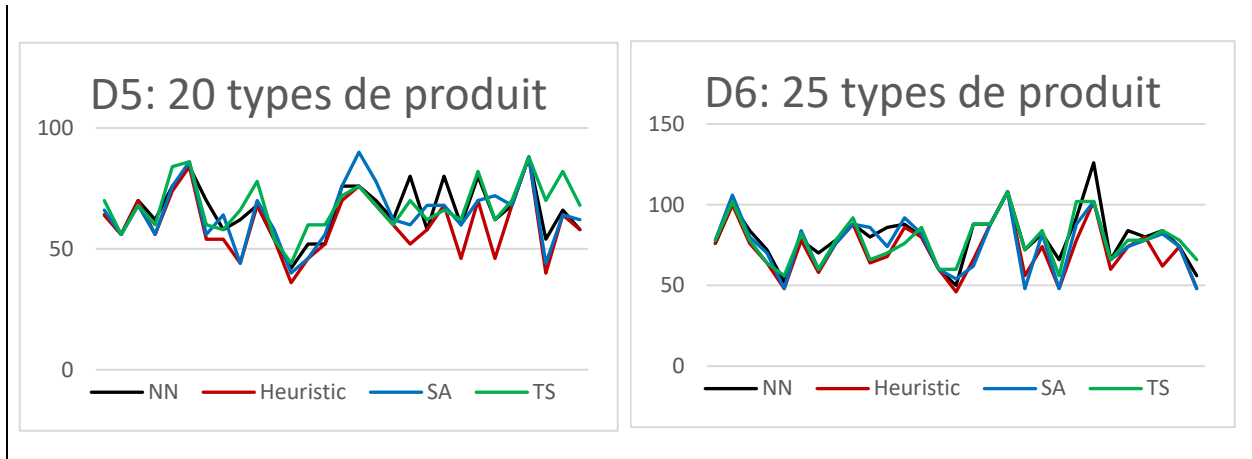


Figure 3. 10 : Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 8

Nous avons remarqué que dans la taille pour un lot égale à 8 :

- Notre méthode donne de bons résultats dans tous les cas où le nombre de produits différents est faible, moyen et important (Figure 3.10) par rapport aux autres méthodes NN, SA et TS, on peut remarquer qu'il y a une grande différence dans les cas où le nombre de produits différents est faible (Figure 3.11), et cette différence diminue à chaque fois que nous augmentons le nombre de produits différents.
- La méthode NN donne de bons résultats (Figure 3.10) par rapport aux méthodes SA et TS dans le cas où le nombre de produits différents est faible et moyen (D1, D2, D3 et D4 dans la Figure 3.10). Et dans les cas où le nombre de produits différents est important (D5 et D6 dans la Figure 3.10), les deux méthodes de la métaheuristiques (SA et TS) donnent de bons résultats par rapport à la méthode NN.

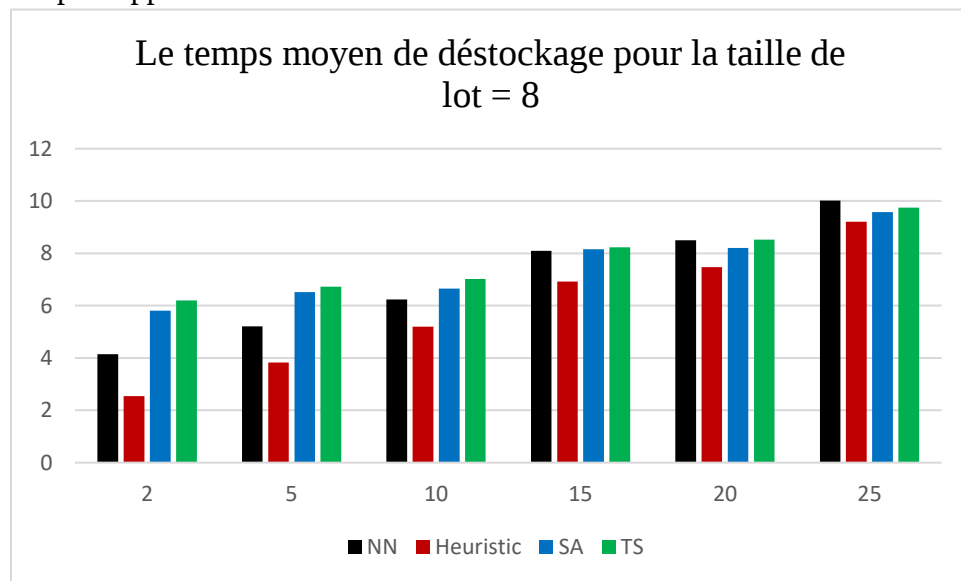
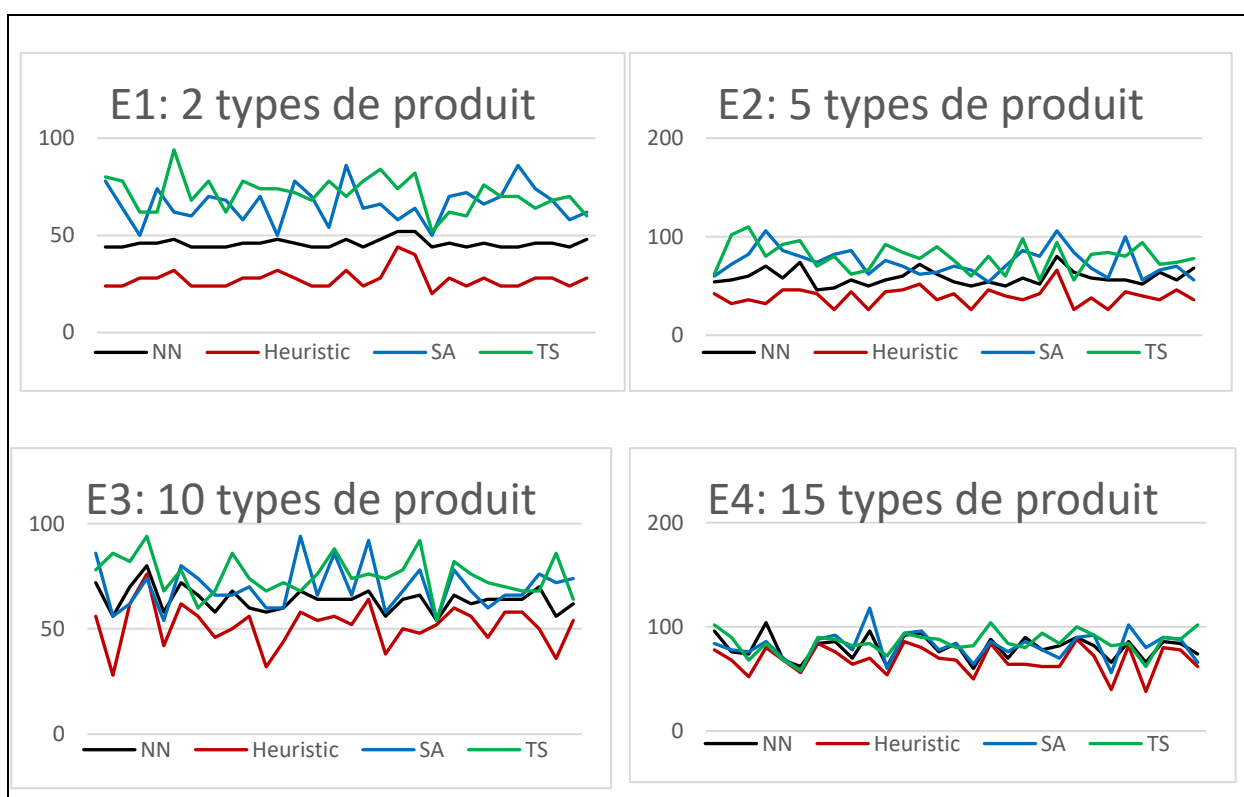


Figure 3. 11 : Le temps de déstockage pour la taille de lot = 8

3.7.5. Temps moyen de déstockage/ Taille de lot = 10

Tableau 3. 5 le temps moyen de déstockage d'un produit pour une taille de lot égale à 10.

Nombre de types	NN	Heuristic	SA	TS
2	4,56	2,672	6,702	7,24
5	5,798	3,928	7,544	7,954
10	6,402	5,146	7,286	7,76
15	8,15	6,828	8,366	8,694
20	8,826	7,51	8,718	9,066
25	10,102	9,156	10,034	10,232



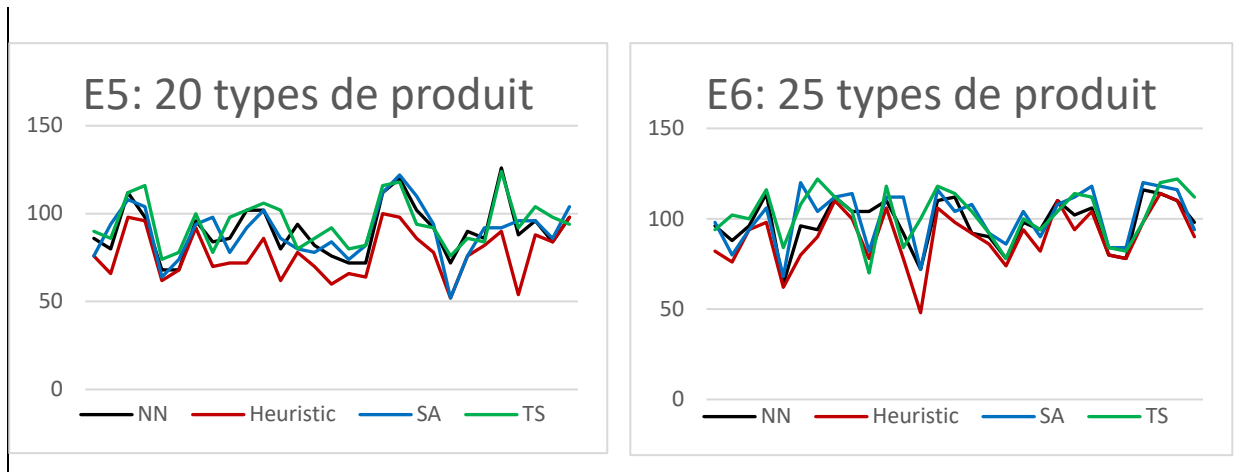


Figure 3. 12 Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 10

Nous avons remarqué que dans la taille du lot égale à 10 :

- Notre méthode donne de bons résultats dans tous les cas (le nombre de produits différents est faible, moyen et important) (Figure 3.12) par rapport aux autres méthodes NN, SA et TS, on peut remarquer qu'il y a une grande différence dans les cas où le nombre de produits différents est faible (Figure 3.13), et cette différence diminue à chaque fois que nous augmentons le nombre de produits différents.
- la méthode NN et SA donne de bons résultats dans tous les cas (Figure 3.13) par rapport à la méthode TS.
- La méthode NN donne de bons résultats par rapport à la méthode SA dans le cas où le nombre de produits différents est faible et moyen (E1, E2, E3 et E4 dans la Figure 3.12). Et dans les cas où le nombre de produits différents est important (E5 et E6 dans la Figure 3.12), la méthode SA donne de bons résultats par rapport à la méthode NN.

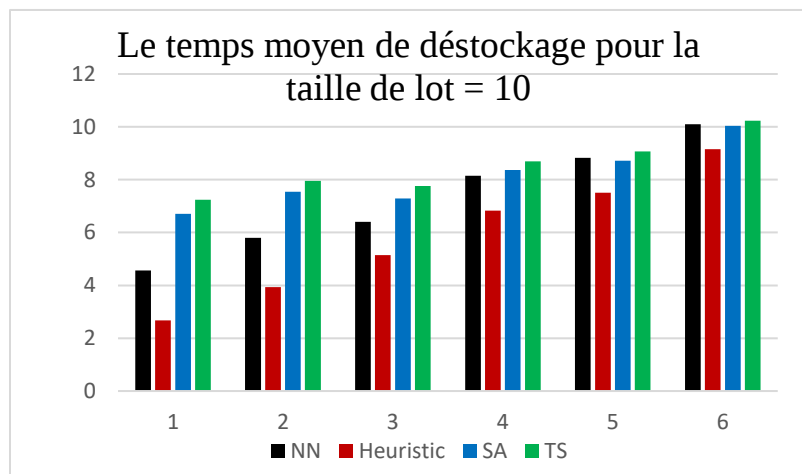


Figure 3. 13 : Le temps de déstockage pour la taille de lot = 10

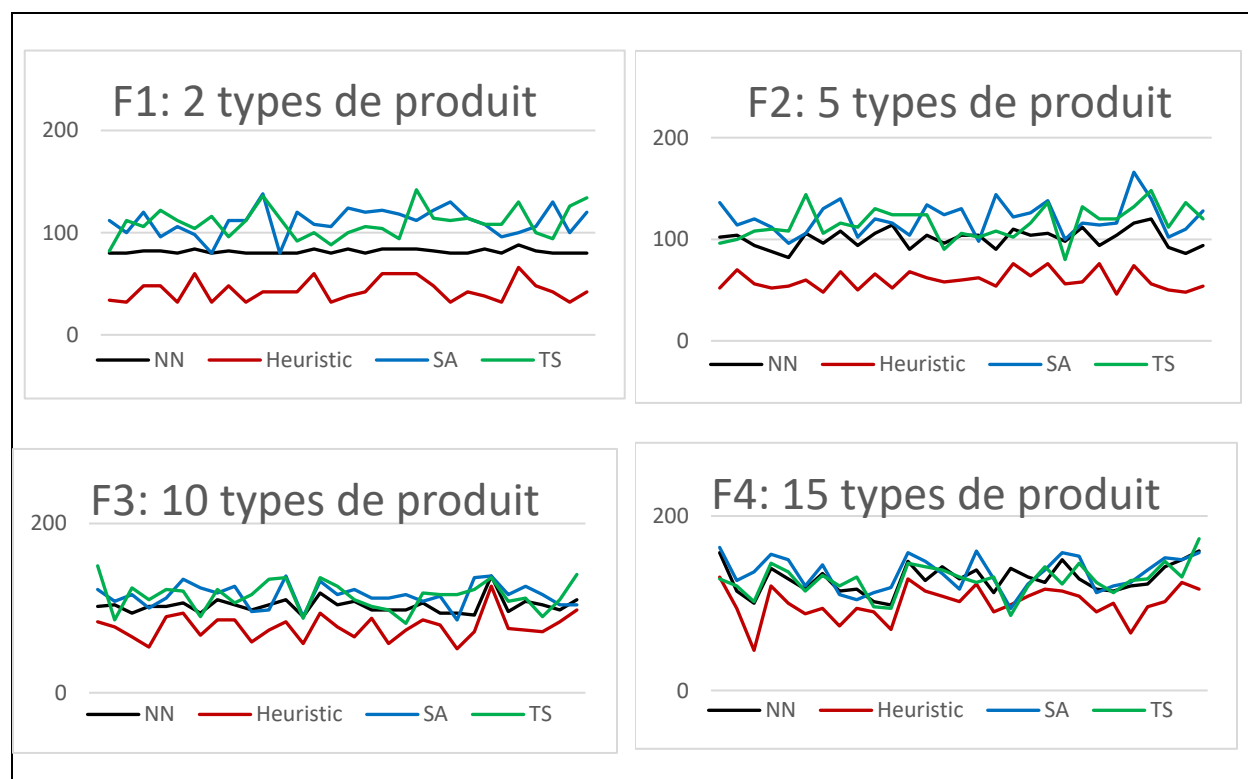
3.7.6. Temps moyen de déstockage/ Taille de lot = 15

Contrairement aux tailles des lots précédentes, dans cette taille du lot (15) on a changé les paramètres des algorithmes (TS et SA) donc une température initiale $T_i = 52$, une température finale $T_f = 0.001$, et un coefficient de refroidissement $\alpha = 0.9996$ pour le recuit simulé. Concernant la recherche taboue, nous avons utilisé une liste taboue de taille $L=15$ et un nombre d'itérations $N_i=26000$.

Ce changement a été effectué pour un but très important, il s'agit que notre méthode donne des même résultats à chaque fois qu'on modifié les valeurs des algorithmes (RT et SA), ce que signifié la fiabilité et la ponctualité de notre méthode.

Tableau 3. 6 : le temps moyen de déstockage d'un produit pour une taille de lot égale à 15.

Nombre de types	NN	Heuristic	SA	TS
2	5,436	2,824	7,52	7,258667
5	6,764	4,172	8,066667	7,801333
10	7,089333	5,473333	7,922667	7,701333
15	8,782667	6,830667	8,92	8,794667
20	9,013333	7,278667	8,813333	8,769333
25	10,43733	8,488	9,850667	9,785333



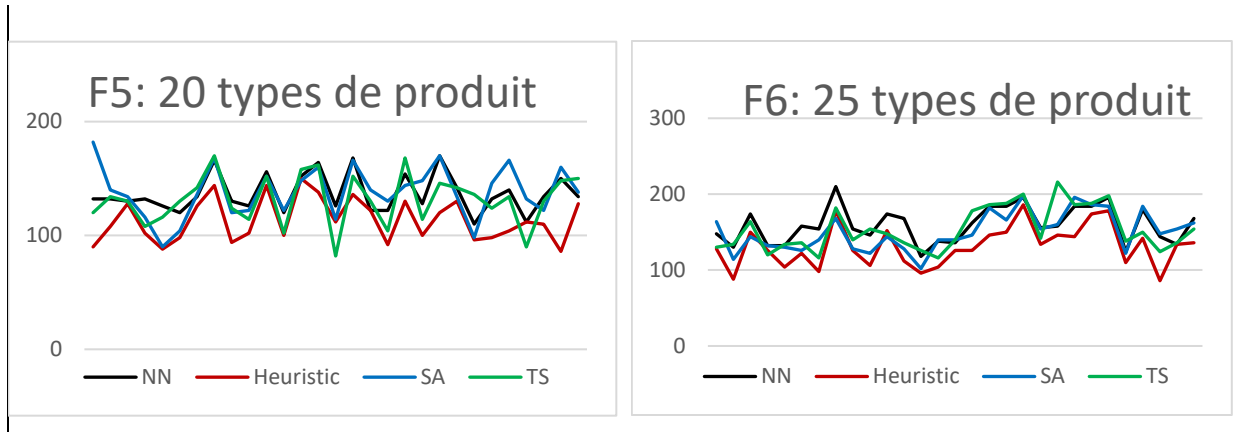


Figure 3. 14: Le temps de déstockage pour 5/10/15/20/25 types de produit dans la taille du lot = 15

Nous avons remarqué que dans la taille du lot égale à 15 :

- Notre méthode donne de bons résultats dans tous les cas (le nombre de produits différents est faible, moyen et important (Figure 3.14) par rapport aux autres méthodes NN, SA et TS, on peut remarquer qu'il y a une grande différence dans les cas où le nombre de produits différents est faible (Figure 3.15), et cette différence diminue à chaque fois que nous augmentons le nombre de produits différents.
- La méthode NN donne de bons résultats (Figure 3.15) par rapport aux méthodes SA et TS dans le cas où le nombre de produits différents est faible et moyen (F1, F2, F3 et F4 dans la Figure 3.14). Et dans les cas où le nombre de produits différents est important (F5 et F6 dans la Figure 3.14), les deux méthodes de la métaheuristiques (SA et TS) donnent de bons résultats par rapport à la méthode NN.

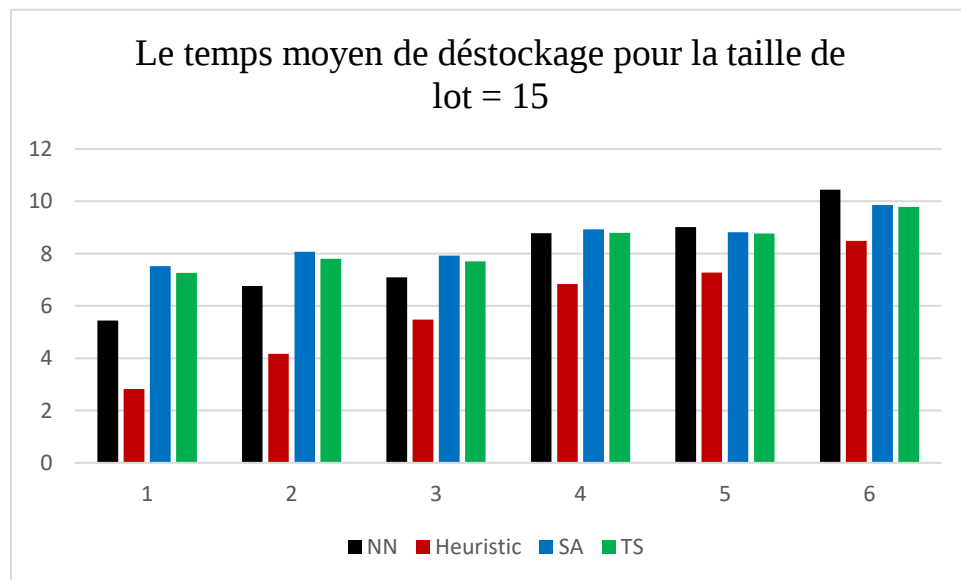


Figure 3. 15 : Le temps de déstockage pour la taille de lot = 15

3.7.7. Résumé

Le dernier tableau (Tableau 3.7) montre le temps de calcul (simulation) d'une pièce en fonction de la taille du lot (ligne) pour chaque méthode (colonne).

La première colonne dans ce tableau indique la taille du lot, et les autres colonnes indiquent le temps de calcul de chaque méthodes (SA, TS et NN).

Temps de calcul

Tableau 3. 7: le temps de calcul pour un produit en fonction de la taille du lot pour chaque

Taille du lot	NN (s)	Heuristique (s)	SA (s)	TS (s)
2	0.9994429329999548	1.0197980649999997	42.298547579999999	29.238400963000004
3	1.0183688169999982	1.261252651999996	46.02174114900001	30.38438025100001
5	1.1383016949999842	1.2331465799999933	46.214755697999999	32.51695745000001
8	1.1767640100000108	1.206766749999579	46.70173950600008	34.388641892999954
10	1.0812763600000608	1.1882456179999963	49.820587046000014	40.19030796000004
15	1.1680407060000562	1.3525008960000378	107.50965628899985	123.67319257100007

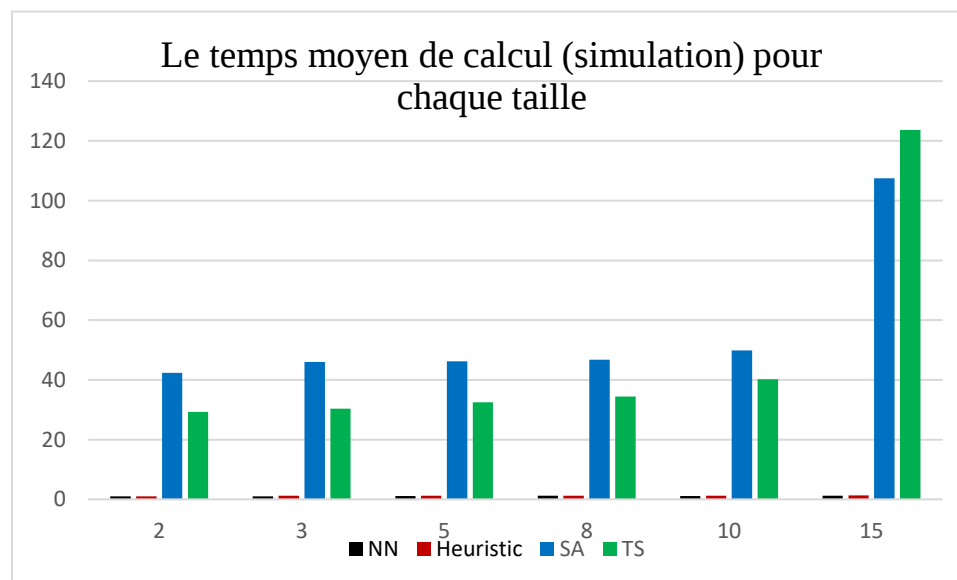


Figure 3. 16 : le temps de calcul des méthodes pour chaque taille du lot

Après cette étude on a trouvé les résultats suivants :

- Nous avons remarqué que notre méthode donne de bons résultats dans tous les cas par rapport aux autres méthodes (NN, SA et TS), on peut remarquer une grande différence dans les cas où le nombre de produits différents est faible pour toutes les tailles de lot, et cette différence augmente à chaque fois que nous augmentons la taille du lot et diminue à chaque fois que nous augmentons le nombre de produits différents.

- La méthode NN donne aussi de bons résultats par rapport aux métaheuristiques (SA et TS) dans les cas où le nombre de produits différents est faible ou moyen (2, 5, 10 et 15) et la taille du lot est moyenne ou supérieur (8, 10 et 15). Et dans les cas où le nombre de produits différents est très faible (2) et la taille du lot est faible (2 et 3).
- Les métaheuristiques (SA et TS) donne des résultats satisfaisantes par rapport aux NN dans les cas où le nombre de produits différents est supérieur (20 et 25) et la taille du lot est moyenne ou sup (8, 10 et 15). Et dans les cas où la taille du lot est faible 2, 3 et 5.
- Le temps moyen de déstockage augmente à chaque fois que nous augmentons la taille du lot dans les cas où le nombre de produits différents est faible (2, 5 et 10).
- Le temps moyen de déstockage va diminuer chaque fois que la taille du lot est augmentée, dans les cas où le nombre de produits différents est important : 15, 20 et 25.
- Notre méthode et la méthode NN donnent les résultats dans un temps très réduit presque négligeable (Figure 3.16) ; au contraire, les résultats de la métaheuristiques nécessitent un temps assez important pour se calculent. Ce dernier, augmente à chaque fois que nous augmentons la taille du lot.

3.8. Conclusion

Ce chapitre conclut cette thèse. Il comporte les principales contributions. Toutes les contributions de cette thèse s'articulent autour de l'adaptation de techniques de l'intelligence artificielle pour la minimisation du temps moyen de déstockage d'un lot de produits. Nous avons d'abord modélisé le problème mathématiquement, Nous avons ensuite proposé une heuristique dédiée à la résolution du problème étudié. Et enfin nous avons adapté les deux métaheuristiques : Recuit Simulé et Recherche Taboue au même problème. Les résultats obtenus ont démontré une grande précision de l'heuristique proposée.

Conclusion Générale

Conclusion Générale

Dans ce travail de thèse, nous nous sommes intéressés à la minimisation du temps moyen de déstockage d'un lot de charges à partir d'un AS/RS profond. Pour cela nous avons utilisé des techniques de l'intelligence artificielle, à savoir les machines learning et les métaheuristiques. Dans les AS/RS profond l'accès à une charge pour la déstocker n'est pas immédiat. Il est souvent nécessaire de retirer toutes les charges la précédant avant de la déstocker. Parmi les charges retirées, il est possible d'avoir une que nous souhaitons déstocker dans la suite. C'est pour cette raison que nous avons considéré le déstockage de lots de produits au lieu de produits un à un.

Nous pouvons résumer les contributions de cette thèse en deux parties. Dans la première partie, nous avons utilisé la technique de la machine learning pour apprendre à une intelligence artificielle de choisir la meilleure position à chaque fois qu'une nouvelle charge arrive au système ; de telle sorte que le temps moyen de déstockage dans ce système est minimisé. En d'autres termes, nous rangeons les charges à l'intérieur des racks, afin de pouvoir prédire le meilleur emplacement pour chaque charge de manière à minimiser son temps de déstockage. Le temps total (moyen) de déstockage est également minimisé. Pour cela, nous avons utilisé des ensembles de données de différentes dimensions pour obtenir un modèle machine learning généralisé, qui peut être utilisé pour les AS/RS de n'importe quelle dimension. Nous avons entraîné le modèle à classer les emplacements dans le rack en fonction de leur état (vide ou plein) en utilisant la classification. Ensuite, nous l'avons entraîné à calculer le temps de déstockage requis pour chaque emplacement à l'aide de la régression polynomiale multi-variable. Ce qui nous a donné des résultats très satisfaisants. À partir de ces résultats, nous déterminons la meilleure solution (temps d'accès minimal), ce qui nous donne un temps de déstockage moyen minimal. Enfin, nous avons comparé le modèle avec le stockage aléatoire et le stockage dans l'emplacement donnant un temps de déstockage minimal.

La deuxième contribution de cette thèse, présentée en chapitre 3, consiste en la minimisation du temps moyen d'un lot de produits. Ce problème est assez particulier et ne ressemble à aucun des problèmes classiques que l'on trouve dans le domaine du génie industriel. Nous avons d'abord proposé une modélisation mathématique du problème décrit. Nous avons ensuite proposé une heuristique permettant de résoudre le modèle mathématique proposé, nous l'avons appelé heuristique de minimisation du temps de déstockage par lots (BRTM-heuristique). Nous avons également adapté les deux métaheuristiques : le recuit simulé (SA) et la recherche tabou (TS) à la résolution du problème. Les résultats de l'heuristique BRTM seront comparés aux résultats donnés par les deux métaheuristiques afin de valider la précision de l'heuristique BRTM.

References

- Anand, Rimmi, Divya Aggarwal, and Vijay Kumar. 2017. "A Comparative Analysis of Optimization Solvers." *Journal of Statistics and Management Systems* 20 (4): 623–35.
- Atz, Thomas, Daniel Lantschner, and Willibald A Günthner. n.d. "SIMULATIVE THROUGHPUT CALCULATION FOR STORAGE PLANNING," 10.
- Bertolini, Massimo, Giovanni Esposito, Davide Mezzogori, and Mattia Neroni. 2019. "Optimizing Retrieving Performance of an Automated Warehouse for Unconventional Stock Keeping Units." *Procedia Manufacturing* 39: 1681–90. <https://doi.org/10.1016/j.promfg.2020.01.272>.
- Bessenouci, Hakim Nadir, Zaki Sari, and Latéfa Ghomri. 2012. "Metaheuristic Based Control of a Flow Rack Automated Storage Retrieval System." *Journal of Intelligent Manufacturing* 23 (4): 1157–66. <https://doi.org/10.1007/s10845-010-0432-1>.
- Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Information Science and Statistics. New York: Springer.
- Bozer, Y. A., J. S. Shiek, and J. A. White. 1982. "Modelling Automated Storage/Retrieval Systems." In *TIMS/ORSA Joint National Meeting, Detroit, MI*.
- Bozer, Yavuz A., and John A. White. 1984. "Travel-Time Models for Automated Storage/Retrieval Systems." *IIE Transactions* 16 (4): 329–38.
- Bukchin, Yossi, and Tal Raviv. 2020. "Optimal Retrieval in Puzzle-Based Storage Systems with Simultaneous Load and Block Movement." <https://doi.org/10.13140/RG.2.2.29406.05442>.
- Cardin, Olivier, Pierre Castagna, Zaki Sari, and Nihad Meghelli. 2012. "Performance Evaluation of In-Deep Class Storage for Flow-Rack AS/RS." *International Journal of Production Research* 50 (23): 6775–91. <https://doi.org/10.1080/00207543.2011.624561>.
- Černý, Vladimír. 1985. "Thermodynamical Approach to the Traveling Salesman Problem: An Efficient Simulation Algorithm." *Journal of Optimization Theory and Applications* 45 (1): 41–51.
- Chang, Ding-Tsair, Ue-Pyng Wen, and JAMES T. LIN. 1995. "The Impact of Acceleration/Deceleration on Travel-Time Models for Automated Storage/Retrieval Systems." *IIE Transactions* 27 (1): 108–11.
- Chen, Zhuxi, and Jatinder N D Gupta. n.d. "A Three-Step Heuristic for Bi-Directional Flow-Rack AS/RS with Duration-of-Stay Storage Policy," 7.
- Chen, Zhuxi, Xiaoping Li, and Jatinder N.D. Gupta. 2015. "A Bi-Directional Flow-Rack Automated Storage and Retrieval System for Unit-Load Warehouses." *International Journal of Production Research* 53 (14): 4176–88. <https://doi.org/10.1080/00207543.2014.980459>.
- . 2016. "Sequencing the Storages and Retrievals for Flow-Rack Automated Storage and Retrieval Systems with Duration-of-Stay Storage Policy." *International Journal of Production Research* 54 (4): 984–98. <https://doi.org/10.1080/00207543.2015.1035816>.
- Chetty, OV Krishnaiah, and M. Sarveswar Reddy. 2003. "Genetic Algorithms for Studies on AS/RS Integrated with Machines." *The International Journal of Advanced Manufacturing Technology* 22: 932–40.

References

- Chun Park, Byung, Robert D. Foley, John A. White, and Edward H. Frazelle. 2003. "Dual Command Travel Times and Miniload System Throughput with Turnover-Based Storage." *Iie Transactions* 35 (4): 343–55.
- Dallari, Fabrizio, Gino Marchet, and Marco Melacini. 2009. "Design of Order Picking System." *The International Journal of Advanced Manufacturing Technology* 42 (1): 1–12.
- D'Antonio, Gianluca, and Paolo Chiabert. 2019a. "Analytical Models for Cycle Time and Throughput Evaluation of Multi-Shuttle Deep-Lane AVS/RS." *The International Journal of Advanced Manufacturing Technology* 104 (5–8): 1919–36. <https://doi.org/10.1007/s00170-019-03985-8>.
- . 2019b. "Analytical Models for Cycle Time and Throughput Evaluation of Multi-Shuttle Deep-Lane AVS/RS." *The International Journal of Advanced Manufacturing Technology* 104 (5–8): 1919–36. <https://doi.org/10.1007/s00170-019-03985-8>.
- "Definition of Harvard | Dictionary.Com." n.d. www.Dictionary.Com. Accessed March 4, 2021. <https://www.dictionary.com/browse/harvard>.
- Eben-Chaime, Moshe. 1996. "An Integrative Model for Automatic Warehousing Systems." *International Journal of Computer Integrated Manufacturing* 9 (4): 286–92.
- Eben-Chaime, Moshe, and Nava Pliskin. 1997. "Operations Management of Multiple Machine Automatic Warehousing Systems." *International Journal of Production Economics* 51 (1–2): 83–98.
- Eder, Michael. 2020a. "An Approach for a Performance Calculation of Shuttle-Based Storage and Retrieval Systems with Multiple-Deep Storage." *The International Journal of Advanced Manufacturing Technology* 107 (1–2): 859–73. <https://doi.org/10.1007/s00170-019-04831-7>.
- . 2020b. "An Approach for Performance Evaluation of SBS/RS with Shuttle Vehicles Serving Multiple Tiers of Multiple-Deep Storage Rack." *The International Journal of Advanced Manufacturing Technology* 110 (11–12): 3241–56. <https://doi.org/10.1007/s00170-020-06033-y>.
- Egbelu, Pius J. 1991. "Framework for Dynamic Positioning of Storage/Retrieval Machines in an Automated Storage/Retrieval System." *The International Journal of Production Research* 29 (1): 17–37.
- Eldemir, Fahrettin, R. J. Graves, and C. J. Malmberg*. 2004. "New Cycle Time and Space Estimation Models for Automated Storage and Retrieval System Conceptualization." *International Journal of Production Research* 42 (22): 4767–83.
- Elsayed, E. A., and M.-K. Lee. 1996. "Order Processing in Automated Storage/Retrieval Systems with Due Dates." *IIE Transactions* 28 (7): 567–77.
- Eynan, Amit, and Meir J. Rosenblatt. 1994. "Establishing Zones in Single-Command Class-Based Rectangular AS/RS." *IIE Transactions* 26 (1): 38–46.
- Fan, Wang, Guo Dongfen, Tang Shujuan, He Hong, and Cui Xiao. 2015. "Travel Time Model of the Storage/Retrieval Machine for Multi-Deep AS/RS Based on Flexsim." *The Open Cybernetics & Systemics Journal* 9 (1): 1833–39. <https://doi.org/10.2174/1874110X01509011833>.
- Firth, Don. 1988. *Profitable Logistics Management*. McGraw-Hill Ryerson.
- Gaouar, Nihed, Zaki Sari, and Noureddine Ghouali. 2005. "Développement d'une Heuristique de Stockage/Déstockage pour un Système AS/R à Convoyeur Gravitationnel," 11.

References

- Ghomri, L., Z. Sari, A. Guezzan, and T. Sari. 2008. "Modèle Analytique Du Temps de Simple Cycle d'un AS/RS Multi Allées." In *Conférence Internationale de Modélisation et Simulation*.
- Ghomri, Latefa, and Olivier Cardin. 2017. "Determination of an Empirical Model of Average Rank for Multi-Deep AS/RS Based on Simulation." In *2017 Winter Simulation Conference (WSC)*, 3184–95. Las Vegas, NV: IEEE. <https://doi.org/10.1109/WSC.2017.8248037>.
- Ghomri, Latéfa, and Zaki Sari. 2015. "Mathematical Modeling of Retrieval Travel Time for Flow-Rack Automated Storage and Retrieval Systems." *IFAC-PapersOnLine* 48 (3): 1906–11. <https://doi.org/10.1016/j.ifacol.2015.06.365>.
- . 2017. "Mathematical Modeling of the Average Retrieval Time for Flow-Rack Automated Storage and Retrieval Systems." *Journal of Manufacturing Systems* 44 (July): 165–78. <https://doi.org/10.1016/j.jmsy.2017.05.002>.
- Glover, Fred. 1989. "Tabu Search—Part I." *ORSA Journal on Computing* 1 (3): 190–206.
- Gudehus, Timm. 1972. "Grundlagen Der Spielzeitberechnung Für Automatische Hochregallager." *Deutsche Hebe-Und Fördertechnik* 18: 63–68.
- Gue, Kevin R, and Byung Soo Kim. n.d. "Puzzle-Based Storage Systems," 27.
- Hachemi, Khalid, and Hassane Alla. 2008. "Piloteage dynamique d'un système automatisé de stockage/déstockage à convoyeur gravitationnel." *Journal Européen des Systèmes Automatisés* 42 (5): 487–508. <https://doi.org/10.3166/jesa.42.487-508>.
- Hachemi, Khalid, and Béatrix Besombes. 2013. "Integration of Products Expiry Dates in Optimal Scheduling of Storage/Retrieval Operations for a Flow-Rack AS/RS." *International Journal of Industrial and Systems Engineering* 15 (2): 216. <https://doi.org/10.1504/IJISE.2013.056097>.
- Hale, Trevor S., Michael E. Hanna, Faizul Huq, and Adrian Gil. 2015. "Closed Form Models for Dwell Point Locations in a Multi-Aisle Automated Storage and Retrieval System." *International Journal of Industrial and Systems Engineering* 19 (3): 364–88. <https://doi.org/10.1504/IJISE.2015.068202>.
- Hameed, Hanan M. n.d. "The Automatic Storage and Retrieval System: An Overview." *International Journal of Computer Applications* 177 (16): 8.
- Hamzaoui, Mohammed. 2020. "Investigations Sur l'AS/RS à Convoyeurs Gravitationnels Bidirectionnels."
- Hamzaoui, Mohammed Adel, and Zaki Sari. 2015. "Optimal Dimensions Minimizing Expected Travel Time of a Single Machine Flow Rack AS/RS." *Mechatronics* 31 (October): 158–68. <https://doi.org/10.1016/j.mechatronics.2014.10.006>.
- Hamzaoui, Mohammed, and Sari Zaki. 2020. "Cycle Time Models for the Bidirectional Flow-Rack AS/RS." *FME Transactions* 48 (2): 211–26. <https://doi.org/10.5937/fmet2001211H>.
- Han, Min-Hong, Leon F. McGinnis, Jin Shen Shieh, and John A. White. 1987. "On Sequencing Retrievals in an Automated Storage/Retrieval System." *IIE Transactions* 19 (1): 56–66.
- Hao, Jingjing, Yugang Yu, and Linda L. Zhang. 2015. "Optimal Design of a 3D Compact Storage System with the I/O Port at the Lower Mid-Point of the Storage Rack." *International Journal of Production Research* 53 (17): 5153–73. <https://doi.org/10.1080/00207543.2015.1005767>.
- Hatton, Gerry. 1990. "Designing a Warehouse or Distribution Centre." *The Gower Handbook of Logistics and Distribution Management (4 Th Ed.)*. Gower Publishing, Aldershot, 175–93.
- Hausman, Warren H., Leroy B. Schwarz, and Stephen C. Graves. 1976. "Optimal Storage Assignment in Automatic Warehousing Systems." *Management Science* 22 (6): 629–38.

References

- Heskett, James L., Nicholas A. Glaskowsky, and Robert M. Ivie. 1973. *Business Logistics: Physical Distribution and Materials Management*. Vol. 1. Ronald Press Company New York.
- Huq, Faizul, Trevor S. Hale, Heather S. Lutz, and Nikhil A. Pujari. 2018. "Closed Form Models for Dwell Point Locations with Turnover-Based Storage Assignment Policy." *International Journal of Industrial and Systems Engineering*, May. <https://www.inderscienceonline.com/doi/10.1504/IJISE.2018.091434>.
- Hwang, Hark, and Seong Beak Lee. 1990. "Travel-Time Models Considering the Operating Characteristics of the Storage and Retrieval Machine." *The International Journal of Production Research* 28 (10): 1779–89.
- Hwang, Hark, Young-keun Song, and Kap-hwan Kim. 2004. "The Impacts of Acceleration/Deceleration on Travel Time Models for Carousel Systems." *Computers & Industrial Engineering* 46 (2): 253–65.
- Jablonský, Josef. 2015. "Benchmarks for Current Linear and Mixed Integer Optimization Solvers." *Acta Universitatis Agriculturae et Silviculturae Mendelianae Brunensis* 63 (6): 1923–28.
- Karbstein, Marika. 2022. "Introduction to Modeling with Python."
- Kirkpatrick, Scott, C. Daniel Gelatt Jr, and Mario P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220 (4598): 671–80.
- Kluyver, Thomas, Ragan-Kelley Benjamin, Perez Fernando, Granger Brian, Bussonnier Matthias, Frederic Jonathan, Kelley Kyle, et al. 2016. *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. F. Loizides and B. Schmidt.
- Koster, R. (M.) B. M. de, T. Le-Duc, and Y. Yugang. 2008. "Optimal Storage Rack Design for a 3-Dimensional Compact AS/RS." *International Journal of Production Research* 46 (6): 1495–1514. <https://doi.org/10.1080/00207540600957795>.
- Koster, René de, Tho Le-Duc, and Kees Jan Roodbergen. 2007. "Design and Control of Warehouse Order Picking: A Literature Review." *European Journal of Operational Research* 182 (2): 481–501. <https://doi.org/10.1016/j.ejor.2006.07.009>.
- Kota, Venkat R., Don Taylor, and Kevin R. Gue. 2015. "Retrieval Time Performance in Puzzle-Based Storage Systems." *Journal of Manufacturing Technology Management* 26 (4): 582–602. <https://doi.org/10.1108/JMTM-08-2013-0109>.
- Kouloughli, Sihem, and Zaki Sari. 2015. "Multi-Aisle AS/RS Dimensions Optimization for Cycle Time Minimization." *The International Journal of Advanced Manufacturing Technology* 79 (1): 675–92.
- Kouloughli, Sihem, Zaki Sari, and Tewfik Sari. 2010. "Optimisation Des Dimensions d'un AS/RS Multi-Allée Basée Sur Un Modèle Analytique Du Temps de Simple Cycle." *Journal Européen Des Systèmes Automatisés (JESA)* 44 (2): 135–59.
- Lehmann, Timo, and Jakob Hußmann. 2022. "Travel Time Model for Multi-Deep Automated Storage and Retrieval Systems with Different Storage Strategies." *International Journal of Production Research*, August, 1–16. <https://doi.org/10.1080/00207543.2022.2110536>.
- . n.d. "Travel Time Model for Multi-Deep Automated Storage and Retrieval System with a Homogeneous Allocation Structure," 15.
- Lerher, T., M. Sraml, I. Potrc, and T. Tollazzi. 2010. "Travel Time Models for Double-Deep Automated Storage and Retrieval Systems." *International Journal of Production Research* 48 (11): 3151–72. <https://doi.org/10.1080/00207540902796008>.

References

- Lerher, Tone. 2016. "Travel Time Model for Double-Deep Shuttle-Based Storage and Retrieval Systems." *International Journal of Production Research* 54 (9): 2519–40. <https://doi.org/10.1080/00207543.2015.1061717>.
- Li, Li, and Zhuxi Chen. 2019. "Hungarian-Based Heuristics for Single-Machine Flow-Rack AS/RS with Determined Storage and Retrieval Locations." In *Proceedings of the 3rd International Conference on Computer Science and Application Engineering - CSAE 2019*, 1–7. Sanya, China: ACM Press. <https://doi.org/10.1145/3331453.3361669>.
- Liu, Debao, Shandong Mou, Meixiu Lu, and Yaohua Wu. 2014. "Travel Time Model of a New Compact Storage System." *International Journal of Control and Automation* 7 (7): 447–60. <https://doi.org/10.14257/ijca.2014.7.7.37>.
- Manzini, Riccardo, Riccardo Accorsi, Giulia Baruffaldi, Teresa Cennerazzo, and Mauro Gamberi. 2016. "Travel Time Models for Deep-Lane Unit-Load Autonomous Vehicle Storage and Retrieval System (AVS/RS)." *International Journal of Production Research* 54 (14): 4286–4304. <https://doi.org/10.1080/00207543.2016.1144241>.
- Marolt, Jakob, Nenad Kosanić, and Tone Lerher. 2021. "Relocation and Storage Assignment Strategies Evaluation in a Multiple-Deep Tier-Captive Automated Vehicles Storage and Retrieval System with Undetermined Retrieval Sequence." Preprint. In Review. <https://doi.org/10.21203/rs.3.rs-603627/v1>.
- Metahri, Dhiyaeddine, and Khalid Hachemi. 2017. "Automated Storage and Retrieval Systems: A Performances Comparison between Free-Fall-Flow-Rack and Classic Flow-Rack." In *2017 6th International Conference on Systems and Control (ICSC)*, 589–94. Batna, Algeria: IEEE. <https://doi.org/10.1109/ICoSC.2017.7958654>.
- . 2018. "Retrieval–Travel–Time Model for Free-Fall-Flow-Rack Automated Storage and Retrieval System." *Journal of Industrial Engineering International* 14 (4): 807–20. <https://doi.org/10.1007/s40092-018-0263-9>.
- Mohsen, and M. D. Hassan. 2002. "A Framework for the Design of Warehouse Layout." *Facilities* 20 (13/14): 432–40.
- Mueller, John, and Luca Massaron. 2016. *Machine Learning for Dummies*. For Dummies. Hoboken, New Jersey: John Wiley & Sons, Inc.
- Mulcahy, D. 1994. "Warehouse Distribution and Operations Handbook McGraw-Hill." NY.
- Nilsson, Nils J. 1998. "MLBOOK - INTRODUCTION TO MACHINE LEARNING." 1998. <https://www.coursehero.com/file/5996903/MLBOOK/>.
- Osisanwo, F. Y., J. E. T. Akinsola, O. Awodele, J. O. Hinmikaiye, O. Olakanmi, and J. Akinjobi. 2017. "Supervised Machine Learning Algorithms: Classification and Comparison." *International Journal of Computer Trends and Technology (IJCTT)* 48 (3): 128–38.
- Pan, C.-H., and C.-H. Wang. 1996. "A Framework for the Dual Command Cycle Travel Time Model in Automated Warehousing Systems." *International Journal of Production Research* 34 (8): 2099–2117.
- Randhawa, Sabah U., Edward D. McDowell, and Wen-Tsao Wang. 1991. "Evaluation of Scheduling Rules for Single-and Dual-Dock Automated Storage/Retrieval System." *Computers & Industrial Engineering* 20 (4): 401–10.
- Randhawa, Sabah U., and Raj Shroff. 1995. "Simulation-Based Design Evaluation of Unit Load Automated Storage/Retrieval Systems." *Computers & Industrial Engineering* 28 (1): 71–79.

References

- Roodbergen, Kees Jan, and Iris F.A. Vis. 2009. "A Survey of Literature on Automated Storage and Retrieval Systems." *European Journal of Operational Research* 194 (2): 343–62. <https://doi.org/10.1016/j.ejor.2008.01.038>.
- Royal Society (Great Britain). 2017. *Machine Learning: The Power and Promise of Computers That Learn by Example*.
- SARI, Z. 1998. "Expressions Analytiques Discretes de Simple et Double Cycle d'un AS/RS Multiallees." internal report of LAT, Tlemcen, Algeria.
- Sari, Z., S. E. Grasman, and N. Ghouali. 2007. "Impact of Pickup/Delivery Stations and Restoring Conveyor Locations on Retrieval Time Models of Flow-Rack Automated Storage and Retrieval Systems." *Production Planning & Control* 18 (2): 105–16. <https://doi.org/10.1080/09537280600909494>.
- Sari, Zaki. 2003. "MODELISATION, ANALYSE ET EVALUATION DES PERFORMANCES D'UN AS/RS A CONVOYEUR GRAVITATIONNEL."
- Sari, Zaki, and Nadir Hakim Bessenouci. n.d.-a. "Design & Modeling of a Single Machine Flow Rack As/Rs," 22.
- . n.d.-b. "Design & Modeling of a Single Machine Flow Rack As/Rs," 22.
- Sari, Zaki, Can Saygin, and Nouredine Ghouali. 2005. "Travel-Time Models for Flow-Rack Automated Storage and Retrieval Systems." *The International Journal of Advanced Manufacturing Technology* 25 (9–10): 979–87. <https://doi.org/10.1007/s00170-003-1932-3>.
- Seidmann, Abraham. 1988. "Intelligent Control Schemes for Automated Storage and Retrieval Systems." *The International Journal Of Production Research* 26 (5): 931–52.
- Sinha, Priyanka. 2013. "Multivariate Polynomial Regression in Data Mining: Methodology, Problems and Solutions" 4 (12): 4.
- Van Den Berg, Jeroen P. 1996. "Multiple Order Pick Sequencing in a Carousel System: A Solvable Case of the Rural Postman Problem." *Journal of the Operational Research Society* 47 (12): 1504–15.
- Wang, Jih-Yau, and Yuehwern Yih. 1997. "Using Neural Networks to Select a Control Strategy for Automated Storage and Retrieval Systems (AS/RS)." *International Journal of Computer Integrated Manufacturing* 10 (6): 487–95.
- Wen, U. P., D. T. Chang, and S. P. Chen. 2001. "The Impact of Acceleration/Deceleration on Travel-Time Models in Class-Based Automated S/R Systems." *IIE Transactions* 33 (7): 599–608.
- Xu, Xianhao, Yeming (Yale) Gong, Xiangxiang Fan, Guwen Shen, and Bipan Zou. 2018. "Travel-Time Model of Dual-Command Cycles in a 3D Compact AS/RS with Lower Mid-Point I/O Dwell Point Policy." *International Journal of Production Research* 56 (4): 1620–41. <https://doi.org/10.1080/00207543.2017.1361049>.
- Xu, Xianhao, Guwen Shen, Yugang Yu, and Wei Huang. 2015. "Travel Time Analysis for the Double-Deep Dual-Shuttle AS/RS." *International Journal of Production Research* 53 (3): 757–73. <https://doi.org/10.1080/00207543.2014.921351>.
- Xu, Xianhao, Xiaozhen Zhao, Bipan Zou, Yeming (Yale) Gong, and Hongwei Wang. 2020. "Travel Time Models for a Three-Dimensional Compact AS/RS Considering Different I/O Point Policies." *International Journal of Production Research* 58 (18): 5432–55. <https://doi.org/10.1080/00207543.2019.1659519>.

References

- Xu, Xianhao, Xiaozhen Zhao, Bipan Zou, and Mingze Li. 2019. "Optimal Dimensions for Multi-Deep Storage Systems under Class-Based Storage Policies." *Cluster Computing* 22 (3): 861–75. <https://doi.org/10.1007/s10586-018-2873-9>.
- Yang, Peng, Lixin Miao, Zhaojie Xue, and Lei Qin. 2015. "Optimal Storage Rack Design for a Multi-Deep Compact AS/RS Considering the Acceleration/Deceleration of the Storage and Retrieval Machine." *International Journal of Production Research* 53 (3): 929–43. <https://doi.org/10.1080/00207543.2014.942441>.
- Yang, Peng, Kaidong Yang, Mingyao Qi, Lixin Miao, and Bin Ye. 2017. "Designing the Optimal Multi-Deep AS/RS Storage Rack under Full Turnover-Based Storage Policy Based on Non-Approximate Speed Model of S/R Machine." *Transportation Research Part E: Logistics and Transportation Review* 104 (August): 113–30. <https://doi.org/10.1016/j.tre.2017.05.010>.
- Yu, Yugang, and René B.M. De Koster. 2012. "Sequencing Heuristics for Storing and Retrieving Unit Loads in 3D Compact Automated Warehousing Systems." *IIE Transactions* 44 (2): 69–87. <https://doi.org/10.1080/0740817X.2011.575441>.
- Yu, Yugang, and M.B.M. de Koster. 2009a. "Designing an Optimal Turnover-Based Storage Rack for a 3D Compact Automated Storage and Retrieval System." *International Journal of Production Research* 47 (6): 1551–71. <https://doi.org/10.1080/00207540701576346>.
- Yu, Yugang, and René B.M. de Koster. 2009b. "Optimal Zone Boundaries for Two-Class-Based Compact Three-Dimensional Automated Storage and Retrieval Systems." *IIE Transactions* 41 (3): 194–208. <https://doi.org/10.1080/07408170802375778>.
- Zaerpour, Nima, Yugang Yu, and René de Koster. 2017. "Small Is Beautiful: A Framework for Evaluating and Optimizing Live-Cube Compact Storage Systems." *Transportation Science* 51 (1): 34–51. <https://doi.org/10.1287/trsc.2015.0586>.

Resumé

Dans ce travail de thèse de doctorat, nous considérons les systèmes automatisés de stockage/déstockage (AS/RS) multi-profondeurs, c'est-à-dire les AS/RS où chaque casier est conçu pour contenir plusieurs charges. Plus particulièrement, nous étudions ici le cas des AS/RS à convoyeurs gravitationnels. Cette configuration d'AS/RS utilise deux machines, la première s'occupant des opérations de stockage et l'autre des opérations de déstockage. Dans ce type d'AS/RS, les opérations de déstockage sont assez complexes, puisque le déstockage d'une charge nécessite le retrait et le remplacement de toutes les charges qui la précèdent dans le casier. Pour cette raison, il est préférable de considérer le retrait de lots de charges plutôt que de considérer le retrait de charges une par une. Le problème qui nous intéresse ici consiste à considérer une disposition de charges dans un rack profond et un lot de types de charges à déstocker (une commande). Nous allons d'abord, localiser dans le rack les charges à récupérer (pour satisfaire la commande), puis définir la bonne séquence d'opérations de déstockage ; dans le but de minimiser le temps global de déstockage du lot. Le problème que nous nous proposons d'étudier ici est assez particulier et ne ressemble à aucun des problèmes classiques que l'on trouve dans le domaine du génie industriel. La première contribution de ce travail de thèse est la modélisation mathématique du problème décrit. La seconde contribution est une heuristique permettant de résoudre le modèle mathématique proposé ? Nous baptisons cette heuristique : heuristique de minimisation du temps de déstockage par lot (BRTM-heuristique). Nous adaptons également les deux métaheuristiques : le recuit simulé et la recherche tabou à la résolution du problème. Les résultats de l'heuristique BRTM seront comparés aux résultats donnés par les deux métaheuristiques afin de valider la précision de l'heuristique BRTM.

Mots clés: Systèmes automatisés de stockage/déstockage ; temps de déstockage de lots ; optimisation ; modélisation mathématique ; heuristiques ; métaheuristiques.

Abstact

In this work of doctoral thesis, we consider multi-deep automated storage and retrieval systems (AS/RS), i.e., AS/RS where each bin is designed to contain several goods. More particularly, the case of flow-rack AS/RS is studied here. This configuration of AS/RS uses two cranes, the first one deals with storage operations and the other with the retrieval operations. In this type of AS/RS, the retrieval operations are quite complex, since retrieving a load requires the withdrawal and the reallocation of all the loads preceding it in the bin. For this reason, it is preferable to consider the retrieving of batches of loads rather than considering the retrieving of loads one by one. The problem we are interested in here consists in considering an arrangement of loads in a deep rack and a batch of load types to be retrieved (an order); we will first, locate in the rack the loads to be retrieved (to satisfy the order), then define the right sequence of retrieval

operations; for the aim of minimizing the overall batch retrieval time. This problem we propose to study here is quite particular and does not look like any of the conventional problems found in the field of industrial engineering. The first contribution of this work is the mathematical modeling of the described problem. The second contribution is a heuristic allowing to solve the proposed mathematical model, we call it batch retrieval time minimization heuristic (BRTM-heuristic). We also adapt the two metaheuristics: simulated annealing (SA) and tabu search (TS) to the problem resolution. The results of the BRTM-heuristic will be compared with the results given by both metaheuristics in order to validate the accuracy of the BRTM-heuristic.

Keywords: Automated storage/retrieval systems; batch retrieval time; optimization; mathematical modeling; heuristics; metaheuristics.

ملخص

في أطروحة الدكتوراه هذه، ندرس أنظمة التخزين/الاسترجاع الآلية متعددة الأعماق، أي أنظمة التخزين/الاسترجاع الآلية حيث تم تصميم كل رف ليحتوي على حمولات متعددة. على وجه الخصوص، ندرس هنا حالة أنظمة التخزين/الاسترجاع الآلية بواسطة ناقلات الجاذبية. يستخدم هذا النظام الآلي جهازين، الأول يتعامل مع عمليات التخزين والآخر مع عمليات الاسترجاع. وفي هذا النوع من أنظمة التخزين/الاسترجاع الآلي، تكون عمليات إزالة المخزون معقدة للغاية، لأن إزالة الحمولة تتطلب إزالة واستبدال جميع الحمولات التي تسبقها في الخزانة. ولهذا السبب، يفضل النظر في إزالة عدة حمولات (طلب) بدلاً من إزالة حمولة الواحدة تلو الأخرى. المشكلة التي نلقينا هنا هي التفكير في ترتيب الحمولات في رف عميق والكثير من أنواع الحمولات التي يجب استرجاعها (طلب). سنقوم أولاً بتحديد موقع الأحمال التي سيتم استردادها في الرف (لتلبية الطلب)، ثم تحديد التسلسل الصحيح لعمليات الاسترجاع؛ من أجل التقليل إلى أدنى حد من الوقت الإجمالي لاسترجاع الطلب. المشكلة التي نقترح دراستها هنا خاصة إلى حد ما ولا تشبه أيًا من المشاكل الكلاسيكية الموجودة. المساهمة الأولى في هذه الأطروحة هي النمذجة الرياضية للمسألة الموصوفة. المساهمة الثانية هي استخلاص لحل النموذج الرياضي المقترح؟ نسمي هذه الطريقة: طريقة التقليل من الوقت الإجمالي لاسترجاع الطلب (الحمولات). BRTM. نقوم أيضًا بتكييف طريقتين للميتاهوريستيكس من أجل مقارنة نتائج طريقة BRTM بالنتائج التي قدمتها ميتاهوريستيكس من أجل التحقق من دقة طريقتنا.

الكلمات المفتاحية: أنظمة التخزين/الاسترجاع الآلية متعددة الأعماق، الوقت الإجمالي لاسترجاع الطلب (الحمولات)، تحسين، النمذجة الرياضية، الطريقة، الميتاهوريستيكس.