

République Algérienne Démocratique et Populaire  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Abderrahmane Mira de Béjaia



جامعة بجاية  
Tasdawit n'Bgayet  
Université de Béjaïa

Faculté des Sciences Exactes  
Département de Recherche Opérationnelle

## Mémoire de fin de cycle

Pour l'Obtention du Diplôme de Master en Mathématique Appliquée

Option :

Modélisation mathématique et évaluation des  
performances des réseaux

Thème :

*Problème de Stockage des Conteneurs*

AU NIVEAU DE L'ENTREPRISE *BMT*

Réalisé par :

*Mr Mohammed Lamine KHAIRONE*

Soutenu le 19 juin 2019 devant le jury composé de :

<b>Président :</b>	M <sup>me</sup> . AOUDIA Fazia	Université de Béjaia
<b>Examinatrice :</b>	M <sup>me</sup> . ANZI Aicha	Université de Béjaia
	M <sup>me</sup> . HALIMI Naouel	Université de Béjaia
<b>Encadreur :</b>	M <sup>lle</sup> . AOUDIA Zohra	Université de Béjaia
<b>Invités :</b>	M <sup>r</sup> . ZIDANE Mbarek	Chef de Département Opérationnelle BMT
	M <sup>r</sup> . BOUMERZOUG Moussa	Chef service informatique BMT

*Promotion 2018-2019*

---

---

# Remerciements

---

Toute notre gratitude, grâce et remerciement vont à dieu le tout puissant qui nous a donné la force et la patience, le courage et la volonté pour élaborer ce travail.

Nos remerciements les plus chaleureux et les plus vifs s'adressent d'abord à notre honorable promotrice *M<sup>lle</sup> Zohra AOUDIA* qui nous a aidé à réaliser ce travail.

Nous remercions les membres de jury qui nous font l'honneur de présider et d'examiner ce modeste travail.

Nous tenons également à exprimer notre gratitude envers tous les enseignants de département Recherche Opérationnelle à l'Université Abderrahmane Mira qui ont contribué à notre formation et à l'élaboration de ce travail.

Nous tenons une place particulière aux personnels de l'entreprise *BMT*, en particulier Mr *M. BOUMERZOUG* le chef de service informatique et Mr *M. ZIDANE* chef des opérations pour leurs conseils et leurs orientation durant notre stage.

Enfin, nous adressons nos plus sincères remerciements à tous nos proches et amis, qui nous ont toujours encouragées au cours de la réalisation de ce mémoire.

*Merci à vous tous.*

---

---

# Dédicace

---

*Je dédie ce modeste travail à :*

*À mes chers parents qui m'ont élevé,*

*À ma sœur, mes frères et ma famille qui m'ont soutenu,*

*À ma fiancée et sa famille,*

*À mes amis qui ont été toujours à mes côtés.*

*M<sup>r</sup> Mohammed Lamine KHAIROUNE*

# Table des matières

<b>Table des matières</b>	<b>i</b>
<b>Table des figures</b>	<b>iii</b>
<b>Introduction générale</b>	<b>1</b>
<b>1 État de l'art</b>	<b>5</b>
Introduction . . . . .	5
1.1 La conteneurisation . . . . .	6
1.1.1 Description et propriétés du conteneur . . . . .	6
1.1.2 Histoire de la conteneurisation . . . . .	6
1.2 Description des ports et des terminaux à conteneurs . . . . .	7
1.2.1 Description et rôles des terminaux à conteneurs . . . . .	8
1.2.2 Les équipements de manutention . . . . .	8
1.2.3 Les véhicules internes de transfert . . . . .	10
1.2.4 Configurations de la cour de stockage . . . . .	10
1.3 Le problème de stockage de conteneurs . . . . .	12
1.4 Les méthodes de résolution . . . . .	13
Conclusion . . . . .	14
<b>2 Rappels théoriques et outils</b>	<b>15</b>
2.1 Les problèmes d'optimisation combinatoire . . . . .	15
2.1.1 Les problèmes faciles . . . . .	15
2.1.2 Les problème difficiles . . . . .	16
2.2 Modélisation des problèmes d'optimisation combinatoire . . . . .	16
2.2.1 La programmation linéaire . . . . .	16
2.2.2 La programmation linéaire et non-linéaire en nombres entiers . . . . .	16
2.2.3 La théorie des graphes . . . . .	17
2.3 Les méthodes de résolution . . . . .	17
2.3.1 Les méthodes exactes . . . . .	17
2.3.2 Les méthodes approchées . . . . .	17
2.4 La programmation linéaire en nombres entiers . . . . .	17
2.4.1 Les méthodes exactes . . . . .	18
2.4.2 Les méthodes approchées . . . . .	19
2.5 Le calcul informatique . . . . .	23
2.5.1 Microsoft Excel . . . . .	24

2.5.2	Visual Basic for Application <i>VBA</i> . . . . .	25
	Conclusion . . . . .	25
<b>3</b>	<b>Modélisation et résolution du problème</b>	<b>26</b>
	Introduction . . . . .	26
3.1	Présentation d'entreprise d'accueil BMT SPA . . . . .	26
3.1.1	Définition de BMT . . . . .	26
3.1.2	Structure organisationnelle de BMT . . . . .	27
3.1.3	Le terminal à conteneurs . . . . .	27
3.1.4	Les outils de gestion du terminal . . . . .	29
3.1.5	Les procédures import/export de BMT . . . . .	29
3.1.6	Quelques services de BMT . . . . .	30
3.1.7	Les objectifs de BMT . . . . .	31
3.1.8	Position du problème . . . . .	32
3.2	Le problème de sélection des conteneurs à récupérer . . . . .	32
3.2.1	Présentation de problème . . . . .	32
3.2.2	La formulation mathématique du problème . . . . .	32
3.3	La résolution du problème de sélection de conteneur à récupérer . . . . .	33
3.3.1	Codage utilisé pour notre algorithme . . . . .	33
3.3.2	Génération d'une population initiale . . . . .	34
3.3.3	Opérateur de sélection . . . . .	34
3.3.4	Opérateur de croisement . . . . .	35
3.3.5	Opérateur de mutation . . . . .	35
3.3.6	Opérateur d'insertion . . . . .	35
3.3.7	Fonction d'évaluation (fitness) . . . . .	35
3.3.8	Critère d'arrêt . . . . .	36
3.4	Le Problème de Rélocalisation des Conteneurs (PRC) . . . . .	36
3.4.1	Présentation du problème . . . . .	37
3.4.2	La formulation mathématique du problème . . . . .	38
3.5	La résolution de PRC . . . . .	47
	Conclusion . . . . .	49
<b>4</b>	<b>Analyse et interprétation des résultats</b>	<b>51</b>
	Introduction . . . . .	51
4.1	Description de notre outil informatique . . . . .	51
4.1.1	Les indices utilisés . . . . .	51
4.1.2	Les tableaux de donnée utilisées . . . . .	53
4.1.3	L'interface homme-machine (IHM) . . . . .	54
4.2	Collecte de données . . . . .	58
4.2.1	Les données du générateur que nous avons construit . . . . .	58
4.3	Interprétation des résultats . . . . .	59
	Conclusion . . . . .	61
	<b>Conclusion générale et perspective</b>	<b>62</b>



# Table des figures

1.1	Différentes formes de conteneurs	6
1.2	Grue de quai	9
1.3	Rail-Mounted Gantry Crane (RMGC)	9
1.4	Rubber-Tyred Gantry Crane (RTGC)	10
1.5	Terminal qui utilise des RTGCs	11
1.6	Terminal qui utilise des RMGCs	12
2.1	Croisement cellulaire	19
2.2	Mutation cellulaire	20
2.3	Croisement en un point	21
2.4	Croisement en deux points	21
2.5	Logo de Microsoft Excel	24
3.1	Les partenariats de BMT	27
3.2	Codage d'un chromosome	34
3.3	Exemple qui montre une rue avec 6 piles, 4 niveaux et 16 conteneurs	38
3.4	Exemple illustre la stratégie de remplissage	49
4.1	Tableau de bord de projet	53
4.2	La table de données <b>tblDataBloc</b>	53
4.3	La table de données <b>tblEtatEmplacement</b>	54
4.4	L'interface du générateur	55
4.5	L'interface ( <b>État des emplacements</b> )	55
4.6	L'interface ( <b>Les données du problème</b> )	56
4.7	L'interface ( <b>Les paramètres de l'algorithme génétique</b> )	56
4.8	L'interface ( <b>La liste des mouvements à faire</b> )	57
4.9	L'interface ( <b>Tableau de bord d'évaluation du projet</b> )	57
4.10	L'organigramme du générateur	58
4.11	La première application	59
4.12	La deuxième application	59
4.13	La troisième application	59
4.14	Tableau de bord d'évaluation de la première application	60
4.15	Tableau de bord d'évaluation de la deuxième application	61
4.16	Tableau de bord d'évaluation de la troisième application	61

# Introduction générale

La conteneurisation des marchandises a joué un rôle important dans le développement des réseaux internationaux de transport inter-modal. En effet, avec l'arrivée du conteneur depuis les années 50 les opérations de transfert d'un mode de transport à un autre demeurent l'élément clé d'un système de transport performant. Parmi ces points de transfert, les terminaux portuaires de conteneurs sont généralement identifiés comme un maillon important de la chaîne logistique globale de bout en bout.

Les terminaux à conteneurs constituent des interfaces inter-modales essentielles pour le réseau de transport mondial. La compétitivité d'un terminal est principalement reflétée par son efficacité de transbordement à cause des charges payées par un bateau qui dépendent du temps de rotation et du nombre de conteneurs chargés et déchargés à un terminal. Le terminal à conteneurs est généralement exploité avec trois types d'équipements :

- des grues de quai qui assurent le chargement et le déchargement des navires,
- des camions de cour (véhicules auto guidés, cavaliers, etc.) qui assurent le transport des conteneurs du quai vers les zones de stockage et vice versa,
- des grues de cour qui assurent le stockage des conteneurs dans les zones de stockage.

En effet, une manutention efficace des conteneurs dans les terminaux est d'une importance cruciale pour la réduction des coûts de transport et la détermination des plans d'embarquement. De ce fait, il est nécessaire de construire de nouveaux terminaux portuaires ou, au moins d'améliorer les capacités des anciens. De plus, les infrastructures des terminaux existants vont également s'accroître. Le problème qui se pose est de savoir si les terminaux existants sont assez performants pour manutentionner des flux plus importants de conteneurs ou si les équipements existants doivent être remplacés par d'autres plus efficaces. Dans le but de répondre à ces questions, Il existe de nombreuses études sur l'optimisation des opérations dans un terminal à conteneurs utilisant des nouvelles techniques de la recherche opérationnelle[14]. Ces études portent généralement sur des problèmes spécifiques tels que l'ordonnancement d'un type d'équipement, l'affectation de navires aux quais ou la gestion des espaces de stockage dans le but d'optimiser au maximum le trafic de conteneurs [9].

Les problèmes opérationnels des terminaux à conteneurs ont été divisés en plusieurs sous problèmes, comme l'assignation des navires, planification d'arrivée, planification des grues de quai, la planification des cavaliers de cour, planification des grues de cour et l'assignation de stockage des conteneurs.

L'organisation de la zone de stockage de conteneurs du terminal requiert une attention particulière. En effet, le respect des contraintes de temps de chargement des navires, des camions et des trains dépend fortement de l'emplacement des conteneurs dans le terminal. Les temps de recherche des conteneurs sont parfois considérables et entraînent des retards importants provoquant des pénalités financières pour les sociétés gérant les terminaux. Afin de pouvoir gérer efficacement cette zone, il est nécessaire de connaître l'emplacement optimal de chaque conteneur. Notre travail dans ce mémoire traite précisément ce problème.

D'une manière générale, un terminal à conteneurs est un système complexe qui englobe des interactions hautement dynamiques entre des informations parfois incomplètes sur des événements futures et différents équipements de manutention, de transport et de stockage. Ainsi, il existe plusieurs problèmes de décision concernant les terminaux à conteneurs maritimes. Ils peuvent être regroupés en deux grandes catégories : les problèmes de conception et les problèmes de planification. La résolution des problèmes de conception se font généralement avant la construction d'un terminal à conteneurs. Ces problèmes doivent être analysés aussi bien sur le plan économique que sur le plan technique[17].

Dans ce mémoire, nous nous intéressons au problème de stockage de conteneurs. Ce dernier consiste à déterminer des emplacements de stockage adéquats pour les conteneurs qui arrivent dans un terminal et qui y séjournent temporairement. Une résolution efficace de ce problème nécessite la prise en considération des contraintes réelles du terminal, telles que le type d'équipement de transfert et de manutention utilisé ainsi que la configuration de la cour de stockage. Outre ces contraintes, il est aussi important d'assurer l'accessibilité de chaque conteneur au moment de sa date de départ, car des opérations supplémentaires peuvent être nécessaires pour déplacer les objets encombrants si tel n'est pas le cas. Cette contrainte n'est pas rigoureusement traitée dans la plupart des travaux rencontrés dans la littérature, qui proposent des stockages par groupe, en affectant des groupes de conteneurs à des groupes d'emplacements de stockage. Cette méthode peut fonctionner dans le cas des conteneurs qui ont les mêmes dates de départ et qui partent par le même train ou bien le même navire, mais elle ne convient pas aux conteneurs qui partent par camion. L'un des rares travaux qui ont traité la contrainte d'accessibilité des conteneurs tout en déterminant un emplacement de stockage précis pour chacun d'eux a proposé de les placer dans chaque pile suivant l'ordre décroissant de leurs dates de départ et d'interdire toute sorte de réarrangement, [16]. Cette solution ne convient pas aux terminaux à conteneurs qui reçoivent simultanément des quantités très importantes de conteneurs, qui ont des surfaces de stockage limitées et des dates de départ variables, raison pour laquelle, nous proposons des méthodes de résolution du problème de stockage de conteneurs qui déterminent un emplacement de stockage précis pour chaque conteneur tout en minimisant les remaniements (voire même les éliminer complètement dans le cas où la grandeur de la cour de stockage le permet). En plus de cela, nous minimisons aussi la distance totale à parcourir par les véhicules de transfert RTGC.

Pour mieux rentabiliser l'utilisation de conteneurs, des tentatives d'amélioration sont conti-

nuellement menées dans les terminaux. Cela a soulevé des questions telles que la nécessité d'augmenter les capacités de stockage, l'instauration de moyens de réduction des retards, l'utilisation de techniques efficaces pour assurer la sécurité des ports et des biens. Des réponses satisfaisantes à ces questions s'imposent pour que les ports maritimes puissent faire face à l'accroissement du secteur industriel à l'échelle mondiale et à l'apparition de nouvelles puissances économiques. En effet, avec la mondialisation, les besoins d'importation et d'exportation de conteneurs ne cessent de croître. Cela a conduit à la création de très grands navires qui peuvent transporter des nombres très élevés de conteneurs. Ainsi, le nombre de conteneurs qui doivent être traités par unité de temps dans un port augmente également. Pour faire face à ce phénomène, deux possibilités de résolution existent :

- La première consiste à effectuer des travaux d'agrandissement du port, notamment de la zone de stockage. Les conséquences directes de cette option sont les perturbations de fonctionnement durant les périodes de travaux. En plus de cela, elle nécessite l'investissement d'une quantité non négligeable de capitaux dont la rentabilité n'est pas garantie à cause des concurrences qui existent entre les ports, et les changements économiques.
- La deuxième option consiste à trouver des moyens de gestion qui augmentent la productivité du port. Cette solution semble moins risquée que la précédente, mais elle nécessite des efforts intellectuels permanents.

Notre contribution porte sur cette dernière méthode d'amélioration. Ce choix est fait en se basant sur les avantages que comporte l'action de rendre plus performantes les activités portuaires par rapport à celui de réaliser des travaux d'agrandissement de l'espace de stockage, car ce dernier aussi grand soit-il ne peut être productif s'il n'est pas bien géré. À cela s'ajoute le fait que choisir la première option revient à réaliser des travaux d'agrandissement à chaque fois que la quantité de conteneurs gérés augmente, ce qui correspond à un risque de perturbations répétitives sur le fonctionnement du port et aussi à des dépenses dont la rentabilité n'est pas garantie.

L'accroissement continu des nombres de conteneurs qui arrivent simultanément dans un port rend indispensable l'utilisation de moyens efficaces de gestion de l'espace de stockage. Il est évident que les méthodes de planification manuelles de stockage de conteneurs qui ne sont basées que sur les expériences des ouvriers et des cadres nécessitent beaucoup de temps et de labeur lorsque le nombre de conteneurs est très grand. Ainsi, elles doivent être remplacées par des logiciels ou bien des algorithmes efficaces et rapides.

Pour répondre à ces besoins, nous proposons dans ce mémoire, des systèmes d'aide à la décision qui ont pour objectif de faciliter la gestion de quantités importantes de conteneurs en fournissant en des temps raisonnables de très bons plans de stockage. Deux problèmes sont distingués : *la sélection des conteneurs* et *la relocalisation des conteneurs*. Pour chacun d'eux, une étude analytique est d'abord menée afin d'étudier le contexte, d'identifier les contraintes physiques et opérationnelles, et de proposer un modèle mathématique de stockage de conte-

neurs. Après cela, des algorithmes de résolution sont proposés.

Afin de bien présenter notre travail, nous allons l'organiser comme suit :

- Dans le premier chapitre nous présente un historique de la conteneurisation, les différents types de terminaux à conteneurs portuaires, et nous décrivons le problème de stockage de conteneurs proprement dit. Les différentes stratégies de stockage qui sont proposées dans la littérature sont présentées dans ce chapitre, de même que des méthodes de résolution analytiques et des méthodes de résolution numériques.
- Le deuxième chapitre, est consacré à la présentation d'une description de l'optimisation combinatoire, une description des principales méthodes d'optimisation utilisées dans la littérature et le calcul informatique.
- Le troisième chapitre, consiste à présenter l'entreprise d'accueil *BMT*, la modélisation de notre problème et à proposer des modèles d'aide de décision.
- Dans le quatrième et dernier chapitre nous présentons brièvement notre solveur en langage VBA et nous y exposons interprétations des résultats du modèle à optimiser.

# 1

## État de l'art

### Introduction

La conteneurisation est le fait d'utiliser des conteneurs comme moyen de transfert de biens et de marchandises. Ce concept n'est apparu qu'au  $XX^{\text{ème}}$  siècle (années 50), mais depuis lors il est devenu un élément indispensable dans le domaine du transport[4].

La cour de stockage est une ressource capitale dans un terminal à conteneurs. En effet, le degré d'efficacité de sa gestion se répercute dans la productivité globale du port. Suite à la baisse des frais de transport de marchandises et de biens qu'a occasionné la conteneurisation, les flux de conteneurs ont largement augmenté dans les ports[15]. La conséquence directe de ceci est une croissance vertigineuse du nombre de conteneurs qui séjournent simultanément dans un port, rendant ainsi insuffisants les emplacements de stockage au sol. Une solution à ce problème pourrait être l'augmentation des surfaces des cours de stockage [3, 2], mais elle s'avère inefficace car les nombres de conteneurs qui entrent simultanément dans un port croissent continuellement en fonction du développement des moyens de transport (exemple : les navires porte-conteneurs).

Dans ce chapitre nous présentons un aperçu général sur la conteneurisation, les problèmes de stockage de conteneurs.

## 1.1 La conteneurisation

### 1.1.1 Description et propriétés du conteneur

Un conteneur est une caisse métallique rectangulaire qui sert à emmagasiner des éléments qui doivent être transportés d'un endroit à un autre. Grâce à la standardisation, les dimensions des conteneurs sont réglementées par la norme *ISO 668 : 1995*. L'unité de mesure d'un conteneur est l'équivalent *20 pieds* (EVP), mais il existe des conteneurs de taille *40 pieds* (2 EVP), etc. Généralement, il y a plusieurs formes de conteneur, dont les plus connues sont : STANDARD, HIGH-CUBE, HARD-TOP, OPEN-TOP, FLATRACKS, PLATFORMS, VENTILATED, INSULATED AND REFRIGERATED, BULK, et TANK. La Figure 1.1 illustre ces différents types de conteneurs[17].



FIGURE 1.1: Différentes formes de conteneurs

### 1.1.2 Histoire de la conteneurisation

Le conteneur inter-modal est une boîte en acier dont les dimensions sont standardisées afin de pouvoir l'utiliser pour transporter des marchandises par bateau, camion, train, et avion. Il s'agit d'une technologie dont l'utilisation a un impact social et économique. Cet impact a été initialement estimé par les entrepreneurs de l'industrie maritime, notamment les autorités portuaires et les compagnies maritimes. Mais, comme pour de nombreuses technologies extrêmement efficaces, l'impact du conteneur s'est élargi et affecte le développement mondial.

## Impact du conteneur sur d'autres pratiques en matière de technologie et d'affaires

La nature du travail à quai a évidemment changé puisque la manipulation des conteneurs exige des compétences plus techniques pour l'utilisation des machines. La normalisation de la taille des conteneurs et des outils de manutention signifie que le même équipement de manutention de cargaison pourrait être utilisé pour une grande variété de marchandise. De plus, les navires pourraient être conçus dès le départ pour transporter des conteneurs. L'incertitude dans le transport maritime a été également réduite, car il est plus facile pour une entreprise d'expédition de calculer la vitesse de chargement ou de déchargement des conteneurs que celle de petits colis.

Ces progrès ont favorisé l'accroissement des investissements dans des navires et des compagnies maritimes depuis les années 1970 ([4] : pages 72-76), et aussi la création de navires de plus en plus spacieux. De plus, l'augmentation de l'ampleur des opérations des compagnies maritimes a favorisé l'intégration de transports terrestres. Un exemple clé est la compagnie *Sea-Land*. Elle était au début une entreprise de camionnage, et par la suite, elle s'est fusionnée avec la compagnie ferroviaire *CSX* en 1986 ([6] : pages 160-166).

### Impact global du conteneur

L'impact le plus profond du conteneur est se répercute sur l'économie mondiale dans son ensemble. Dans le monde entier, au début des années 2000, 300 millions de conteneurs de 20 pieds ont été transportés par voie maritime chaque année, avec plus d'un quart de ces expéditions en provenance de Chine ([13] : pages 277). Selon Slack [20] : • ***La mondialisation et la conteneurisation entretiennent une relation réciproque. Il y a peu de doute que l'expansion du commerce mondial et l'expansion des systèmes de fabrication mondiale auraient été impossible sans l'efficacité et les économies que la conteneurisation a apporté.***

## 1.2 Description des ports et des terminaux à conteneurs

Les ports jouent des rôles multiples dans l'industrie maritime et font partie d'un réseau complexe. Ce sont des interfaces reliant la mer et le transport terrestre. Cependant, il existe un grand nombre de définitions de port. Par conséquent, nous allons en citer une celle donnée par Stopfords ([21] : page 81) qui caractérise un port par :

- ***Une zone géographique où les navires sont mis à côté de la terre ferme afin d'y charger et décharger des marchandises - habituellement une zone d'eau profonde comme une baie ou l'embouchure d'un fleuve*** •.

Cette définition est assez simple, mais elle donne un aperçu sur le rôle fondamental d'un port. En même temps, il est important d'en comprendre que le rôle d'un port est plus complexe que

juste le fait d'être un emplacement en bord de mer. Aujourd'hui, les ports sont des éléments clés dans le système du transport mondial, car sans leur existence les navires seraient inutiles dans le transport de marchandises, puisqu'ils n'y auraient aucun endroit où ils pourraient être chargés ou déchargés. Par conséquent, les ports peuvent être considérés comme des facilitateurs du commerce multi-modal (entre terre et mer).

### 1.2.1 Description et rôles des terminaux à conteneurs

D'une façon générale, un terminal à conteneurs est une installation où les conteneurs sont transbordés entre différents moyens de transports.

Le transbordement peut être fait entre navires et véhicules terrestres (camions ou trains), dans ce cas, le terminal est décrit comme étant un terminal maritime. Alternativement, le transbordement peut être fait entre deux véhicules terrestres, généralement entre trains et camions, dans ce cas le terminal est décrit comme étant un terminal intérieur.

Les principales fonctions d'un terminal à conteneurs maritime sont : la réception, le stockage, la préparation, et le chargement de conteneurs entrants (qui arrivent au port par voie maritime, et qui quittent par voie terrestre) ou sortants (qui arrivent au port par voie terrestre et quittent par bateau)[17].

### 1.2.2 Les équipements de manutention

Généralement, des grues sont utilisées pour effectuer les opérations de manutention. Ces dernières sont principalement réparties entre des grues de quai et des grues de cour.

Les grues de quai servent à charger et à décharger les navires porte-conteneurs. Généralement, le déchargement est réalisé avant le chargement. Plusieurs grues de quai peuvent s'occuper simultanément d'un bateau, mais ils doivent suivre un plan de travail bien déterminé qui assure l'équilibre du navire. Généralement, ces genres de grue circulent sur des rails comme le montre la figure 1.2.



FIGURE 1.2: Grue de quai

Quant aux grues de cour, elles servent à placer (ou à enlever) des conteneurs dans (ou de) la cour de stockage. Ils existent généralement deux types de grue de cour : les Rail-Mounted Gantry Cranes (RMGCs) et les Rubber-Tyred Gantry Cranes (RTGCs).

- Les RMGCs sont automatisées et ne nécessitent pas de conducteurs. Elles circulent sur des rails, et sont généralement plus rapides que les RTGCs. La figure 1.3 est un exemple de RMGC.



FIGURE 1.3: Rail-Mounted Gantry Crane (RMGC)

- Les RTGCs (voir Figure 1.4) ont des pneus en caoutchouc, ce qui leurs donne la flexibilité

de circuler librement à l'intérieur de la cour de stockage. Elles sont capables de faire des rotations de  $90^\circ$  pour effectuer des mouvements orthogonaux, mais ce genre de mouvement nécessite du temps et dure environ 15 minutes. Contrairement à elles, les RMGCs ne peuvent circuler que sur des voies ferrées, mais leur principal avantage est leur rapidité car elles sont capables de parcourir  $300m$  en  $1min15sec$ . En plus de cela, elles peuvent être utilisées à tout moment (sauf lorsqu'elles sont en panne ou bien en maintenance), contrairement aux RTGCs qui ne sont opérationnelles qu'en présence de main d'œuvre.



FIGURE 1.4: Rubber-Tyred Gantry Crane (RTGC)

### 1.2.3 Les véhicules internes de transfert

Différents types de véhicules sont utilisés pour effectuer les transferts de conteneurs à l'intérieur d'un terminal à conteneurs : des camions, des voitures autoguidés, et des cavaliers gerbeurs.

### 1.2.4 Configurations de la cour de stockage

Dans les terminaux à conteneurs, les espaces de stockage sont constitués de plusieurs blocs. Cependant, les configurations des blocs diffèrent en fonction des équipements de stockage utilisés. On distingue principalement deux types de configuration : le *modèle compact* que l'on retrouve dans les terminaux à conteneurs qui utilisent des grues de cour, et le *modèle linéaire* que l'on rencontre dans les terminaux à conteneurs qui se servent de cavaliers gerbeurs.

Dans le modèle compact, il n'y a pas d'espaces de séparation prévus entre les piles adjacentes. Un bloc de ce genre est donc constitué de plusieurs rangées qui sont collées les unes aux autres. Chaque rangée est composée de plusieurs travées (rues), qui contiennent à leur tour des piles dans lesquelles sont superposés des conteneurs.

La disposition de ces blocs par rapport aux quais dépend de la nature des grues de cour utilisées. Dans le cas des terminaux à conteneurs qui ont opté pour des grues non-automatisées (RTGCs), les blocs sont disposés parallèlement aux quais. Dans ces blocs, une ou plusieurs rangées, appelée(s) voie(s) de camion, est (sont) réservée(s) à la circulation des véhicules de transfert. De ce fait, ces véhicules circulent dans ces espaces et s'arrêtent devant les travées (rues) souhaitées. Ainsi, les grues se déplacent jusqu'à leurs positions pour effectuer des chargements ou des déchargements. La figure 1.5 en est une illustration. On retrouve ce genre de disposition dans la plupart des ports asiatiques[20].

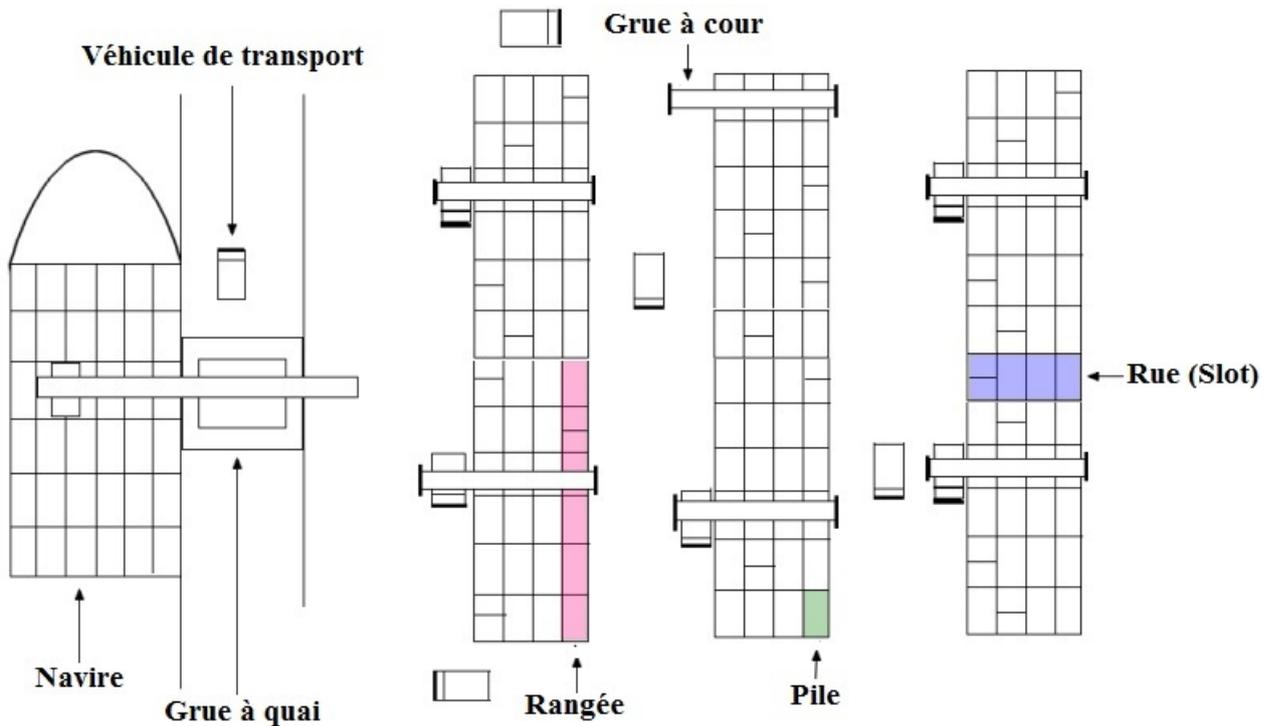


FIGURE 1.5: Terminal qui utilise des RTGCs

Dans le cas des terminaux à conteneurs qui utilisent des grues de quais automatisées (RMGCs), les blocs de stockage sont perpendiculaires aux quais. Les échanges entre les véhicules de transfert et les RMGCs se font dans deux zones d'échange spécifiques, situées de part et d'autre de la cour de stockage. Celle qui est en face des quais est le lieu d'interaction entre les véhicules autoguidés (AGVs) et les RMGCs. Tandis que les échanges entre les camions externes et les RMGCs se font à l'autre côté. La figure 1.6 est une représentation d'un tel terminal à conteneurs.

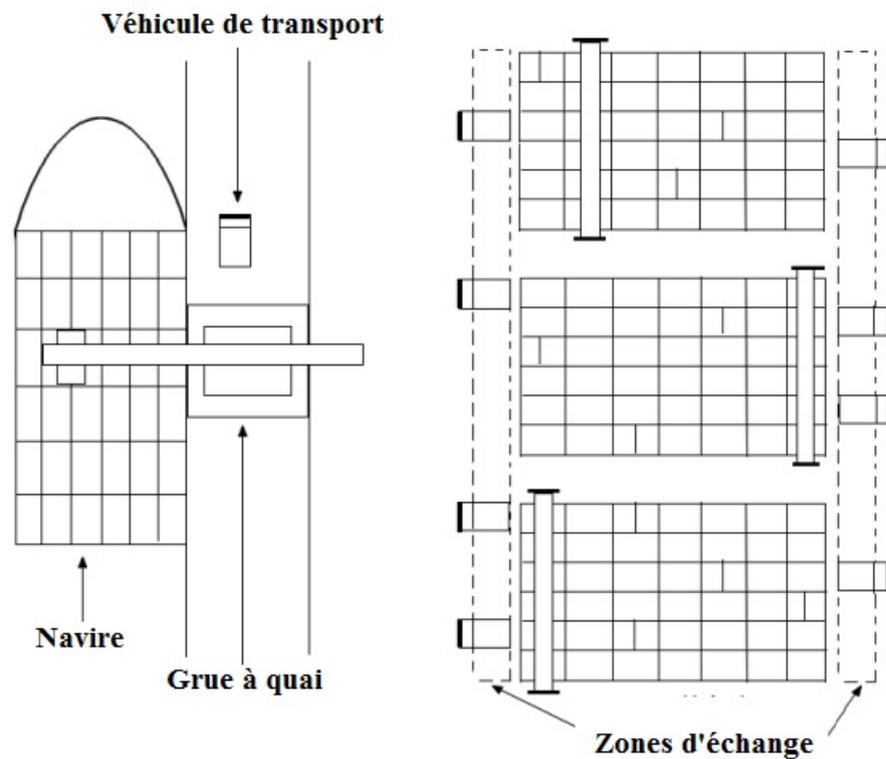


FIGURE 1.6: Terminal qui utilise des RMGCs

### 1.3 Le problème de stockage de conteneurs

La cour de stockage est une ressource capitale dans un terminal à conteneurs. En effet, le degré d'efficacité de sa gestion se répercute sur la productivité globale du port. Suite à la baisse des frais de transport de marchandises et de biens qu'a occasionné la conteneurisation, les flux de conteneurs ont largement augmenté dans les ports. La conséquence directe de ceci est une croissance vertigineuse du nombre de conteneurs qui séjournent simultanément dans un port, rendant ainsi insuffisants les emplacements de stockage au sol.

Une solution de ce problème pourrait être l'augmentation des surfaces des cours de stockage, mais elle s'avère inefficace car les nombres de conteneurs qui entrent simultanément dans un port croissent continuellement en fonction du développement des moyens de transport (exemple : les navires porte-conteneurs). À cela s'ajoute la cherté des parcelles de terrains. Une autre solution à ce problème est d'utiliser l'avantage de la standardisation des tailles des conteneurs, qui rend possible leur superposition.

Pour être efficace, la superposition des conteneurs doit aussi tenir compte des dates d'évacuation. En effet, un rangement aléatoire des conteneurs dans une pile peut rendre certains

conteneurs inaccessibles lors de leurs extractions. Ainsi, il sera nécessaire d'effectuer des opérations de réarrangement de conteneurs juste avant les heures de départ de certains d'entre eux.

Ces opérations sont appelées des *remaniements*, et sont considérées comme des mouvements improductifs qui consomment du temps et qui ralentissent les opérations portuaires. Pour éviter ce genre de désagrément, plusieurs travaux de recherche ont été réalisés ces dernières décennies afin de proposer des méthodes de stockage efficaces et des systèmes d'aide à la décision[17, 5, 25].

## 1.4 Les méthodes de résolution

La résolution du problème de stockage de conteneurs se fait généralement en deux étapes. Dans la première phase, des analyses sont effectuées pour rechercher la méthode de stockage adéquate. Ensuite, la partie simulation permet de vérifier et de mesurer l'efficacité des décisions prises dans la partie précédente.

Certains travaux disponibles dans la littérature se sont uniquement focalisés sur l'étude analytique du problème, en proposant des méthodes de stockage pertinentes. Alors que d'autres, en plus de cela, se sont servis de la simulation pour prouver l'efficacité des techniques de stockage qu'ils ont proposées.

Dans la majorité des terminaux à conteneurs, les conteneurs ont des durées de séjour variables. Cependant, avec l'agrandissement continu des navires porte-conteneurs, les terminaux sont de plus en plus confrontés à la nécessité d'une bonne gestion de la cour de stockage et des équipements de manutention. Lorsque les durées de séjour des conteneurs au port sont trop longues, le risque encouru est un encombrement de la cour de stockage, ce qui est néfaste car pouvant provoquer des pénuries d'espace et des difficultés à accéder à certains conteneurs. Le risque d'avoir ce genre de situation est plus élevé avec le cas des conteneurs importés qui sont généralement réclamés individuellement par des camions. Ainsi, pour encourager les propriétaires à ne pas laisser leurs conteneurs au port pendant des durées indéterminées, certains auteurs ont proposé de fixer une durée de stockage gratuit au delà de laquelle il y a des frais de stockage[5].

Grâce au développement croissant et continu de la technologie, l'ordinateur devient de plus en plus sophistiqué. L'une des principales utilités de cette machine est le fait qu'elle soit capable d'effectuer rapidement des tâches (calculs) qui pourraient nécessiter beaucoup de temps à l'homme. En plus de cela, la diversité des langages de programmation fait de lui un outil très convoité pour effectuer des calculs ou bien même des simulations numériques. Ces atouts sont de plus en plus exploités pour résoudre le problème du stockage de conteneurs dans un terminal portuaire.

Un autre avantage de la simulation est le fait qu'elle permette de réaliser plus facilement des études quantitatives. Sculli et Hui [19], s'en sont servi pour étudier un cas particulier du

problème de stockage de conteneurs, où tous les conteneurs ont les mêmes dimensions. Ils ont basé leur étude sur l'observation des variations de quatre critères de performance : le taux de remplissage de la cour de stockage, le nombre de mouvements improductifs, le nombre de conteneurs qui n'ont pas pu être stockés, et aussi le nombre de places non occupées.

Jiang et al. ont proposé dans [11] un système, codé en  $C++$ , qui contient à la fois une méthode de répartition de la cour de stockage et une technique d'attribution des charges de travail aux grues de cour. Ce système, qui est spécialement conçu pour la résolution du problème de stockage de conteneurs qui sont en transition entre des navires différents, progresse en deux phases successives.

Jusqu'ici, les travaux relatés n'incluaient pas de techniques d'optimisation précises. Cependant, des publications apparues au cours de ces dernières décennies ont prouvé l'utilité de la recherche opérationnelle dans la résolution du problème de stockage de conteneurs. Certaines des méthodes d'optimisation rencontrées dans la littérature, fournissent des résultats optimaux. Mais le principal point faible de ces méthodes exactes est le fait qu'elles nécessitent beaucoup de temps de calcul, dans la plupart des cas. Ainsi, pour remédier à cela, des algorithmes heuristiques et méta-heuristiques sont créés. Ces derniers nécessitent généralement des temps de calcul raisonnables, mais l'optimalité des solutions qu'ils procurent n'est pas garantie. Les algorithmes heuristiques, proposés dans la littérature, sont généralement spécifiques à des cas particuliers de problèmes. Alors que les algorithmes méta-heuristiques sont plus flexibles, les plus connus (utilisés) d'entre eux sont : *la méthode tabou*, *le recuit simulé*, *les algorithmes génétiques*, et *l'algorithme de colonie de fourmis*.

## Conclusion

Dans ce chapitre, nous avons expliqué qu'est ce qu'un conteneur, l'histoire de la conteneurisation et présenté les changements qu'a apporté cette technologie dans le monde entier. Nous avons présenté une étude bibliographique sur le problème de stockage de conteneurs dans un terminal portuaire, ainsi que sur des méthodes de résolution analytiques et numériques. Certains papiers ont utilisé des techniques d'optimisation, qui sont expliquées dans le prochain chapitre, pour résoudre le problème de stockage de conteneurs. Ces méthodes d'optimisation peuvent être : exactes, heuristiques, ou méta-heuristiques.

# 2

## Rappels théoriques et outils

La recherche opérationnelle (RO) peut se définir comme la mise en œuvre de méthodes scientifiques, essentiellement mathématiques d'optimisation, en vue de prendre la meilleure décision possible. Établir plus précisément quand faire appel à la recherche opérationnelle dans un processus de prise de décision est une tâche plus difficile. En effet, l'expert en recherche opérationnelle exploite des connaissances provenant de sources particulièrement variées qu'on peut classer en trois grandes catégories : les mathématiques appliquées, l'informatique et les modèles métiers [12].

Parmi les outils de la recherche opérationnelle et non des moindres on cite l'optimisation combinatoire. Celle-ci définit un cadre formel pour de nombreux problèmes de l'industrie, de la finance ou de la vie quotidienne. Elle permet de modéliser, d'analyser et de résoudre analytiquement ou numériquement les problèmes consistant à déterminer la (les) solution (s) satisfaisant un objectif quantitatif tout en respectant d'éventuelles contraintes.

### 2.1 Les problèmes d'optimisation combinatoire

Les problèmes d'optimisation combinatoire sont répartis en deux grandes classes :

#### 2.1.1 Les problèmes faciles

Un problème est dit appartenir à la classe des problèmes faciles s'il existe un algorithme polynomial pour le résoudre [1]. Parmi ces problèmes on cite :

- Le problème de cheminement.
- Le problème d'affectation.

- Le problème de flot maximum.
- Le problème de transport.

### 2.1.2 Les problèmes difficiles

Nous considérons comme difficiles des problèmes qui ne sont pas faciles [1]. On peut citer comme exemple :

- Le problème de coloration (des sommets et des arêtes).
- Le problème de voyageur de commerce (TSP).
- Le problème d'ordonnancement.
- Le problème du sac-à-dos.

## 2.2 Modélisation des problèmes d'optimisation combinatoire

Les problèmes d'optimisation combinatoire peuvent être modélisés en utilisant :

### 2.2.1 La programmation linéaire

Un programme linéaire a la forme suivante :

$$(PL) = \begin{cases} \min z = cx, \\ Ax \geq b, \\ x \geq 0. \end{cases}$$

Où  $A$  est une matrice ( $m \times n$ ),  $b$  un vecteur de dimension  $m$ ,  $c$  un vecteur de dimension  $n$  et  $x$  est un vecteur inconnu.

### 2.2.2 La programmation linéaire et non-linéaire en nombres entiers

Un programme linéaire ou non-linéaire en nombres entiers est le suivante : on cherche à maximiser, ou minimiser une fonction de  $k$  variables, lorsque ces variables sont soumises à des contraintes :

$$(P) = \begin{cases} \min z = f(x_1, x_2, \dots, x_k) \\ g_\alpha(x) \leq A_\alpha; \alpha \in I \\ x_i \in \mathbb{N}; i = 1 \dots k. \end{cases}$$

Lorsque la fonction  $z$ , l'ensemble des contraintes  $I$  et les contraintes  $g_\alpha$  sont linéaires, la programmation est dite linéaire. Lorsque la fonction  $f$  ou les contraintes  $g_\alpha$  ne sont pas linéaires, on dit que le problème de programmation est non linéaire.  $x$  Est un vecteur inconnu. Dans le cas particulier, où les contraintes  $x_i \in \mathbb{N}$  sont remplacées par  $x_i \in \{0, 1\}$ , on dit qu'on a un programme linéaire ou non-linéaire en 0 – 1.

### 2.2.3 La théorie des graphes

La théorie des graphes est un outils très puissant de modélisation de la recherche opérationnelle en général, et de l'optimisation combinatoire en particulier. Elle comprend les définitions des différentes structures de graphes connues (chemin, parcours, ensemble stable, etc...), ainsi que les problèmes de décision ou d'optimisation liés à ces structures (plus court chemin, postier chinois, voyageur de commerce, etc..)[1]

Les problèmes connus des graphes sont

## 2.3 Les méthodes de résolution

Les méthodes d'optimisation peuvent être classées en deux grandes classes : *exactes* et *heuristiques*. Dans les méthodes exactes, le principe consiste à trouver une solution optimale globale, en utilisant une recherche exhaustive (exemple : *Branch and bound*, *Coupe de Gomory*, *Branch and Cut* ...). Néanmoins, Dans de nombreux problèmes d'optimisation (trop de données, taille énorme des instances, ...), la détermination d'une solution exacte prend trop de temps de calcul pour qu'il soit raisonnable de chercher une telle solution. C'est ainsi que l'orientation est faite vers les méthodes heuristiques.

### 2.3.1 Les méthodes exactes

On peut définir une méthode exacte comme une méthode qui garantit l'obtention de la solution optimale pour un problème d'optimisation. L'utilisation de ces méthodes s'avèrent particulièrement intéressante, mais elles sont souvent limitées au cas des problèmes de petite taille[?].

### 2.3.2 Les méthodes approchées

Se sont des méthodes de calcul qui fournissent rapidement une solution réalisable, pas nécessairement optimale ou exacte, mais proche de l'optimum pour un problème d'optimisation de grande taille.

## 2.4 La programmation linéaire en nombres entiers

La programmation en nombres entiers n'est pas uniquement un outil de modélisation mais comprend également un ensemble de méthodes de résolution pour les problèmes d'optimisation combinatoire[1].

**Définition 2.4.1.** *Un problème d'optimisation en nombres entiers est un problème de classe NP-difficile (sauf cas particuliers), dont toutes les variables sont contraintes à ne prendre*

que des valeurs entières, il s'écrit sous la forme suivante :

$$(PLNE) = \begin{cases} \min z = cx, \\ Ax \geq b, \\ x \in \mathbb{N}^n. \end{cases}$$

Si ces variables sont booléennes (qui prennent les valeurs 0 ou 1), alors on parle d'un programme binaire (bivalent) qui s'écrit comme suit :

$$(PB) = \begin{cases} \min z = cx, \\ Ax \geq b, \\ x \in \{0, 1\}^n. \end{cases}$$

Résoudre ces problèmes revient à déterminer les valeurs (entières/binaires) des variables qui permettent de minimiser la fonction objectif  $z$ .

Beaucoup de méthodes ont été élaborées pour la résolution de tels problèmes. Elles sont réparties en deux grandes catégories : méthodes exactes et méthodes approchées [1].

### 2.4.1 Les méthodes exactes

1. **Algorithme de Branch and Bound** : C'est une méthode de recherche arborescente basée sur le principe de séparation et d'évaluation. Elle consiste en la construction d'un arbre de recherche qui sera exploré de manière à éviter les branches inutiles qui sont des branches contenant des solutions non intéressantes ou carrément non réalisable.

L'exploration se fait avec des évaluations des branches et des comparaisons avec une valeur seuil de critère à optimiser. Cette technique donne de bons résultats pour les problèmes de petite taille, mais dans le cas des grandes tailles elle risque de générer des branches très étendues[1, 24].

2. **Coupe de Gomory** : Ce sont des méthodes bien souvent itératives visant à isoler l'enveloppe convexe des solutions entières, et cela par l'ajout progressivement des contraintes supplémentaire (inégalités valides) au programme linéaire continu et dans chaque itération on résout un  $PL$  augmenté jusqu'à l'obtention d'une solution entière. Ces contraintes permettent en fait d'éliminer les parties inutiles du polyèdre des solutions du programme continu sans éliminer les solutions entières[18, 24].

3. **La méthode de Branch and Cut** : C'est une méthode qui conjugue entre les efforts de l'algorithme de Branch and Bound et de la méthode des coupes polyédrales, elle est utilisée pour la première fois pour résoudre le problème de *linear ordering*.

La résolution d'un programme linéaire en nombres entiers se fait en résolvant d'abord la relaxation continue de ce dernier, par la suite on applique la méthode des coupes

sur la solution trouvée, si celle-ci n'arrive pas à obtenir une solution entière alors le problème est divisé en plusieurs sous problème qui seront résolus de la même façon[24].

### 2.4.2 Les méthodes approchées

Pour résoudre un problème difficile de grande taille on n'a pas d'autre choix que de recourir à des méthodes approchées. Dans cette catégorie on peut distinguer :

- ▶ Les heuristiques pour lesquelles on peut quantifier l'erreur (recherche locale, méthodes gloutonnes, ... etc).
- ▶ Les méta-heuristiques, qui sont des méthodes inspirées de phénomènes naturels (algorithme génétiques, colonies de fourmis et d'abeilles ... etc).

Pour des raisons pragmatiques, nous ne détaillons que l'une des approches métaheuristiques citées, à savoir l'approche des algorithmes génétiques.

#### Algorithmes génétiques

Les algorithmes génétiques sont créés en 1975 par le chercheur américain Jonh Henry Holland, qui a publié, dans [10], le premier modèle formel appelé *the canonical genetic algorithm*. Ils font partis des algorithmes évolutionnaires, et sont basés sur les processus naturels de reproduction animale. En effet, ces processus sont mieux connus grâce aux progrès des recherches effectuées sur les cellules animales, qui ont révélé l'existence de deux phénomènes remarquables qui se produisent au cours de la reproduction animale, à savoir : le croisement et la mutation.

Le croisement, comme son nom l'indique, est le phénomène qui associe une partie d'un chromosome masculin à une autre partie d'un chromosome féminin, pour créer un nouveau chromosome. La Figure 2.1 en est une illustration.

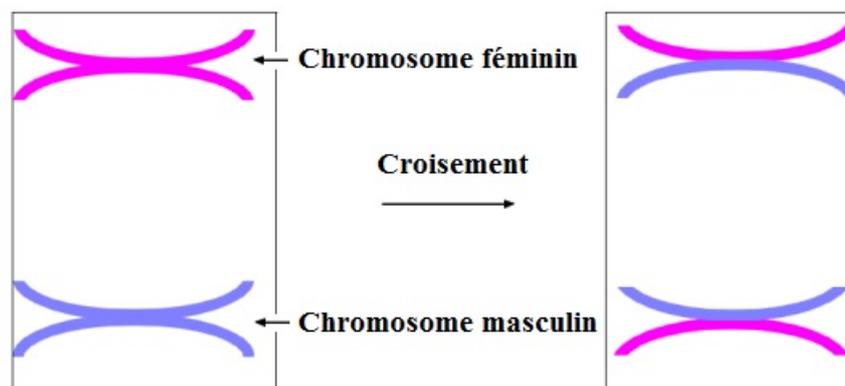


FIGURE 2.1: Croisement cellulaire

La mutation est un phénomène naturel très rare qui n'est pas indispensable dans la reproduction animale. Comme son nom l'indique, elle modifie certaines informations contenues dans un chromosome (voir la figure 2.2), et par conséquent elle change certains caractères du nouvel individu créé. Dans la vie réelle, elle peut avoir des conséquences néfastes telles que des malformations et des maladies génétiques. Mais, depuis quelques temps, les chercheurs essaient d'exploiter les aspects positifs de ce phénomène, surtout dans le domaine de l'agriculture. D'ailleurs, ce phénomène a une utilité en optimisation, car il peut éviter une convergence rapide vers un optimum local.

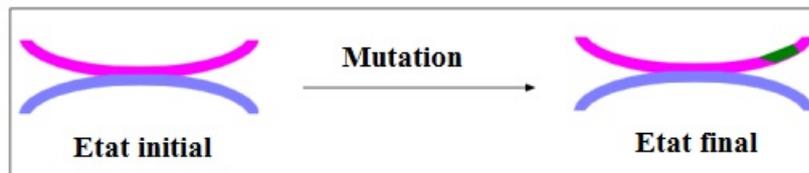


FIGURE 2.2: Mutation cellulaire

L'implémentation d'un algorithme génétique requiert la détermination préalable de six paramètres qui sont détaillés ci-dessous[16].

1. **Un principe de codage**, autrement dit une manière de représenter un individu de la population étudiée. Les types de codage les plus fréquemment rencontrés dans la littérature sont : le codage binaire (qui est représenté par une succession de 0 et 1), le codage à caractères multiples, et le codage sous forme d'arbre (qui utilise une structure arborescente, à partir de laquelle pourront être issus un ou plusieurs fils). Il est capital de bien choisir un codage adéquat au problème étudié, car cela conditionne le succès de l'algorithme.
2. **Une manière de générer la population initiale**, qui doit permettre de créer des individus qui ne sont pas identiques. L'efficacité ou la non-efficacité de cette méthode se répercute sur les résultats finaux de l'algorithme, car une population initiale de très mauvaise qualité peut conduire à une convergence prématurée vers un optimum local, qui peut être très éloigné de l'optimum global. Il est donc plus avantageux d'avoir une population initiale qui est bien répartie sur tout le domaine de définition du problème étudié.
3. **Une fonction d'évaluation**, qui permet de calculer la performance (appelée fitness) de chaque individu de la population. Elle sert aussi à connaître le meilleur individu (solution courante) d'une population, mais elle est souvent aussi utilisée pour sélectionner des individus à partir desquels la prochaine génération sera créée.
4. **Un opérateur de croisement**, qui permet de créer des descendants de la population initiale, il permet aussi aux anciennes générations de transmettre leurs caractères aux

nouvelles générations. Les méthodes de croisement les plus rencontrées dans la littérature sont : le croisement en un point, le croisement en deux points, la méthode du couper-et-joindre, le croisement uniforme, et le croisement avec trois parents.

- **Croisement en un point** : cette méthode choisit un point de section quelconque, ensuite elle intervertit les parties qui sont à sa droite dans les deux parents. La figure 2.3 en est une illustration.

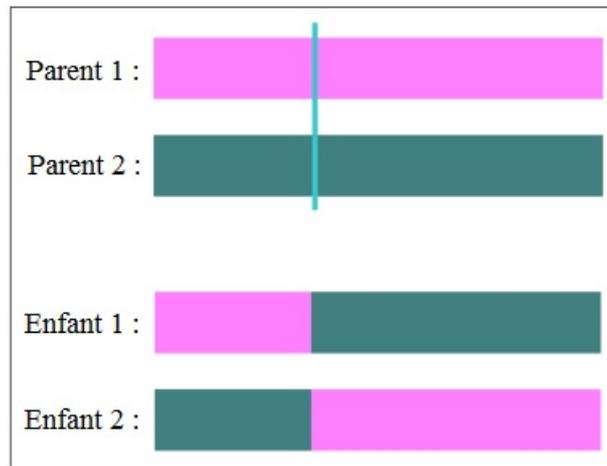


FIGURE 2.3: Croisement en un point

- **Croisement en deux points** : elle est semblable au croisement en un point, mais deux points de section sont choisis au lieu d'un seul. Ensuite les parties qui sont entre ces deux sections sont échangées, comme le montre la figure 2.4.

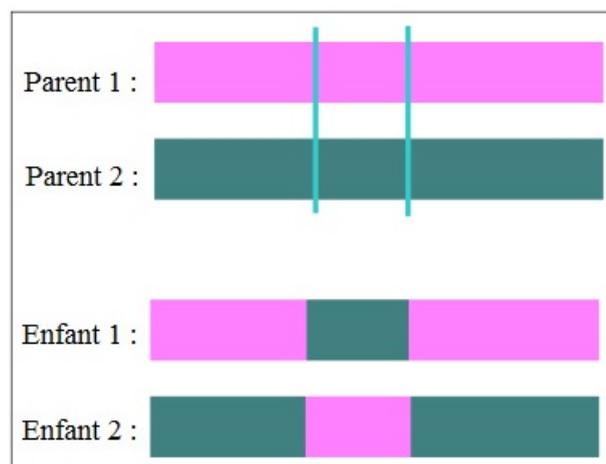


FIGURE 2.4: Croisement en deux points

Une variante de cette méthode est le croisement partiel (ou *partially mapped crossover*), qui fut proposé par Goldberg et Lingle [8]. L'originalité de cette méthode est

le fait qu'elle fournit une technique de correction des nouveaux chromosomes afin de remédier aux possibles violations de certaines contraintes. Elle progresse comme suit.

- 1 Choisir aléatoirement deux points de section;
  - 2 Échanger les parties qui sont à leurs intérieurs, pour créer deux nouveaux enfants;
  - 3 Déterminer un processus de correction;
  - 4 Utiliser ce processus pour légaliser les enfants.
5. Le cinquième paramètre de l'algorithme génétique est **l'opérateur de mutation**. Ce dernier effectue une légère perturbation sur un chromosome en altérant un ou plusieurs de ses gènes. Il permet essentiellement de maintenir la diversité de la population, mais il peut parfois améliorer la valeur (performance) d'un chromosome. En plus de cela, il peut éviter une convergence prématurée de l'algorithme génétique. Les méthodes de mutation les plus connues dans la littérature sont : le basculement d'un bit, l'utilisation d'une borne, la méthode non-uniforme, la méthode uniforme, la méthode gaussienne, et la méthode d'échange.
- Le basculement d'un bit** : il concerne uniquement le cas où les chromosomes sont codés en binaire, et consiste simplement à changer le contenu d'un gène (0 devient 1, et 1 devient 0).
6. L'algorithme génétique fait aussi intervenir des **paramètres numériques**, qui sont : la taille maximale d'une population ( $NI_{Max}$ ), le nombre de générations successives à

créer ( $NG_{Max}$ ), la probabilité de croisement ( $p_c$ ), et la probabilité de mutation ( $p_m$ ).

```

1 Créer une population initiale;
2 Évaluer chaque individu;
3 Initialiser le compteur du nombre de générations,  $NG = 1$  ;
4 tant que  $NG < NG_{Max}$  faire
5     Sélectionner une partie de la population courante;
6     Initialiser une nouvelle génération qui contient les individus sélectionnés;
7     tant que la nouvelle génération ne contient pas  $NI_{Max}$  éléments faire
8         Choisir dans la population sélectionnée deux parents ( $P_1$  et  $P_2$ );
9         Choisir aléatoirement entre 0 et 1 un réel noté  $c$ ;
10        si  $c < p_c$  alors
11            | faire un croisement entre  $P_1$  et  $P_2$  pour créer deux enfants;
12        fin
13        Ajouter dans la nouvelle génération le meilleur enfant;
14    fin
15    Choisir aléatoirement entre 0 et 1 un réel noté  $m$ ;
16    si  $m < p_m$  alors
17        | faire une mutation sur un individu choisi aléatoirement dans la nouvelle
18        | génération;
19    fin
20    Évaluer la nouvelle génération;
21    Incrémenter le compteur,  $NG = NG + 1$ ;
22 fin

```

**Algorithme 1 :** La succession des étapes de l'algorithme génétique

La sélection des individus à partir desquels on crée une nouvelle génération peut être réalisée de plusieurs manières. Les méthodes de sélection les plus connues sont : la méthode élitiste, la roue de la fortune, la sélection par rang, la sélection par tournoi, et la sélection uniforme. Nous ne nous intéresserons qu'à la méthode élitiste que nous nous utilisons dans ce mémoire.

**Méthode élitiste :** elle sélectionne les meilleurs individus en se basant sur leurs fitness. L'avantage de cette méthode est le fait qu'elle permet de ne pas perdre les meilleurs individus au cours des générations, mais cela augmente aussi le risque d'une convergence prématurée vers un optimum local.

## 2.5 Le calcul informatique

Le calcul informatique est nécessaire pour résoudre pratiquement, c'est-à-dire numériquement, les problèmes posés. Ainsi, la notion du temps de calcul est centrale dans le développement et la mise au point d'un algorithme.

Plusieurs bibliothèques logicielles, distribuées sous des licences libres ou commerciales, offrent aux experts en recherche opérationnelle des outils rapidement utilisables basés sur des algorithmes et implantations efficaces. Ces librairies sont dédiées soit à des problèmes mathématiques (comme la programmation linéaire ou non-linéaire), soit à des problèmes métiers (outil dédié à un problème de tournées ou un problème d'ordonnancement). La connaissance des *problèmes métiers* est en effet la troisième composante à maîtriser pour le spécialiste de la recherche opérationnelle.

### 2.5.1 Microsoft Excel

Microsoft Excel est un logiciel tableur de la suite bureautique Microsoft Office développé et distribué par l'éditeur Microsoft. La version la plus récente est Excel 2019 [23].



FIGURE 2.5: Logo de Microsoft Excel

Il est destiné à fonctionner sur les plates-formes *Microsoft Windows*, *Mac OS X*, *Android* ou *Linux*. Le logiciel Excel intègre des fonctions de calcul numérique, de représentation graphique, d'analyse de données (notamment de tableau croisé dynamique) et de programmation, laquelle utilise les macros écrites dans le langage *VBA* (Visual Basic for Applications) qui est commun aux autres logiciels de Microsoft Office. Depuis sa création au début des années 1980 mais surtout à partir de sa version 5 (en 1993), Excel a connu un grand succès tant auprès du public que des entreprises prenant une position très majoritaires face aux logiciels concurrents. Les principaux formats de fichiers natifs portent l'extension *xls* (1995 – 2003) et *xlsx* (2007 – 2016). Chaque fichier correspond à un classeur, lequel contient des feuilles de calculs organisées. Chaque feuille correspond à un tableau de lignes et de colonnes pouvant contenir des valeurs (numériques ou non) ainsi que des formules permettant les calculs. Suivant les versions, les classeurs peuvent aussi inclure les éléments de programmation que sont les macros. Depuis avril 2014, l'application Excel 2013 est disponible sur *iPad*, *iPhone*, *Windows Phone* et sur *OS Android* depuis 2015. Cela entre dans la logique *Office mobile* de Microsoft de

rendre ses logiciels disponibles sur le plus grand nombre de supports (ordinateurs, tablettes, smartphones)[23].

### 2.5.2 Visual Basic for Application *VBA*

Le *VBA* (Visual Basic for Application) est un langage de programmation permettant d'utiliser du code dont la syntaxe est proche du langage *Visual Basic* pour exécuter les nombreuses fonctionnalités notamment des applications *Microsoft Office* et par exemple de l'application *Excel* qui nous avons amenés beaucoup utilisé pendant notre stage pratique à *BMT*.

Créer des programmes en *VBA* permet notamment d'automatiser des tâches répétitives réalisées sous *Excel* grâce à la possibilité de manipuler les objets propres à *Excel* (Classeurs, Feuilles, Cellules, . . .), de stocker de l'information, de les traiter (nettoyer et visualiser), d'effectuer des instructions sous conditions et éventuellement, à plusieurs reprises en fonction d'un ou plusieurs critères.

Le langage *VBA* permet également de contrôler la quasi-totalité de l'*IHM* (Interface Homme-Machine : c'est-à-dire tous les médias qui permettent à un utilisateur d'interagir avec la machine) notamment d'*Excel*, ce qui inclut la possibilité de manipuler les fonctionnalités de l'interface utilisateur comme les menus, les barres d'outils et le fait de pouvoir personnaliser les boîtes de dialogue et les formulaires utilisateurs. Cela permet notamment de rendre une application développée sous *Excel* plus conviviale.

## Conclusion

Dans ce chapitre nous avons passé en revue quelques rappels théoriques sur la Recherche opérationnelle (RO), sur l'optimisation combinatoire, ses modèles ainsi que ses méthodes de résolution et l'outil informatique qui sont nécessaires pour modéliser des problèmes posés et les résoudre numériquement.

# 3

## Modélisation et résolution du problème

### Introduction

Le transport maritime devient de nos jours, de plus en plus important, il représente une alternative crédible et intéressante au transport terrestre et aérien, notamment avec l'évolution du phénomène de conteneurisation.

Dans ce chapitre, nous traitons le problème de stockage de conteneurs. Pour ce faire, nous commençons par une présentation du terminal à conteneurs de Béjaia avec la description de la position du problème. Par la suite nous proposons une modélisation mathématique par parties pour le problème, ainsi que la description d'algorithmes de résolution.

### 3.1 Présentation d'entreprise d'accueil BMT SPA

#### 3.1.1 Définition de BMT

BMT (Béjaia Mediterranean Terminal) est une entreprise spécialisée dans l'exploitation et la gestion du terminal à conteneurs du port de Béjaia.

BMT SPA a été créé sur décision du Conseil des Participations de l'Etat (CPE) en mai 2004, sous forme de partenariat entre l'Entreprise Portuaire de Bejaia (EPB) et le groupe PORTEK (société Singapourienne). PORTEK est un opérateur de terminaux à conteneurs, présent dans plusieurs ports dans le monde, et également spécialisée dans les équipements portuaires.

Le capital social de BMT s'élève à cinq milliards de dollars américain répartis à raison de 51% pour l'EPB et 49% pour le groupe PORTEK.



FIGURE 3.1: Les partenariats de BMT

L'entreprise BMT est implantée au sein du port de Bejaia. La situation géographique de celui-ci offre des commodités exceptionnelles. Au voisinage de BMT se trouve la gare ferroviaire et quelques minutes la sépare d'un aéroport international ce qui lui facilite le transport des marchandises de toute nature vers plusieurs destinations [22].

### 3.1.2 Structure organisationnelle de BMT

Dans la structure organisationnelle de BMT on distingue six directions principales : Direction Générale (**DG**), Direction de Ressources Humaines et Moyennes (**DRHM**), Direction des Opérations (**DO**), Direction Marketing (**DM**), Direction des Finances et de Comptabilités (**DFC**), Direction Technique (**DT**).

La direction qu'on a ciblée pour définir une problématique c'est la Direction des Opérations (**DO**), elle assure la planification des escales, du parc à conteneurs et la planification des ressources [22](équipes et équipements). Elle comprend quatre services :

**Service acconage** : Assure la gestion des opérations au niveau du terminal.

**Service manutention** : Assure la gestion des opérations au niveau des navires.

**Service ressources** : Assure une meilleure affectation des équipements et ressources.

**Service logistique** : Assure le suivi des moyens logistiques ainsi que la prestation logistique globale.

### 3.1.3 Le terminal à conteneurs

Le terminal à conteneurs se décompose en deux grandes zones :

- La partie quai.
- La partie terrestre.

### La partie quai

Le rôle de cette zone est de servir de point de transfert des conteneurs entre le terminal et les navires.

Les caractéristiques de cette zone sont résumées dans le tableau suivant :

Quai accostage	Longueur	500 m
	Profondeur	12 m
	Superficie	60 h
	Nombre de poste à quai	4

TABLE 3.1: Les caractéristiques du quai.

### La partie terrestre

Cette partie est subdivisée en quatre (04) zones :

- Parc à conteneurs pleins (importés),
- Zone visite,
- Zone de dépotage/empotage,
- Parc à conteneurs vides et empotés.

1. **Le parc à conteneurs pleins (importés) :** Dans cette cour sont entreposés temporairement les conteneurs déchargés des navires et destinés à être livrés aux clients par voie ferroviaire ou routière. Cette zone est répartie en cinq (05) blocs (A, B, C, D, E) disposés parallèlement au quai, chaque bloc est constitué de six (06) tronçons adjacents horizontaux formant les rangées et de 54 tronçons adjacents verticaux formant les travées (aussi appelés slots ou rues), de plus les conteneurs sont stockés en pile de sept (07) niveau. Ainsi, la position d'un conteneur dans la cour est caractérisée par une adresse formée du bloc, travée(rue), rangée et niveau.
2. **Zone visite :** Dans cette zone s'effectue le contrôle des marchandises portées dans les conteneurs (service vétérinaire, phytosanitaire, ...etc.), les conteneurs ayant fait la visite seront soit transférés à la zone du stockage ou livrés à leur propriétaire.
3. **La zone dépotage/empotage :** Dans cette zone s'effectue les opérations de dépotage et d'empotage tels que :
  - **Empotage :** C'est l'opération de chargement des marchandises à l'intérieur d'un conteneur, il peut être effectué soit dans les locaux du client soit à l'intérieur du terminal.
  - **Dépotage :** C'est l'opération de déchargement d'un conteneur de son contenu. Les marchandises dépotées sont livrées à leur propriétaire et les conteneurs vides sont transférés vers la ZEP (Zone Extra Portuaire) là ou ils sont stockés temporairement avant d'être réclamés.

4. **Parc à conteneurs vides et empotés :** Dans cette zone située près du quai sont stockés les conteneurs vides et empotés destinés à l'exportation. Tous les conteneurs vides sont stockés après leur restitution dans la ZEP située à trois kilomètres du port. Ceux qui sont réclamés par leur propriétaire sont rapprochés au terminal à l'aide des camions routiers et temporairement stockés dans cette zone avant qu'ils soient embarqués.

### 3.1.4 Les outils de gestion du terminal

Afin d'améliorer les opérations de manutention des conteneurs BMT s'est dotée d'un système informatique de gestion du terminal pour assurer une meilleure traçabilité du conteneur et de sa sécurité. Le système installé comprend le CTMS, . . .

#### CTMS (Container Terminal Management System)

BMT dispose d'un système logistique de gestion du terminal à conteneurs (CTMS) qui a pour objectif d'effectuer des activités, d'assurer une bonne planification du terminal, d'offrir un niveau élevé de l'efficacité opérationnelle pour ses clients, d'améliorer le service et s'adapter aux besoins des clients. Le CTMS assure plusieurs tâches telles que :

- le suivi du processus d'import et d'export ;
- la gestion de retour des conteneurs vides au terminal ;
- la gestion des restitutions des conteneurs (vides ou pleins) ;
- le suivi de dépotage des conteneurs ;
- la planification des activités sur les navires (chargement / déchargement) ;
- le suivi des opérations de chargement et de déchargement ;
- la réception des conteneurs à l'exportation ;
- le suivi des opérations de shifting au niveau du parc à conteneurs ;
- la facturation des clients.

### 3.1.5 Les procédures import/export de BMT

#### A l'import

1. **La visite :** pour permettre un bon suivi des visites conteneurs, le transitaire doit remettre au service des opérations certains documents pour la procédure administrative. Par la suite, l'agent de BMT établira une liste complète des conteneurs à préparer pour la visite du lendemain qui sera remise au chef de la section exploitation. Il doit à son tour confirmer la mise à disposition des conteneurs en zone de visite pour le lendemain.
2. **La pesée :** Le client est appelé à présenter au service des Opérations les documents nécessaires. A ce moment-là, l'agent BMT fait charger le conteneur sur un camion remorque pour effectuer la pesée.

3. **La livraison :** Pour permettre un suivi rigoureux des livraisons, le transitaire doit remettre un dossier complet. Par conséquent, l'agent chargé des opérations commerciales devrait confirmer la conformité du dossier pour établir le CDO (Container Delivery Order) et l'enregistrer sur fichier électronique consacré au suivi des livraisons.
4. **Le dépotage :** Le transitaire doit remettre à l'agent de BMT chargé des dépotages un dossier spécifique à l'opération. Par la suite, l'agent de BMT prépare le Container Movement Request (**CMR**) (document nécessaire pour le dépotage à remettre au pointeur affecté à la zone de dépotage), mais au préalable l'agent chargé des opérations commerciales remettra au chef de section exploitation une liste contenant tous les conteneurs à préparer pour le lendemain. Après chaque confirmation de fin dépotage, l'agent doit s'assurer que la lettre de dépotage soit signée par le responsable de section pour clôturer le dossier.

## A L'export

1. **La restitution :** Pour permettre un suivi rigoureux des restitutions, l'agent responsable de BMT doit exiger du pointeur une liste quotidienne des conteneurs restitués avec leurs positions au terminal et s'assurer de comparer les bons reçus avec le nombre total de conteneurs figurants sur la liste.
2. **Suivi des mises à quai :** Cette opération est assurée par l'agent responsable des restitutions, qui doit s'en assurer du bon suivi grâce à la tenue d'un fichier électronique mis à jours avec la saisie des restitutions journalières, et ce avec le concours du pointeur désigné et chargé pour le suivi des restitutions conjointement avec l'agent responsable des restitutions à la fin de la journée. La signature des mises à quai est assurée par le chef de section.
3. **Mise à disposition :** Le suivi des mises à disposition devrait être assuré par l'agent chargé des opérations commerciales responsable des mises à disposition, qui doit par conséquent tenir un fichier électronique spécialement consacré aux conteneurs mis à disposition.
4. **Empotage :** Le client est libre d'effectuer cette opération soit à l'intérieur du terminal à conteneurs (empotage à quai), soit à l'extérieur (empotage externe) dans ses propres locaux.

### 3.1.6 Quelques services de BMT

La performance d'un terminal à conteneurs se mesure par le temps d'escale, la rapidité des opérations, la qualité des services et le coût du transit du conteneur. BMT reçoit annuellement un grand nombre de navires pour lesquels elle assure les opérations de planification, de manutention et d'acconage telles que :

## Les planifications

- Planification des escales.
- Planification déchargement/chargement.
- Planification du parc à conteneurs.
- Planification des ressources équipes et moyens matériels.

## Manutention

Après accostage du navire, des équipes spécialisées s'occupent de toutes les opérations, le nombre de conteneurs embarqués ou débarqués par heure (cadence) constitue une mesure de la qualité de service.

## Acconage

Une fois le conteneur disposé dans le parc, les opérations suivantes peuvent alors prendre place :

- Visite.
- Pesée.
- Livraison.
- Mise à disposition.
- Restitution.

BMT assigne des ressources humaines et matérielles pour effectuer ses différents types de prestation aux conteneurs. Les portiques gerbeurs sur pneus (RTGC) sont essentiels à l'exécution de ces opérations. La qualité de service est mesurée à ce niveau par le nombre de livraisons et restitutions effectuées par jour.

### 3.1.7 Les objectifs de BMT

- \* Faire du terminal à conteneurs de BMT une infrastructure moderne à même de répondre aux exigences les plus sévères en matière de qualité dans le traitement du conteneur.
- \* La mise à disposition d'une nouvelle technologie dans le traitement de conteneurs pour :
  1. Un gain de productivité.
  2. Une réduction des coûts d'escale.
  3. Une fiabilité de l'information.
  4. Un meilleur service clientèle.
- \* Sauvegarder la marchandise des clients.
- \* Faire face à la concurrence nationale et internationale.
- \* Gagner des parts importantes du marché.

### 3.1.8 Position du problème

Le flux des conteneurs entrant à BMT a largement augmenté ces dernières années, ce qui a causé une croissance vertigineuse du nombre de conteneurs qui séjournent simultanément au port, rendant ainsi insuffisant l'espace de stockage au sol.

En outre les conteneurs déchargés des navires sont arrangés aléatoirement dans les piles (de 7 étages chacune) de la cour de stockages ce qui rend certains d'entre eux inaccessible au moment de leur récupération ainsi des remaniements (ré-arrangements) des conteneurs sont nécessaires pour accéder à ceux désirés, ces remaniements sont considérés comme des mouvements improductifs, car ils monopolisent le matériel et ralentissent les autres opérations portuaires.

Les mouvements de remaniement sont inévitables vu la limitation de la cour de stockage des conteneurs, mais les autorités portuaires veulent tout de même les minimiser.

Notre objectif est donc de maximiser le nombre des conteneurs à récupérer et chercher un ré-arrangement optimale de l'ensemble des conteneurs dans le parc parmi les emplacements disponibles dans la même rue de stockage, qui minimise les mouvements improductifs à l'avenir et cela en tenant compte des contraintes réelles du problème.

## 3.2 Le problème de sélection des conteneurs à récupérer

Dans cette section nous présentons la première partie du problème et sa formulation mathématique. Ensuite, nous proposons un algorithme pour sa résolution dans la section suivante.

### 3.2.1 Présentation de problème

Le problème de sélection des conteneurs à récupérer, est un problème de planification dans lequel nous disposons d'une liste de conteneurs à restituer, l'état du parc (la position, la date de restitution et Id de chaque conteneur) et les coûts de mouvements du RTGC. L'objectif est d'augmenter le nombre total des conteneurs récupérés, en respectant la contrainte des coûts des opérations.

### 3.2.2 La formulation mathématique du problème

#### Les constants

- $N_D$  C'est le nombre des conteneurs dans la liste,
- $C_r$  C'est le coût de récupération,
- $C_s$  C'est le coût de relocalisation (Shifting),
- $C_d$  C'est le coût de déplacement entre les deux rues adjacentes,
- $C$  C'est le coût total des opérations,
- $A_i$  C'est le nombre de conteneurs au-dessus du  $i^{\text{ème}}$  conteneur avant récupération,

$R_i$  C'est la rue du  $i^{\text{ème}}$  conteneur.

### Les variables

$$x_i = \begin{cases} 1, & \text{Si le } i^{\text{ème}} \text{ conteneur est sélectionné} \\ 0, & \text{Sinon} \end{cases}$$

### La contrainte

La seule contrainte c'est la contrainte des coûts qui peut représenter : le temps, la fiabilité des équipements, un coût financier, etc...

$\{x_j * R_j\}$  ( $j$  allant de 1 jusqu'à  $N_D$ ) l'ensemble de toutes les rues que le RTGC doit atteindre. L'expression  $\min\{x_j * R_j\}$  est une fonction qui retourne la première rue et l'expression  $\max\{x_j * R_j\}$  est une fonction qui retourne la dernière rue.

$$\sum_{i=1}^{N_D} x_i (C_r + A_i \times C_s) + C_d (\max\{x_j \times R_j, j = 1 \dots N_D\} - \min\{x_j \times R_j, j = 1 \dots N_D\}) \leq C$$

### La fonction objectif

L'objectif est de maximiser le nombre des conteneurs récupérés.

$$\max \sum_{i=1}^{N_D} x_i$$

## 3.3 La résolution du problème de sélection de conteneur à récupérer

Nous avons choisi de résoudre ce problème par un algorithme génétique adapté que nous détaillons dans cette section.

### 3.3.1 Codage utilisé pour notre algorithme

Puisque nous sommes dans le cas de la résolution d'un problème de sélection, le codage que nous avons retenu est de longueur égale au nombre total de conteneurs demandés à être préparés. L'individu est représenté par un bit de 1 ou 0 (sélectionné ou pas). La population

initiale est produite de manière spécifique au problème étudié (Algorithme 2), pour que l'algorithme démarre avec une population réalisable. La population est formée d'un certain nombre de chromosomes observés par  $P_{size}$ .

**Données :**  $P_{size}, C_r, C_s, C_d, C$ , Id des conteneurs, leur position et niveau d'accessibilité ;

**Résultat :**  $P_{size}$  solutions réalisables

```

1 Initialisation :  $i = 1$  ;
2 répéter
3   Sélectionner un conteneur aléatoirement;
4   répéter
5     ajouter son voisin ;
6   jusqu'à Son coût dépasse le coût total;
7   renvoi comme une solution réalisable;
8    $i = i + 1$ ;
9 jusqu'à  $i \leq P_{size}$ ;

```

**Algorithme 2 :** Générer aléatoirement des solutions réalisables

La figure 3.2 présente le codage d'une solution quelconque. Le conteneur numéro 4 sera sélectionné en remaniement le premier, suivi du conteneur numéro 7 et enfin le conteneur numéro 10, placé dans le dernier chromosome du code, sera récupéré en dernier. Nous dirons alors que le code génétique d'une solution envisagée est constituée de  $N$  conteneurs.

1	2	3	4	5	6	7	8	9	10
0	0	0	1	0	0	1	0	0	1

FIGURE 3.2: Codage d'un chromosome

### 3.3.2 Génération d'une population initiale

Dans cette étape, une population initiale doit être générée, où chaque chromosome représente une solution réalisable du problème. La procédure que nous avons utilisée pour générer la population initiale des individus, adaptée à notre problème est une génération de  $P_{size}$  individus par l'algorithme 2. Ce type de génération est le meilleur choix, car il permet l'hétérogénéité de la population.

### 3.3.3 Opérateur de sélection

La méthode utilisée est la sélection élitiste. Cette méthode consiste à sélectionner les  $n$  individus dont on a besoin pour la nouvelle génération  $P'$  en prenant les  $n$  meilleurs individus de la population  $P$  après l'avoir triée de manière décroissante selon la valeur de la fonction fitness de ses individus.

### 3.3.4 Opérateur de croisement

Pour obtenir de nouveaux individus (enfants) à partir d'une population initiale d'une itération, nous utiliserons l'opérateur de croisement en 2-points. Ce croisement s'effectuera comme suit :

*Étape 1* : Choisir deux individus de la population actuelle comme parents  $P_1$  et  $P_2$  pour générer deux enfants  $E_1$  et  $E_2$  ;

*Étape 2* : Générer aléatoirement deux positions  $k_1 \in [1, \dots, N]$  et  $k_2 \in [1, \dots, N]$  avec  $k_1 < k_2$  et générer une probabilité de croisement  $\alpha$  ;

*Étape 3* : Si  $\alpha \leq P_{cross}$  ;

*Étape 3-1* : Copier tous les éléments (les éléments qui sont inférieurs à  $k_1$  et supérieurs à  $k_2$ ) des parents choisis sur les deux enfants respectivement, ici, nous copions ceux du parent  $P_1$  à l'enfant  $O_1$ , ceux du parent  $P_2$  à l'enfant  $O_2$  ;

*Étape 3-2* : Compléter la partie restante de l'enfant  $E_1$  par les éléments du parent  $P_2$  et la partie restante de l'enfant  $E_2$  par les éléments du parent  $P_1$  en balayant de gauche à droite et ne reprenant que les éléments non encore transmis ;

*Étape 4* : Si  $\alpha > P_{cross}$ , copier le parent  $P_1$  à l'enfant  $E_1$  et le parent  $P_2$  à l'enfant  $E_2$ .

### 3.3.5 Opérateur de mutation

Les individus de la population issue du croisement vont ensuite subir un processus de mutation. L'opérateur de mutation utilisé est l'opérateur d'échange réciproque.

*Étape 1* : Sélectionner aléatoirement deux gènes  $s_1 \in [1, \dots, N]$  et  $s_2 \in [1, \dots, N]$  avec  $s_1 \neq s_2$  et générer une probabilité de mutation  $\beta$  ;

*Étape 2* : Si  $\beta \leq P_{mut}$ , changer les deux gènes, pour avoir deux enfants mutés noté  $E_{mut1}$  et  $E_{mut2}$  ;

### 3.3.6 Opérateur d'insertion

L'insertion dans le contexte de sélection est employée pour maximiser la fonction fitness. Cette insertion permet d'éliminer de la population les chromosomes les plus faibles. En effet, les individus qui sont générés aléatoirement et les enfants mutés (en nombre égal à  $2 * P_{size}$ ) sont triés, selon leur fonction de fitness dans un ordre décroissant. Seule la moitié supérieure de la population, correspondant aux meilleurs individus, est sélectionnée. Il est à noter que la taille de la population reste fixe (égale à  $P_{size}$ ) de génération en génération.

### 3.3.7 Fonction d'évaluation (fitness)

La fonction d'évaluation consiste à maximiser le nombre de conteneur (noté  $C_{max}$ ).

$$C_{max} = \begin{cases} \sum_{i=1}^{N_D} x_i, C_t \leq C \\ 0, Sinon \end{cases}$$

$$C_t = \sum_{i=1}^{N_D} x_i(C_r + A_i \times C_s) + C_d(max\{x_j \times R_j\} - min\{x_j \times R_j\}) \quad j = 1 \dots N_D$$

$C_t$  Désigne le cout total des mouvements sur les conteneurs sélectionnés pour les récupérer.

### 3.3.8 Critère d'arrêt

Le critère d'arrêt utilisé est le nombre de générations égal à  $Iter_{Max}$ . Après  $Iter_{Max}$  générations, l'algorithme génétique s'arrête et donne le meilleur chromosome qui possède un  $C_{max}$  maximale.

Notre algorithme (génétique) pour la sélection est initialisé par une taille de population ( $P_{size}$ ), une probabilité de croisement ( $P_{cross}$ ), une probabilité de mutation ( $P_{mut}$ ) et par un critère d'arrêt  $Iter_{Max}$ . Les étapes de cet algorithme sont présentés dans l'algorithme 3.

- 1 **Initialisation** :  $P_{size}, P_{cross}, P_{mut}, Iter_{Max}, \mu = 0$ ;
- 2 Générer  $P_{size}$  solutions réalisables par l'algorithme 1;
- 3 **répéter**
- 4     **tant que**  $\mu < 2P_{size}$  **faire**
- 5         Sélectionner aléatoirement deux parents de la population élitiste;
- 6         Croisement des deux parents pour obtenir deux enfants par une probabilité  $P_{cross}$ ;
- 7         Muter les deux enfants par une probabilité  $P_{mut}$ ;
- 8          $\mu = \mu + 1$ ;
- 9     **fin**
- 10     Évaluer tous les chromosomes ( $2P_{size}$ , parents et enfants) par la fonction d'évaluation fitness;
- 11     Ranger les parents et les enfants dans l'ordre décroissant selon leur fonction d'évaluation;
- 12     Supprimer les  $P_{size}$  chromosomes faibles et enregistrer les meilleurs chromosomes ( $Best_{Pop}$ ) selon leur fonction d'évaluation fitness;
- 13     Remplacer( $P_{size}, Best_{Pop}$ );
- 14 **jusqu'à**  $i = Iter_{Max}$ ;

**Algorithme 3** : Algorithme génétique pour la sélection des conteneurs

## 3.4 Le Problème de Rélocalisation des Conteneurs (PRC)

Dans cette section nous présentons la deuxième partie du problème et sa formulation mathématique. Sa résolution sera donnée exposée dans la prochaine section.

### 3.4.1 Présentation du problème

Le Problème de Relocalisation des Conteneurs est un problème de planification d'optimisation combinatoire dans lequel on nous donne une rue composée de  $W$  piles, de niveaux  $H$  et de  $N$  conteneurs. Chaque conteneur  $c$  a une priorité donnée,  $p(c)$ , qui indique son ordre dans la séquence de récupération. La pile dans laquelle le conteneur  $c$  est actuellement placé et notée  $s(c)$  et son niveau est noté  $t(c)$ . Le nombre de conteneurs dans la pile  $s$  est désigné par  $h(s)$ , et  $max(s)$  et  $min(s)$  indiquent respectivement la priorité la plus haute et la plus basse de  $s$ .

L'objectif du PRC est de déterminer la séquence de mouvements avec la longueur minimale à effectuer pour extraire  $K$  conteneurs d'une rue, où  $K \leq N$ . La récupération des conteneurs doit être effectuée en fonction de leur ordre de priorité décroissant, de telle sorte que le conteneur de priorité la plus haute 1, soit récupéré avant le conteneur de priorité 2, le conteneur 2 doit être récupéré avant le conteneur 3, et ainsi de suite, jusqu'à ce que tous les  $K$  conteneurs avec les priorités les plus élevées sont extraits de la rue.

Notons qu'après la fin du processus de récupération  $M = N - K$  conteneurs sont encore dans la rue.

À chaque étape, le conteneur avec la priorité la plus élevée parmi ceux qui sont actuellement placés dans la rue est désigné par  $c^*$ . Les caractéristiques techniques des équipements de manutention (grue à portique, chariot élévateur à fourche, etc.) utilisées dans l'environnement de stockage établissent les types d'opérations à effectuer qui sont :

1. Opération de récupération. Le conteneur avec la priorité la plus haute,  $c^*$ , actuellement placé en haut d'une pile dans la rue, est extraite de celle-ci.
2. Opération de relocalisation. Un conteneur placé au sommet d'une pile et déplacé de sa pile actuelle vers une autre.

Une solution pour le PRC est une séquence de  $n$ -uplets de la forme  $(a; b)$ , où  $a$  est la pile source,  $b$  est la pile cible et un conteneur à déplacer à chaque étape. Dans ce contexte, les mouvements d'extraction sont représentés par  $(a; -)$ , ce qui indique qu'un conteneur est extrait du haut de la pile  $a$ . Il est à noter que chaque solution réalisable pour le PRC contient  $K$  opérations de récupération et que, par conséquent, l'objectif du problème peut être traité en minimisant le nombre de mouvements de relocalisation.

Pour illustrer le PRC, prenons l'exemple illustré à la figure 3.3. Cet exemple montre une rue avec 6 piles, 4 niveaux et 16 conteneurs avec des priorités exclusives qui seront extraites sur la base de l'ordre de priorité prescrit. La priorité de chaque conteneur est représentée par un numéro. Une solution pour l'exemple donné est la suivante :  $((4, 3)(4, 3)(4, 6)(4, -)(2, 5)(2, 4)(2, -)$

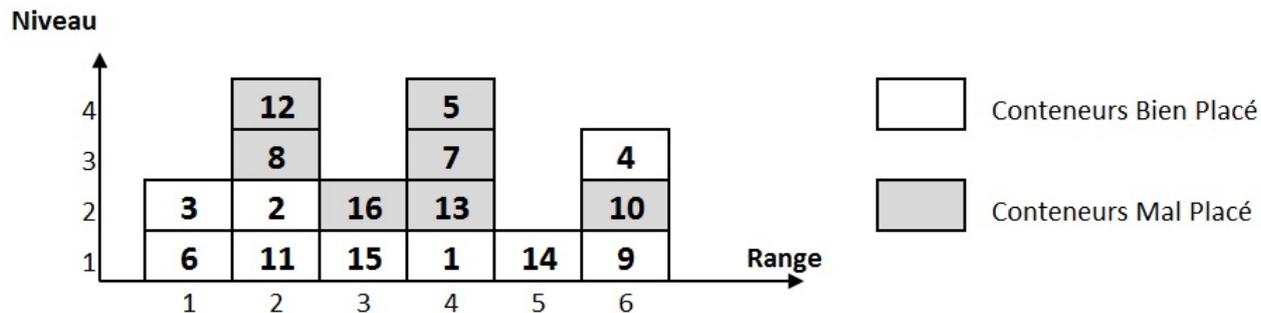


FIGURE 3.3: Exemple qui montre une rue avec 6 piles, 4 niveaux et 16 conteneurs

$(1, -)(6, 4)(6, -)(3, 2)(3, -)(1, -)(2, -)(5, -)(6, 3)(6, -)(3, -)(2, -)(4, 6)(4, -)(6, -)(5, -)(3, 1)(3, -)(1, -)$ .

Comme on peut le vérifier, cette solution est composée de 10 mouvements de relocalisation. Cependant, le nombre de mouvements de relocalisation peut être considérablement réduit.

### 3.4.2 La formulation mathématique du problème

Une analyse simple de l'exemple susmentionné établit la classification suivante des conteneurs en fonction de leurs priorités et des emplacements dans lesquels ils sont actuellement placés dans la rue :

**Conteneur mal placé :** Ce conteneur est actuellement placé au-dessus d'un autre avec une priorité plus élevée. Voir les conteneurs avec un fond gris à la figure 3.3. Il convient de noter que ce type de conteneur doit être déplacé dans une pile différente de la rue afin de permettre la récupération du conteneur de priorité supérieure placé en dessous.

L'ensemble des conteneurs mal placé dans la pile  $s$  est défini comme suit :

$$\Omega(s) = \{c | (s(c) = s) | \exists c' : (s(c') = s), (p(c') < p) \text{ et } (t(c') < t)\}. \quad (3.1)$$

L'ensemble de conteneurs mal placé dans une rue est désigné par :

$$\Omega = \bigcup_{1 \leq s \leq S} \Omega(s), \quad (3.2)$$

où  $S$  est l'ensemble des piles de cette rue.

**Conteneur bien placé :** Ce conteneur n'est actuellement placé au-dessus d'aucun autre avec une priorité plus élevée. Voir les conteneurs avec un fond blanc sur la figure 3.3. Ce type de conteneur ne nécessite aucun mouvement de déplacement avant son extraction. L'ensemble des conteneurs bien placé dans la pile  $s$  est défini comme suit :

$$\Upsilon(s) = \{c | (s(c) = s) \cap (c \notin \Omega(s))\}. \quad (3.3)$$

Dans ce qui suit, nous présentons une méthode pour calculer une limite supérieure pour le PRC. Cette limite supérieure peut être utilisée pour fournir une solution préliminaire au

système PRC ou pour définir un horizon de planification, comme il est illustré dans la section suivante.

**Données** : L'état du parc avec les conteneurs à récupérer, nombre max d'itérations  
**Résultat** : Solution du PRC

- 1 **Initialisation** :  $MeilleureSolution \leftarrow \emptyset$ ;
- 2 **pour**  $i \leftarrow 1$  à  $max$  **faire**
- 3     **pour**  $p \leftarrow 1$  à  $K$  **faire**
- 4          $c^* \leftarrow$  Trouver un conteneur de priorité  $p$ ;
- 5         **tant que**  $c^*$  est le plus haut **faire**
- 6              $O(c^*) \leftarrow$  Conteneurs placés sur  $c^*$  (Équation 3.4);
- 7              $O(c^*)' \leftarrow$  Sélectionner le plus haut conteneur parmi  $O(c^*)$ ;
- 8              $\Theta \leftarrow$  Piles cibles réalisables pour  $O(c^*)'$ ;
- 9              $s' \leftarrow$  Sélectionner aléatoirement une pile disponible parmi l'ensemble  $Theta$ ;
- 10             Relocaliser  $O(c^*)'$  de  $s(c^*)$  pour empiler  $s'$ ;
- 11         **fin**
- 12         Récupérer  $c^*$  du haut de sa pile;
- 13     **fin**
- 14     **Mise à jour de**  $MeilleureSolution$ ;
- 15 **fin**
- 16 **Retour de**  $MeilleureSolution$ ;

**Algorithme 4** : Méthode de calcul de la limite supérieure de la PRC

L'algorithme 5 est le pseudo-code de la méthode. Il parcourt chaque conteneur de la rue selon l'ordre de priorité décroissant (lignes 3 à 13). A chaque étape, on trouve ce conteneur avec la priorité la plus haute,  $c^*$  (ligne 4). Si  $c^*$  est placé en haut de sa pile, il peut être directement récupéré de la rue (ligne 12). Autrement, tant qu'il n'est pas libérée, la méthode identifie les conteneurs qui sont placés au-dessus de lui (ligne 6), notés  $O(c^*)$  et définis formellement comme suit :

$$O(c^*) = \{c | (s(c) = s(c^*)) \cap (t(c) > t(c^*))\}. \quad (3.4)$$

Cet ensemble est composé de conteneurs mal placés (équation 3.2) ils doivent donc être déplacés dans une pile différente.

Un exemple de ce fait est présenté lorsque le conteneur 1 est sur le point d'être récupéré de la rue. Avec la contrainte qu'un seul conteneur est géré à la fois (déplacé ou récupéré). Une fois définis les conteneurs à déplacer à chaque étape, notés  $O(c^*)'$ , le procédé identifie pour eux toutes les piles cibles réalisables (ligne 8). Une pile,  $s$ , pourrait être utilisée comme cible de  $O(c^*)'$  si et seulement si le nombre de créneaux vides est au moins égal à la cardinalité de  $|O(c^*)'|$ . C'est :

$$H - h(s) \geq 1 \quad (3.5)$$

Une pile,  $s'$ , est sélectionnée d'une manière aléatoire parmi un ensemble de piles réalisables noté  $\Theta$  (ligne 9) et  $O(c^*)'$  est repositionnée dans  $s'$  (ligne 10), au moyen du mouvement  $(s(c^*); s')$ . Enfin, afin de fournir une limite supérieure étroite, la méthode est intégrée à une stratégie de redémarrage (lignes 2 à 15) qui itère  $max$  times, où  $max$  est un paramètre dont la valeur est définie par l'utilisateur.

La modélisation du PRC est un programme linéaire mixte en nombres entiers MILPM (Mixed Integer Linear Programming MILP)[7]. Supposons que nous avons un horizon de planification  $T$ , obtenu par une technique d'optimisation (par exemple, l'algorithme 5) et sans perte de généralités, que tous les conteneurs placés dans la rue vont être récupérés, c'est-à-dire que  $K = N$ . Dans ce qui suit, nous introduisons d'abord les familles de variables utilisées par le modèle :

$$b_{i,j,n,t} = \begin{cases} 1, & \text{si } n \text{ est dans } (i,j) \text{ pendant la période } t, \\ 0, & \text{sinon,} \end{cases}$$

$$i = 1 \cdots W, \quad j = 1 \cdots H, \quad n = 1 \cdots N, \quad t = 1 \cdots T.$$

$$x_{i,j,k,l,n,t} = \begin{cases} 1, & \text{si } n \text{ est déplacé de } (i,j) \text{ dans } (k,l) \text{ pendant la période } t, \\ 0, & \text{sinon,} \end{cases}$$

$$i, k = 1 \cdots W, \quad j, l = 1 \cdots H, \quad n = 1 \cdots N, \quad t = 1 \cdots T.$$

$$y_{i,j,n,t} = \begin{cases} 1, & \text{si } n \text{ est extrait de } (i,j) \text{ au cours de la période } t, \\ 0, & \text{sinon,} \end{cases}$$

$$i = 1 \cdots W, \quad j = 1 \cdots H, \quad n = 1 \cdots N, \quad t = 1 \cdots T.$$

$$v_{n,t} = \begin{cases} 1, & \text{si } n \text{ a été récupéré avant la période } t, \\ 0, & \text{sinon,} \end{cases}$$

$$n = 1 \cdots N, \quad t = 1 \cdots T.$$

$$a_{i,j,n,t} = \begin{cases} 1, & \text{si } n \text{ est déplacé de } (i,j) \text{ dans la période } t, \\ 0, & \text{sinon,} \end{cases}$$

$$i = 1 \cdots W, \quad j = 1 \cdots H, \quad n = 1 \cdots N, \quad t = 1 \cdots T.$$

$$c_{i,j,n,t} = \begin{cases} 1, & \text{si } n \text{ est déplacé dans } (i,j) \text{ au cours de la période } t, \\ 0, & \text{sinon,} \end{cases}$$

$$i = 1 \cdots W, \quad j = 1 \cdots H, \quad n = 1 \cdots N, \quad t = 1 \cdots T.$$

$d(i, j, n, t)$  Nombre de conteneurs placés sur le conteneur  $n$  avant sa relocalisation dans  $(i, j)$  pendant la période  $t$ .

$$\forall i = 1 \cdots W, \quad j = 1 \cdots H, \quad n = 1 \cdots N, \quad t = 1 \cdots T.$$

$e(i, j, n, t)$  Nombre de conteneurs placés sur le conteneur  $n$  après sa relocalisation dans  $(i, j)$  pendant la période  $t$ .

$$\forall i = 1 \cdots W, \quad j = 1 \cdots H, \quad n = 1 \cdots N, \quad t = 1 \cdots T.$$

La fonction objectif du MILP est de maximiser les périodes que le dernier conteneur à récupérer  $N$  est hors de la rue :

$$\text{maximiser } \sum_{t=1}^T v_{n,t}. \quad (3.6)$$

Chaque conteneur doit être dans ou hors de la rue :

$$\sum_{i=1}^W \sum_{j=1}^H b_{ijnt} + v_{nt} \leq 1, \forall n = 1 \cdots N, t = 1 \cdots T \quad (3.7)$$

Dans chaque emplacement doit contenir au plus un conteneur :

$$\sum_{n=1}^N b_{ijnt} \leq 1, \forall i = 1 \cdots W, j = 1 \cdots H, t = 1 \cdots T \quad (3.8)$$

Les conteneurs doivent être placés les uns sur les autres et les emplacements vides entre eux ne sont pas autorisés :

$$\sum_{n=1}^N b_{ijnt} \geq \sum_{n=1}^N b_{ij+1nt}, \quad i = 1 \cdots W, j = 1 \cdots H, t = 1, \dots, T. \quad (3.9)$$

A chaque période, au plus une opération de conteneur est autorisée :

$$\sum_{i=1}^W \sum_{j=1}^H \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j+q-1 \leq H \\ l+q-1 \leq H}}^Q x_{ijklqt} + \sum_{i=1}^W \sum_{j=1}^H \sum_{n=1}^N y_{ijnt} \leq 1, \forall t = 1 \cdots T \quad (3.10)$$

Les opérations de récupération doivent être effectuées selon l'ordre de priorité des conteneurs :

$$\sum_{t=1}^T v_{nt} \geq \sum_{t=1}^T v_{n+1t} + 1, n = 1, \dots, N-1. \quad (3.11)$$

Les conteneurs sont hors de la rue s'ils ont déjà été récupérés :

$$v_{nt} = \sum_{i=1}^W \sum_{j=1}^H \sum_{t'=1}^{t-1} y_{ijnt'}, n = 1 \dots N, t = 1, \dots, T. \quad (3.12)$$

Le conteneur  $n$  est déplacé de la position  $(i, j)$  pendant la période  $t$  s'il est placé dans cette fente ( $b(i, j, n, t) = 1$ ) et s'il existe un mouvement de relocalisation associé à l'une des positions  $(i, j), (i, j-1), \dots, (i, j-Q-1)$ . Dans ces cas, au moins  $1, 2, \dots, Q$  conteneurs doivent être déplacés vers d'autres créneaux, respectivement :

$$a_{ijnt} \leq b_{ijnt}, i = 1 \dots W, j = 1 \dots H, n = 1 \dots N, t = 1, \dots, T-1. \quad (3.13)$$

$$a_{ijnt} \leq \sum_{\substack{j'=0 \\ j-j' \geq 1}}^{Q-1} \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j-j'+q-1 \leq H \\ l+q-1 \leq H \\ q > j'}}^Q x_{ij-j'klqt},$$

$$i = 1, \dots, W, j = 1, \dots, H, n = 1 \dots N, t = 1, \dots, T-1. \quad (3.14)$$

$$a_{ijnt} \geq b_{ijnt} + \sum_{\substack{j'=0 \\ j-j' \geq 1}}^{Q-1} \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j-j'+q-1 \leq H \\ l+q-1 \leq H \\ q > j'}}^Q x_{ij-j'klqt} - 1,$$

$$i = 1, \dots, W; j = 1 \dots H; n = 1, \dots, N; t = 1, \dots, T-1. \quad (3.15)$$

L'emplacement de chaque conteneur déplacé, dépend de sa position précédente et du mouvement de déplacement effectué :

$$c_{ijnt} \leq b_{ijn(t+1)}, i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N, t = 1 \dots T-1 \quad (3.16)$$

$$c_{ijnt} \leq \sum_{\substack{j'=0 \\ j-j' \geq 1}}^{Q-1} \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j-j'+q-1 \leq H \\ l+q-1 \leq H \\ q > j'}}^Q x_{kl ij-j'qt} - 1,$$

$$i = 1 \dots W, j = 1 \dots H, n = 1 \dots N, t = 1, \dots, T-1. \quad (3.17)$$

$$c_{ijnt} \geq b_{ijnt+1} + \sum_{\substack{j'=0 \\ j-j' \geq 1}}^{Q-1} \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j-j'+q-1 \leq H \\ l+q-1 \leq H \\ q > j^l}}^Q x_{kl ij-j'qt} - 1,$$

$$i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; t = 1, \dots, T - 1. \quad (3.18)$$

L'emplacement de chaque conteneur dépend de son emplacement précédent et des mouvements effectués :

$$b_{ijnt} = b_{ijn(t-1)} + c_{ijn(t-1)} - a_{ijn(t-1)} - y_{ijn(t-1)},$$

$$i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; t = 2, \dots, T. \quad (3.19)$$

Nombre de conteneurs à déplacer pour récupérer le conteneur  $n$  :

$$d_{ijnt} \leq Q \cdot a_{ijnt},$$

$$i = 1 \dots W, j = 1, \dots, H; n = 1, \dots, N; t = 1, \dots, T - 1,$$

$$\text{avec : } Q - 1 \geq 1. \quad (3.20)$$

$$d_{ijnt} \leq \sum_{j'=j+1}^H \sum_{n'=1}^N a_{ij'n't},$$

$$i = 1 \dots W, j = 1, \dots, H; n = 1, \dots, N; t = 1, \dots, T - 1. \quad (3.21)$$

$$d_{ijnt} \geq \sum_{j'=j+1}^H \sum_{n'=1}^N a_{ij'n't} - Q \cdot (1 - a_{ijnt}),$$

$$i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; t = 1, \dots, T - 1. \quad (3.22)$$

Nombre de conteneurs placés sur chaque conteneur après un mouvement de déplacement :

$$e_{ijnt} = c_{ijnt} \sum_{j'=j+1}^H \sum_{n'=1}^N c_{ij'n't},$$

$$i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; t = 1, \dots, T - 1. \quad (3.23)$$

$$e_{ijnt} \leq Q \cdot c_{ijnt},$$

$$i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; t = 1, \dots, T - 1,$$

$$\text{Sujet à } Q - 1 \geq 1. \quad (3.24)$$

$$e_{ijnt} \leq \sum_{j'=j+1}^H \sum_{n'=1}^N c_{ij'n't}, i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1 \quad (3.25)$$

$$e_{ijnt} \geq \sum_{j'=j+1}^H \sum_{n'=1}^N c_{ij'n't} - Q \cdot (1 - c_{ijnt}),$$

$$i = 1, \dots, W, j = 1, \dots, H, n = 1, \dots, N; t = 1, \dots, T - 1,$$

$$\text{Sujet à } Q - 1 \geq 1 \quad (3.26)$$

Le nombre de conteneurs placés sur chaque conteneur impliqué dans les mouvements de relocalisation doit être égal, avant et après le mouvement :

$$\sum_{i=1}^W \sum_{j=1}^H d_{ijnt} = \sum_{i=1}^W \sum_{j=1}^H e_{ijnt},$$

$$n = 1 \cdots N, t = 1 \cdots T - 1. \quad (3.27)$$

La relocalisation de  $q$  conteneurs de l'emplacement  $(i, j)$  n'est autorisée que s'il y a exactement  $q$  conteneurs supprimés :

$$\sum_{i=1}^W \sum_{j=1}^H \sum_{n=1}^N a_{ijnt} = \sum_{i=1}^W \sum_{j=1}^H \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j+q-1 \leq H \\ l+q-1 \leq H}}^Q q \cdot x_{ijklqt}, t = 1, \dots, T - 1. \quad (3.28)$$

Les mouvements de relocalisation au sein d'une pile ne sont pas autorisés :

$$x_{ijlqt} = 0,$$

$$i = 1, \dots, W; j = 1, \dots, n; l = 1 \cdots H, q = 1 \cdots Q, t = 1 \cdots T,$$

$$\text{Sujet à } j + q - 1 \leq H \text{ et } l + q - 1 \leq H. \quad (3.29)$$

Domaine des variables :

$$b_{ijnt} \in \{0, 1\}, i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T. \quad (3.30)$$

$$x_{ijklqt} \in \{0, 1\},$$

$$i, k = 1 \cdots W, j, l = 1 \cdots H, q = 1 \cdots Q, t = 1 \cdots T,$$

$$\text{Sujet à } j + q - 1 \leq H \text{ et } l + q - 1 \leq H. \quad (3.31)$$

$$y_{ijnt} \in \{0, 1\}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T \quad (3.32)$$

$$v_{nt} \in \{0, 1\}, \quad n = 1 \cdots N, t = 1 \cdots T \quad (3.33)$$

$$a_{ijnt} \in \{0, 1\}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1 \quad (3.34)$$

$$c_{ijnt} \in \{0, 1\}, \quad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; t = 1, \dots, T - 1. \quad (3.35)$$

$$d_{ijnt} \in \{0, Q - 1\}, \quad i = 1, \dots, W; j = 1, \dots, H; n = 1, \dots, N; t = 1, \dots, T - 1. \quad (3.36)$$

$$e_{ijnt} \in \{0, Q - 1\}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1. \quad (3.37)$$

La formulation complète est donc :

$$\begin{aligned}
\max z &= \sum_{t=1}^T v_{n,t} \\
\sum_{i=1}^W \sum_{j=1}^H b_{ijnt} + v_{nt} &\leq 1, \quad n = 1 \cdots N, t = 1 \cdots T \\
\sum_{n=1}^N b_{ijnt} &\leq 1, \quad i = 1 \cdots W, j = 1 \cdots H, t = 1 \cdots T \\
\sum_{n=1}^N b_{ijnt} &\geq \sum_{n=1}^N b_{ij+1nt}, \quad i = 1 \cdots W, j = 1 \cdots H, t = 1 \cdots T, \\
\sum_{i=1}^W \sum_{j=1}^H \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j+q-1 \leq H \\ l+q-1 \leq H}}^Q x_{ijklqt} &+ \sum_{i=1}^W \sum_{j=1}^H \sum_{n=1}^N y_{ijn} \leq 1, \quad t = 1 \cdots T, \\
\sum_{t=1}^T v_{nt} &\geq \sum_{t=1}^T v_{n+1t} + 1, \quad n = 1 \cdots N - 1, \\
v_{nt} &= \sum_{i=1}^W \sum_{j=1}^H \sum_{t'=1}^{t-1} y_{ijn} t', \quad n = 1 \cdots N, t = 1 \cdots T, \\
a_{ijn} &\leq b_{ijn}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1, \\
a_{ijn} &\leq \sum_{\substack{j'=0 \\ j-j' \geq 1}}^{Q-1} \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j-j'+q-1 \leq H \\ l+q-1 \leq H \\ q > j'}}^Q x_{ij-j'kl} t, \\
&\quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1, \\
a_{ijn} &\geq b_{ijn} + \sum_{\substack{j'=0 \\ j-j' \geq 1}}^{Q-1} \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j-j'+q-1 \leq H \\ l+q-1 \leq H \\ q > j'}}^Q x_{ij-j'kl} t - 1, \\
&\quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1, \\
c_{ijn} &\leq b_{ijn(t+1)}, \\
&\quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1, \\
c_{ijn} &\leq \sum_{\substack{j'=0 \\ j-j' \geq 1}}^{Q-1} \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j-j'+q-1 \leq H \\ l+q-1 \leq H \\ q > j'}}^Q x_{klij-j'qt} - 1, \\
&\quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1, \\
c_{ijn} &\geq b_{ijn(t+1)} + \sum_{\substack{j'=0 \\ j-j' \geq 1}}^{Q-1} \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j-j'+q-1 \leq H \\ l+q-1 \leq H \\ q > j'}}^Q x_{klij-j'qt} - 1, \\
&\quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1, \\
b_{ijn} &= b_{ijn(t-1)} + c_{ijn(t-1)} - a_{ijn(t-1)} - y_{ijn(t-1)}, \\
&\quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 2 \cdots T,
\end{aligned}$$

$$d_{ijnt} \leq Q \cdot a_{ijnt}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1, \text{ Avec : } Q - 1 \geq 1,$$

$$d_{ijnt} \leq \sum_{j'=j+1}^H \sum_{n'=1}^N a_{ij'n't}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1,$$

$$d_{ijnt} \geq \sum_{j'=j+1}^H \sum_{n'=1}^N a_{ij'n't} - Q \cdot (1 - a_{ijnt}), \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1,$$

$$e_{ijnt} = c_{ijnt} \sum_{j'=j+1}^H \sum_{n'=1}^N c_{ij'n't}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1,$$

$$e_{ijnt} \leq Q \cdot c_{ijnt}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1,$$

$$e_{ijnt} \leq \sum_{j'=j+1}^H \sum_{n'=1}^N c_{ij'n't}, \quad i = 1 \cdots W, j = 1 \cdots H, n = 1 \cdots N, t = 1 \cdots T - 1,$$

$$e_{ijnt} \geq \sum_{j'=j+1}^H \sum_{n'=1}^N c_{ij'n't} - Q \cdot (1 - c_{ijnt}), \quad i = 1, \dots, W, j = 1, \dots, H, n = 1, \dots, N, t = 1 \cdots T - 1,$$

$$\sum_{i=1}^W \sum_{j=1}^H d_{ijnt} = \sum_{i=1}^W \sum_{j=1}^H e_{ijnt}, \quad n = 1 \cdots N, t = 1 \cdots T - 1,$$

$$\sum_{i=1}^W \sum_{j=1}^H \sum_{n=1}^N a_{ijnt} = \sum_{i=1}^W \sum_{j=1}^H \sum_{k=1}^W \sum_{l=1}^H \sum_{\substack{q=1 \\ j+q-1 \leq H \\ l+q-1 \leq H}}^Q q \cdot x_{ijklqt}, \quad t = 1 \cdots T - 1,$$

$$x_{ijlqt} = 0, \quad i = 1 \cdots W, j, l = 1 \cdots H, q = 1 \cdots Q, t = 1 \cdots T, \text{ Avec : } j + q - 1 \leq H \text{ et } l + q - 1 \leq H,$$

$$\forall b_{ijnt}, x_{ijlqt}, y_{ijnt}, v_{nt}, a_{ijnt}, c_{ijnt} \in \{0, 1\},$$

$$\forall d_{ijnt}, e_{ijnt} \in \{0, Q - 1\},$$

$$i = 1 \cdots W; j, l = 1 \cdots H; n = 1 \cdots N; t = 1 \cdots T.$$

### 3.5 La résolution de PRC

Comme décrit dans la section suivante, le grand nombre de variables de la formulation mathématique introduite précédemment limite son champ d'application à des instances exécution considérable. Afin de surmonter cette limitation, nous proposons une approche heuristique qui nous permet d'obtenir des solutions de haute qualité au moyen de temps de calcul court.

L'heuristique proposée a pour objectif de récupérer les conteneurs demandés dès que possible tout en réduisant le nombre de conteneurs mal placés dans la rue (équation 3.2). Il convient de noter que plus le nombre de conteneurs mal placé dans la rue est faible, plus la probabilité d'exiger des mouvements de relocalisation supplémentaires est réduite.

Le pseudo-code de l'heuristique est décrit dans l'algorithme 5. Comme dans l'algorithme 5, il parcourt toutes les priorités demandées (lignes 1 à 18). A chaque étape, le conteneur suivant

à récupérer  $c^*$ , est trouvé (ligne 2). Si  $c^*$  est actuellement placé en haut d'une pile dans la rue, il peut être récupéré directement (ligne 17). Sinon, l'ensemble des conteneurs placés au-dessus de  $c^*$ , noté  $O(c^*)$  (équation 3.4), est vérifié (ligne 3). Voir le conteneur  $c^* = 1$  sur la figure 3.3, pour lequel  $O(1) = 5; 7; 13$ .

L'objectif principal est de réduire le nombre de conteneurs mal placés en construisant des piles avec uniquement des conteneurs bien placés (équation 3.13). C'est-à-dire des piles dans lesquelles tous les conteneurs sont placés en respectant leur ordre de priorité croissant, de telle sorte que le conteneur le plus haut possède la priorité relative la plus élevée. Dans cet objectif, l'heuristique identifie le conteneur placé au sommet de  $s(c^*)$ , noté  $O(c^*)^+$  (ligne 7). Par exemple dans la figure 3.3, nous obtenons  $O(c^*)^+ = 5$ .

La pile cible de  $O(c^*)^+$  est sélectionnée à l'aide de la fonction de notation suivante pour chaque pile  $s$  :

$$f(O(c^*)^+, s) = \begin{cases} 1, & \text{si } (h(s) + 1 > H) \cup (s = s(c^*)) \\ 0, & \text{sinon} \end{cases} \quad (3.38)$$

La fonction précédente  $f$  mesure l'attractivité des piles en estimant le temps pendant lequel elles nécessiteraient un mouvement de déménagement si  $O(c^*)^+$  est placé sur elles. Les piles sans emplacements vides et la pile source ne sont pas incluses dans l'ensemble des piles cibles réalisables, noté (ligne 8). La pile cible de  $O(c^*)^+$ , notée  $s^*$ , est sélectionnée de manière à minimiser le coût de l'opération et vide les piles d'avant, la plus proche pile l'arrière après l'avant parmi les piles  $\delta$  ayant la valeur la plus élevée et la plus proche (ligne 10), où  $\delta \in [1 : (W - 1)]$ . Le processus se termine lorsque tous les conteneurs initialement inclus dans  $O(c^*)$  sont repositionnés dans une autre pile. Après avoir repositionné  $O(c^*)^+$  dans  $s^*$ , il pourrait y avoir des créneaux vides, notés  $e(s^*) = H - (h(s^*) + 1)$ .

Ces créneaux peuvent être exploités afin de réduire le nombre de conteneurs mal placés. Avant de déplacer  $O(c^*)^+$ , l'heuristique identifie tous les conteneurs les plus haut placés d'une pile  $s (s \neq s(c^*))$  et sélectionne celui dont la priorité est la plus basse et qui satisfait la priorité d'ordre croissant. La raison est de chercher à maximiser le nombre de conteneurs potentiels à placer dans  $s^*$  sans nécessiter de nouvelles relocalisations à l'avenir. La figure 3.4 illustre la stratégie de remplissage. Dans ce cas, les piles cibles réalisables du conteneur  $O(1)^+ = 5$  sont 1, 3 et 5. La mesure d'attractivité indiquée par l'équation 3.38 est illustrée en haut de la figure. Avant de déplacer  $O(1)^+$  dans la pile 5, les conteneurs potentiels mal placés pour combler la place vide ( $e(5) = 2$ ) sont identifiés, les conteneurs 12, 8 et 16. Comme on peut le voir, le conteneur 12 est déplacé du fait que c'est avec la priorité la plus basse que l'ordre de priorité croissant est respecté, après le conteneur 8 est déplacé, ... etc.

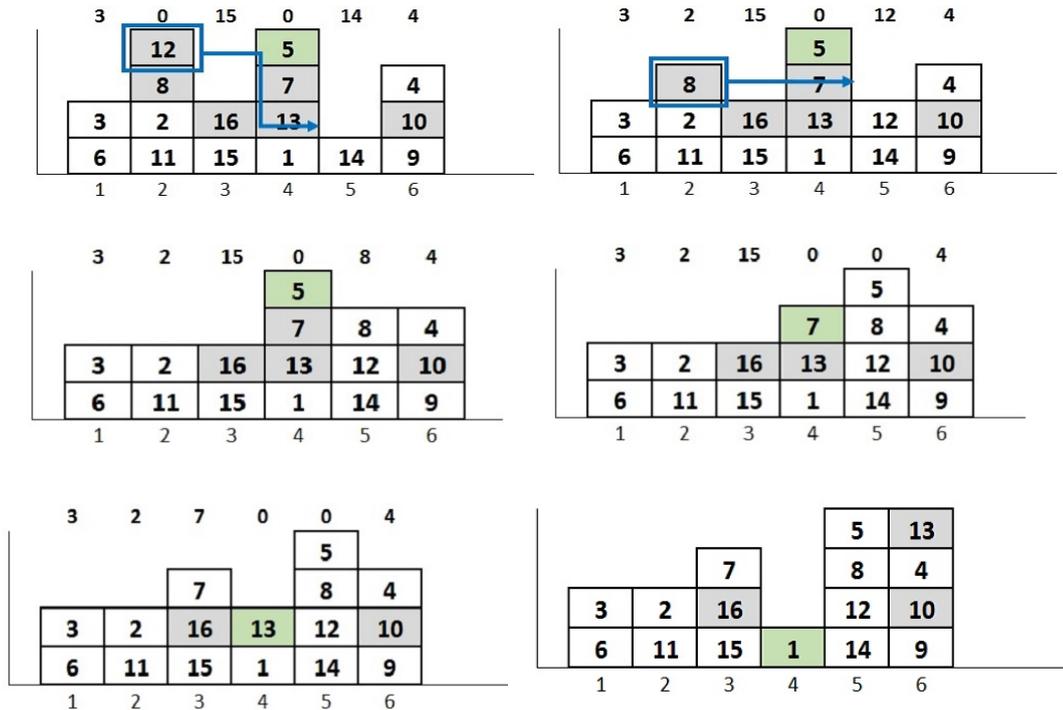


FIGURE 3.4: Exemple illustre la stratégie de remplissage

**Données :** L'état de parc avec les conteneurs à récupérer ;  $\delta$ , plage des piles cibles

**Résultat :** Solution du PRC

```

1 pour  $p \leftarrow 1$  à  $K$  faire
2    $c^* \leftarrow$  Trouver un conteneur de priorité  $p$ ;
3    $O(c^*) \leftarrow$  Conteneurs placés sur  $c^*$  (Équation 3.4);
4   tant que  $O(c^*) \neq 0$  faire
5      $placé \leftarrow Faux$ ;
6     tant que  $!placé$  faire
7        $O(c^*)^+ \leftarrow$  Le plus haut conteneur parmi  $O(c^*)$ ;
8        $\Phi \leftarrow$  Piles cibles réalisables pour  $O(c^*)^+$ ;
9       si  $\Phi \neq 0$  alors
10         $s^* \leftarrow$  Sélectionnez la pile de  $\Phi$  en utilisant  $\delta$ ;
11        Remplissez  $s^*$  avec des blocs mal placés;
12        Relocaliser  $O(c^*)^+$  de  $s(c^*)$  dans  $s^*$ ;
13      fin
14       $placé \leftarrow Vrai$ ;
15    fin
16  fin
17  Récupère le  $c^*$  du haut de sa pile;
18 fin

```

**Algorithme 5 :** Pseudo-code de l'approche heuristique visant à résoudre le PRC

## Conclusion

Dans ce chapitre, nous avons traité les deux parties du problème de stockage de conteneurs après une présentation du terminal portuaire de BMT et définie la problématique rencontrée. Une modélisation mathématique, dans laquelle on tient compte des contraintes physiques et opérationnelles du problème, est proposée. Trois principaux objectifs sont visés dans ce modèle, qui sont : la minimisation des distances à parcourir par les portiques gerbeurs sur pneus (RTGC), la minimisation du nombre de remaniements et la maximisation des conteneurs à retirer du parc. La solution que nous avons proposée pour la résolution du problème de stockage de conteneurs est illustrée en détail dans ce chapitre.

Deux parties du problème de stockage des conteneurs sont distinguées : la première partie est la sélection des conteneurs ayant pour objectif de maximiser le nombre des restitutions et la deuxième partie est la relocalisation des conteneurs pour que tous les conteneurs puissent être stockés sans provoquer des remaniements (futurs) inévitables dans le cas où la cour de stockage est trop chargée. Un algorithme génétique, qui est une méthode de résolution approchée est proposé pour résoudre la première partie du problème et un algorithme heuristique pour la seconde partie.

Des simulations numériques ont été faites pour valider ces deux algorithmes sont traités dans le chapitre suivant.

# 4

## Analyse et interprétation des résultats

### Introduction

Afin de valider la solution que nous avons proposée à l'entreprise *BMT* pour optimiser les mouvements des conteneurs et d'améliorer la productivité du parc, nous avons analysé les données de l'entreprise, généré les données qui nous manquent et interprété les résultats.

Dans ce chapitre nous allons présenter l'outil informatique que nous avons développé pour implémenter notre solution et validé avec une simulation et nous finirons par une interprétation des résultats.

### 4.1 Description de notre outil informatique

Les expériences présentées dans cette section ont été réalisées sur un *PC* doté d'un processeur *Intel(R) Core(TM) i3 CPU M380* à  $2.53GHz$  et de  $8 Go$  de *RAM*. Notre solution a été implémentée à l'aide de *Microsoft Excel 2016 MSO (16.0.4639.1000) 64 bits*, les algorithmes et les interfaces Homme-Machine ont été programmés par le langage d'applications *VBA*.

#### 4.1.1 Les indices utilisés

Les indices d'information et de performance que nous utilisons dans notre outil sont les suivants :

##### Les indices d'information

- Nombre total des conteneurs dans un bloc ( $N$ ),

- nombre des Rues ( $X$ ),
- nombre de Rangées ( $Y$ ),
- nombre des Niveaux ( $Z$ ),
- la première date pour qu'un conteneur soit retiré de bloc est  $date_{Min}$ ,
- la dernière date pour qu'un conteneur soit retiré de bloc est  $date_{Max}$ ,
- le coût de retirer un conteneur dans un rue ( $Cr$ ),
- le coût de shifting ou de relocalisation d'un conteneur dans un rue ( $Cs$ ),
- le coût de déplacement d'une RTGC d'une rue à l'autre ( $Cd$ ),
- le coût total des mouvements qui peuvent se produire dans une journée est le coût d'opération ( $C$ ).

### Les indices de performance pour chaque date

- Nombre de conteneurs mal placé dans un bloc ( $nbr\_Mp$ ),
- nombre de conteneurs pour la préparation ou la taille de la file d'attente ( $nbr\_Cp$ ),
- nombre des récupérations ( $nbr\_R$ ),
- nombre des remaniements ( $nbr\_Shifting$ ),
- nombre des piles non vides ( $nbr\_PnV$ ),
- nombre des piles mal arrangées ( $nbr\_PmA$ ),
- nombre des conteneurs mal placés par rapport au nombre total des conteneurs ( $nbr\_Mp/N$ ),
- nombre des conteneurs retirés du bloc sur la taille de la file d'attente ( $nbr\_R/nbr\_Cp$ ),
- nombre des piles mal arrangés sur le nombre des piles non vide ( $nbr\_PmA/nbr\_PnV$ ).

### Les indices de performance globale

- Le nombre moyen de conteneurs qu'on a récupérés,
- le nombre moyen de conteneurs moyenne qu'il faut les récupérer (l'objectif),
- le taux de satisfaction des demandes c'est Le nombre moyen des conteneurs qu'on a récupéré sur le nombre moyen des conteneurs qu'il faut récupérer.
- l'état actuel du bloc est égal au nombre de conteneurs dans le bloc sur le nombre total des emplacements,
- l'état initial c'est l'état d'un bloc au début du projet,
- nombre total des opérations (retrait et déplacement), pourcentage des retraits et pourcentage des déplacements.

À partir de ces indicateurs nous avons pu construire un tableau de bord utile pour mesurer l'efficacité de notre méthode.

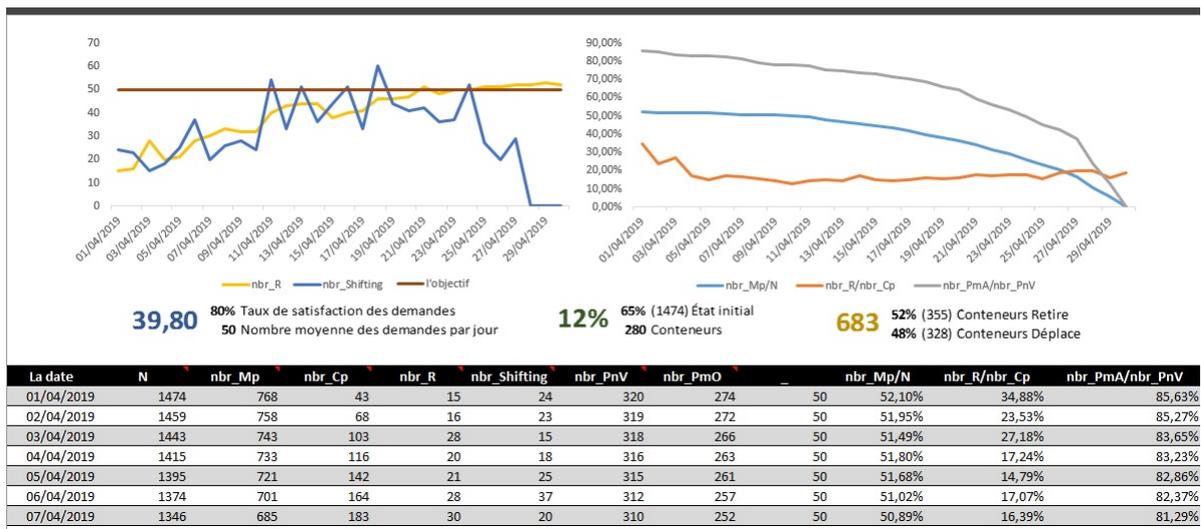


FIGURE 4.1: Tableau de bord de projet

#### 4.1.2 Les tableaux de donnée utilisées

**tblDataBloc** : Contient les informations sur les conteneurs qui séjournent dans ce bloc, les attributs de cette table sont les suivants :

- Le code qui identifie chaque conteneur (**Code/ID**),
- Les indices d'emplacement de chaque conteneur (**X, Y, Z**),
- La date de départ de chaque conteneur (**date de départ**).

Code/ID	X	Y	Z	date de départ
ERP_A_841BMT	8	4	1	25/04/2019
ERP_A_3461BMT	34	6	1	04/04/2019
ERP_A_2961BMT	29	6	1	20/04/2019
ERP_A_2921BMT	29	2	1	16/04/2019
ERP_A_4251BMT	42	5	1	28/04/2019
ERP_A_2951BMT	29	5	1	14/04/2019
ERP_A_5151BMT	51	5	1	17/04/2019
ERP_A_1031BMT	10	3	1	06/04/2019
ERP_A_131BMT	1	3	1	19/04/2019
ERP_A_1061BMT	10	6	1	28/04/2019
ERP_A_1351BMT	13	5	1	12/04/2019
ERP_A_3321BMT	33	2	1	25/04/2019
ERP_A_1311BMT	13	1	1	12/04/2019
ERP_A_3851BMT	38	5	1	28/04/2019

FIGURE 4.2: La table de données **tblDataBloc**

**tblEtatEmplacement** : Contient les informations de chaque emplacement dans le bloc, les attributs de cette table sont les suivants :

- Les indices d'emplacement (**X, Y, Z**),
- l'état de position c'est :
  - *Occupée*, par un conteneur,
  - *Libre*, on peut les utiliser,

- *Vide*, on ne peut pas les utiliser, car l'un des emplacements en dessous est Libre.
- le numéro de la pile (**N° pile**),
- la date de départ de chaque conteneur (**date de départ**),
- l'état du conteneur, si le conteneur a été bien placé ou mal placé (**L'état de conteneur**),
- priorité de quitter la zone (**Priorité**),
- nombre de conteneurs mal placés au-dessus de ce conteneur (**O+**),
- nombre de conteneurs mal placés au-dessus de ce conteneur avant de le récupéré (**Accessibilité**),
- première variable de décision si le conteneur est : (1) *Accessible* ou (0) *Inaccessible* (**VDA**),
- deuxième variable de décision si le conteneur quitte le bloc (1) ou non (1) (**Variable de décision**),
- le code qui identifie chaque conteneur (**nom**).

X	Y	Z	L'état de position	N° pile	Date de départ	L'état de conteneur	Priorité	O+	Accessibilité	VDA	Variable de décision	nom
1	1	1	Occupée	1	10/04/2019	Bien Placé	10	2	2	0		ERP_A_111BMT
1	1	2	Occupée	1	06/04/2019	Bien Placé	6	2	2	0		ERP_A_112BMT
1	1	3	Occupée	1	28/04/2019	Mal Placé	28	0	0	0		ERP_A_113BMT
1	1	4	Occupée	1	11/04/2019	Mal Placé	11	0	0	0		ERP_A_114BMT
1	1	5	Libre	1					0	0	0	
1	1	6	Vide	1					0	0	0	
1	1	7	Vide	1					0	0	0	
1	2	1	Occupée	2	14/04/2019	Bien Placé	14	2	2	0		ERP_A_121BMT
1	2	2	Occupée	2	20/04/2019	Mal Placé	20	1	1	0		ERP_A_122BMT
1	2	3	Occupée	2	27/04/2019	Mal Placé	27	0	0	0		ERP_A_123BMT
1	2	4	Occupée	2	04/04/2019	Bien Placé	4	0	0	0		ERP_A_124BMT
1	2	5	Libre	2					0	0	0	
1	2	6	Vide	2					0	0	0	
1	2	7	Vide	2					0	0	0	
1	3	1	Libre	3					0	0	0	
1	3	2	Vide	3					0	0	0	

FIGURE 4.3: La table de données **tblEtatEmplacement**

### 4.1.3 L'interface homme-machine (IHM)

Les IHM permettent principalement d'afficher des informations d'une façon visuelle pour permettre à l'utilisateur de superviser un processus et d'interagir avec la machine.

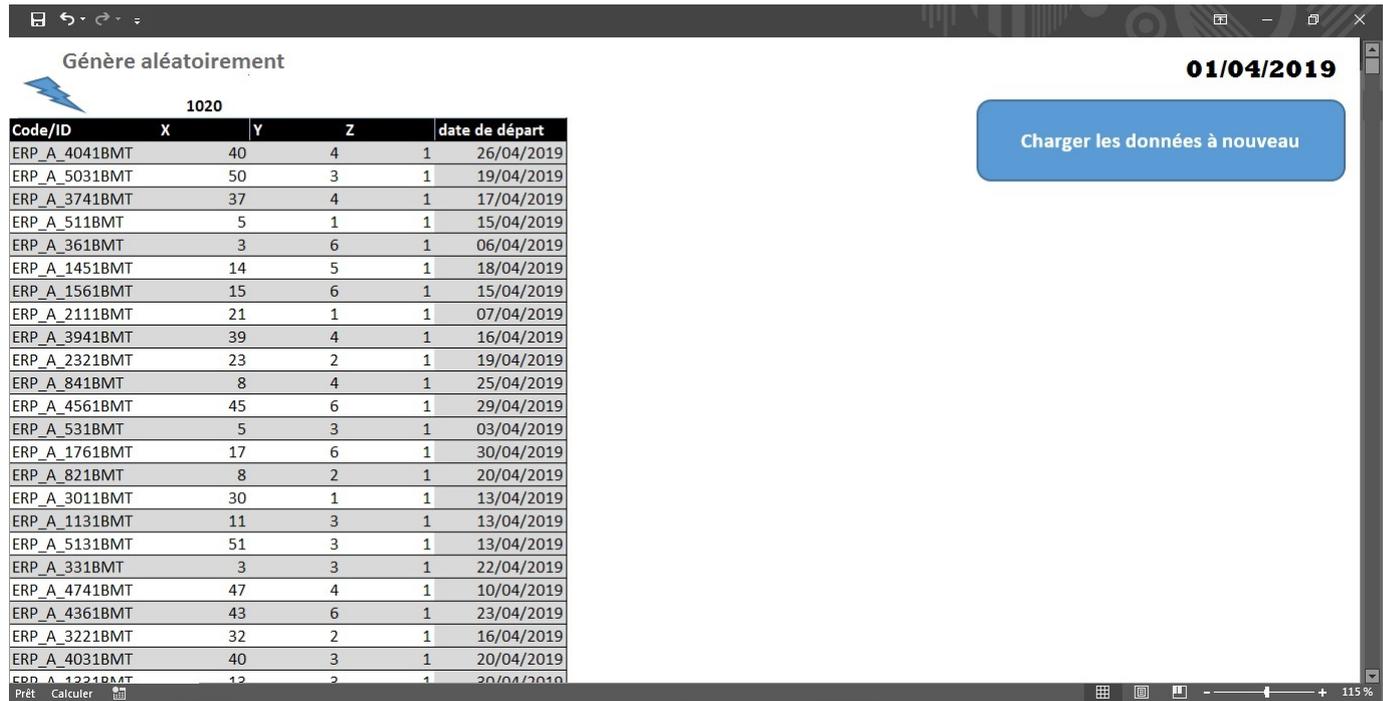


FIGURE 4.4: L'interface du générateur

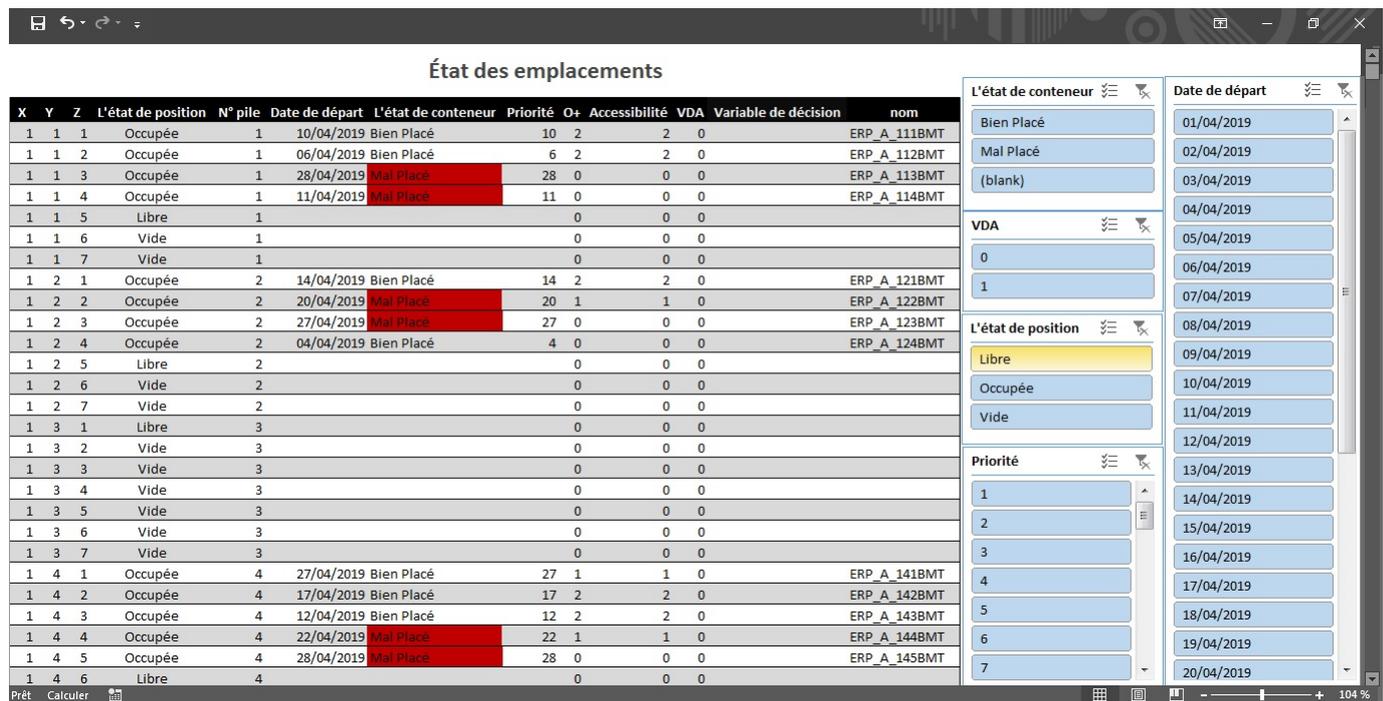


FIGURE 4.5: L'interface (État des emplacements)

**Les données du problème**

P	R	A	VD	Ci
60	2	0	0	-
101	3	1	0	-
143	4	5	0	-
176	5	0	0	-
331	8	3	0	-
371	9	1	0	-
415	10	1	0	-
537	13	1	0	-
551	14	0	0	-
619	15	2	0	-
648	16	0	0	-
660	16	1	0	-
745	18	0	0	-
746	18	0	0	-
809	20	2	0	-
870	21	0	0	-
992	24	0	0	-
1038	25	4	0	-
1159	28	0	0	-
1160	28	1	0	-
1297	31	2	0	-
1320	32	1	0	-
1336	32	2	0	-
1423	34	5	0	-
1489	36	1	0	-
1491	36	0	0	-
1510	36	1	0	-

Min(R)	0
Somme(Xi)	0
Max(R)	0
Ct	-
Le coût de récupération Cr	660,00
Le coût de shifting Cs	990,00
Le coût de déplacement d'une RTG du rue a l'autre Cd	330,00
Le coût total de l'opération C	36 960,00

FIGURE 4.6: L'interface (Les données du problème)

**Les paramètres de l'algorithme génétique**


**Université Abderrahmane Mira de Béjaia**  
**Département Recherche Opérationnelle**  
**2018/2019**

Types de valeur	Ensembles par génération :	41
<input checked="" type="radio"/> Entiers seulement	Valeur minimale autorisée :	0
<input type="radio"/> Réels	Plus haute valeur autorisée :	1
Première génération	Plage de jeu d'entrée :	Data!\$D\$5:\$D\$45
<input checked="" type="radio"/> Nombres aléatoires	Adresse de la cellule de score final :	Data!K3
<input type="radio"/> De la course précédente	Elites (ensembles immortels) par génération :	20
Type d'objectif	Parents potentiels par enfant :	20
<input checked="" type="radio"/> Maximiser le résultat	Génération demandées :	2
<input type="radio"/> Minimiser le résultat	Max. mutations par série :	2

Entrez vos paramètres pour cette exécution, puis cliquez sur Démarrer (ou Annuler).  
 DÉPLACEZ VOTRE SOURIS SUR TOUTE ZONE D'ENTREE pour obtenir des informations sensibles au contexte.

FIGURE 4.7: L'interface (Les paramètres de l'algorithme génétique)

Nombre d'enregistrement: 2514

La liste des mouvements à faire

Type de'action	Nom	X	Y	Z	X1	Y1	Z1	Description
Retire	ERP_A_2867BMT	28	6	7				
Déplace	ERP_A_3027BMT	30	2	7	30	4	7	
Retire	ERP_A_3026BMT	30	2	6				
Retire	ERP_A_3025BMT	30	2	5				
Retire	ERP_A_3237BMT	32	3	7				
Retire	ERP_A_3717BMT	37	1	7				
Déplace	ERP_A_3817BMT	38	1	7	38	2	6	
Retire	ERP_A_3816BMT	38	1	6				
Déplace	ERP_A_3857BMT	38	5	7	38	6	5	
Déplace	ERP_A_3856BMT	38	5	6	38	6	6	
Déplace	ERP_A_3855BMT	38	5	5	38	6	7	
Déplace	ERP_A_3854BMT	38	5	4	38	1	6	
Retire	ERP_A_3853BMT	38	5	3				
Retire	ERP_A_4117BMT	41	1	7				
Déplace	ERP_A_4247BMT	42	4	7	42	5	7	
Retire	ERP_A_4246BMT	42	4	6				
Retire	ERP_A_4517BMT	45	1	7				
Déplace	ERP_A_4565BMT	45	6	5	45	4	7	
Retire	ERP_A_4564BMT	45	6	4				
Déplace	ERP_A_4637BMT	46	3	7	46	5	7	
Déplace	ERP_A_4636BMT	46	3	6	46	6	3	
Déplace	ERP_A_4635BMT	46	3	5	46	6	4	
Déplace	ERP_A_4634BMT	46	3	4	46	6	5	
Retire	ERP_A_4633BMT	46	3	3				
Déplace	ERP_A_4957BMT	49	5	7	49	4	6	
Déplace	ERP_A_4956BMT	49	5	6	49	4	7	
Déplace	ERP_A_4955BMT	49	5	5	49	2	7	
Retire	ERP_A_4954BMT	49	5	4				
Déplace	ERP_A_5117BMT	51	1	7	51	6	4	

Type de'action

- Déplace
- Retire

FIGURE 4.8: L'interface (La liste des mouvements à faire)

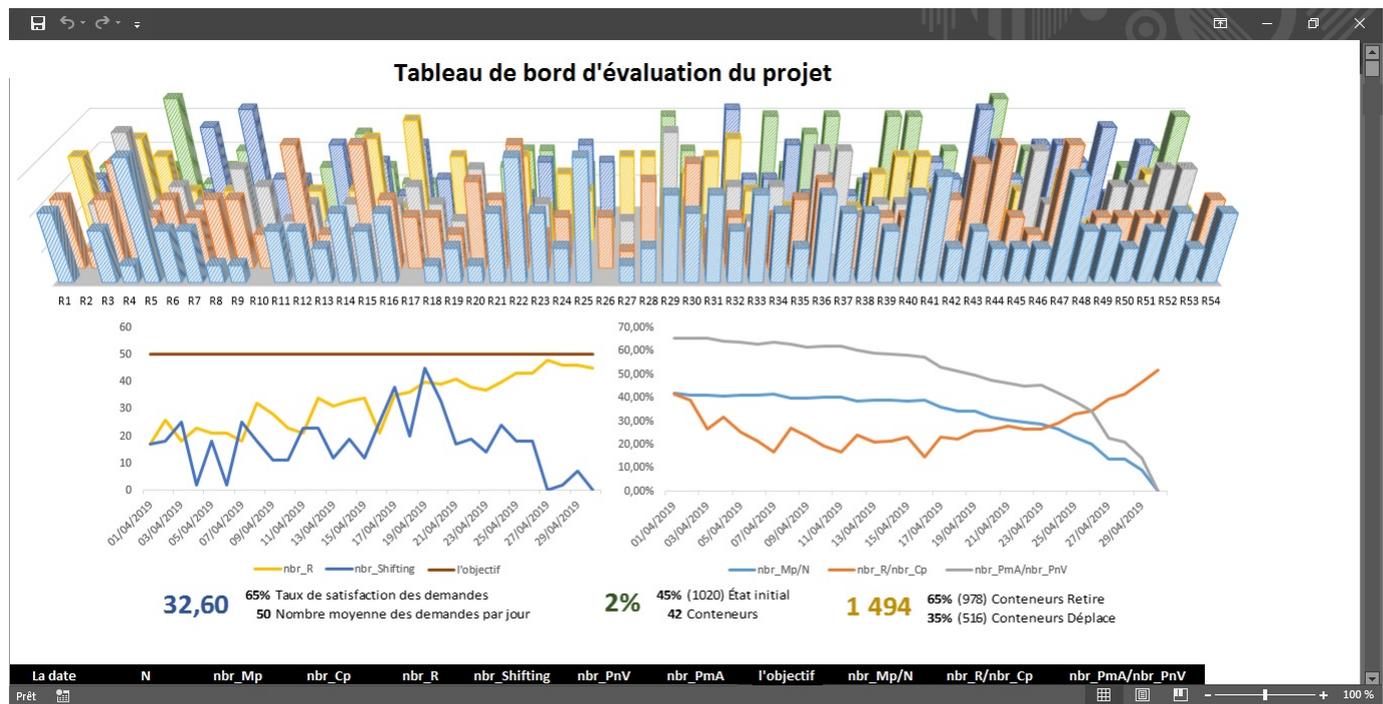


FIGURE 4.9: L'interface (Tableau de bord d'évaluation du projet)

## 4.2 Collecte de données

Les données que nous avons pu collecter concernant les mouvements des RTGCs dans le parc, les états des blocs et les coûts des opérations, ne sont pas suffisamment fiables pour les utiliser totalement dans notre méthode, pour cela nous gardons les données fiables et nous remplaçons les données non fiables par des données qui satisfont les contraintes de stockage (l'emplacement des conteneurs). Pour les autres types de données, nous nous sommes basés sur les sources d'information délivrées par le département des opérations pour générer les données manquantes. Exemple : le nombre moyen d'opérations par heure dans le mois d'avril est 6,6 opérations. Autre exemple d'anomalie dans les données délivrées par le système d'information de BMT, l'inexistence des dates de sorties des conteneurs, ce qui peut être une thématique de mémoire. Pour ces données, nous les avons générées aléatoirement dans une durée d'un mois.

### 4.2.1 Les données du générateur que nous avons construit

Pour appliquer notre méthode décrite précédemment à trois jeux de données, nous avons utilisé notre générateur pseudo-aléatoire, pour  $n$  conteneurs, il génère une date de départ entre ( $DateMin$ ,  $DateMax$ ) et un emplacement tout en respectant les contraintes de stockage.

Ces données sont composées de trois paramètres utilisés pour générer les trois applications : Nombre de conteneurs à générer aléatoirement ( $N$ ) et Date de départ de chaque conteneur à générer aléatoirement entre ( $DateMin$  et  $DateMax$ ).

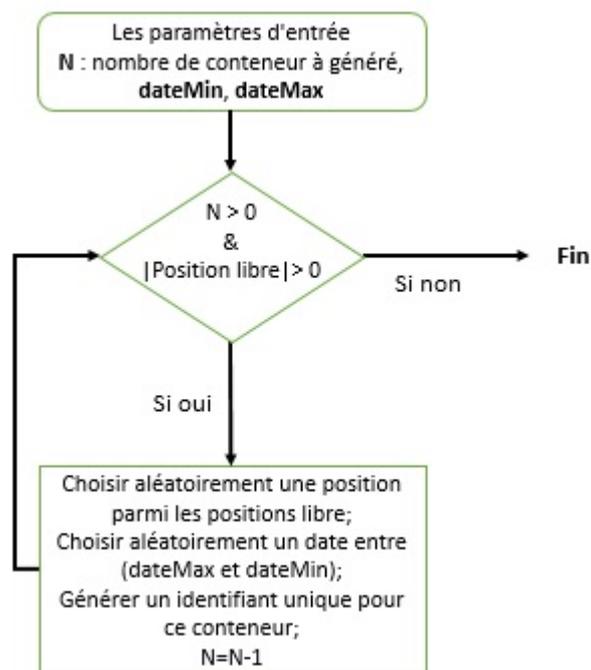


FIGURE 4.10: L'organigramme du générateur

Ces trois jeux de données nous permettront de mesurer l'efficacité de notre méthode par comparaison et également de voir l'évolution des indicateurs de performance opérationnelle.

Les paramètres des trois applications sont les suivants : Pour les trois applications ( $X = 54$ ,  $Y = 6$  et  $Z = 7$  qui indique 2268 emplacements,  $dateMin = 01/04/2019$ ,  $dateMax = 30/04/2019$ ),

La première application :  $N = 1020$  qui représente 45% du nombre total d'emplacements,

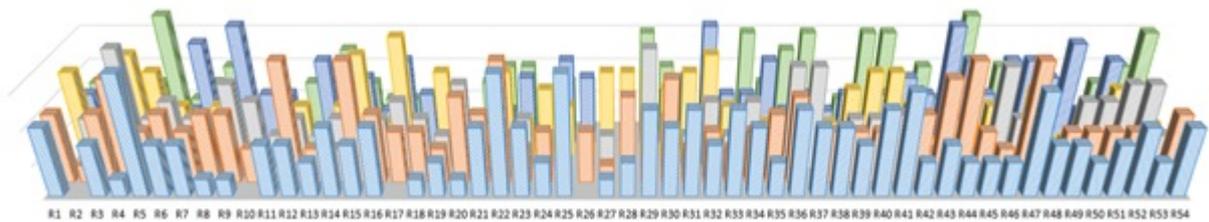


FIGURE 4.11: La première application

La deuxième application :  $N = 1474$  qui représente 65% du nombre total d'emplacements,

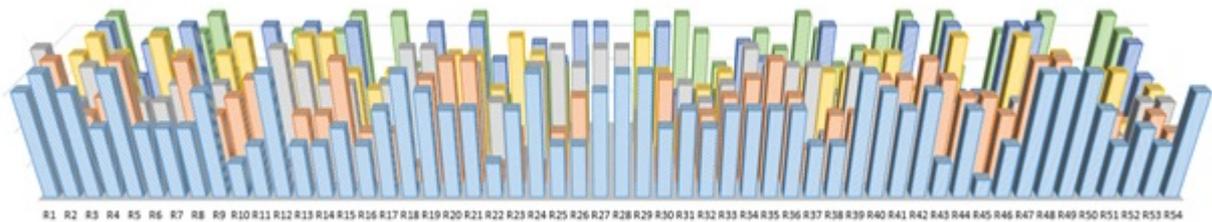


FIGURE 4.12: La deuxième application

La troisième application :  $N = 2154$  qui représente 95% du nombre total d'emplacements,

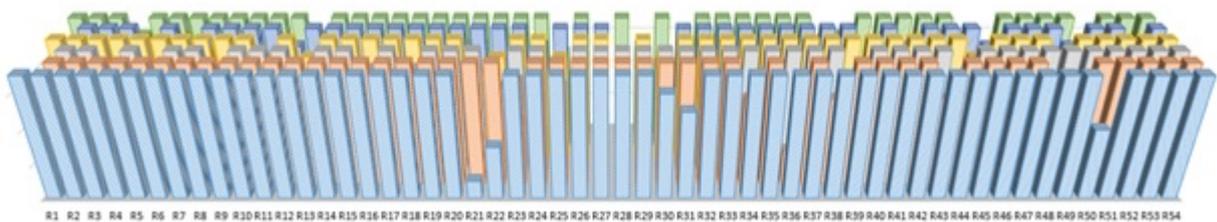


FIGURE 4.13: La troisième application

### 4.3 Interprétation des résultats

Après la simulation des données générées, nous avons obtenu les résultats suivants sur le tableau de bord d'évaluation du projet de chaque application.

Dans les tableaux suivants, nous remarquons premièrement la croissance du nombre des récupérations des conteneurs ( $nbr\_R$ ) au fil du temps dans les trois applications, et ce nombre se rapproche de l'objectif, ce qui mesure l'efficacité de l'algorithme génétique que nous avons adopté. La deuxième remarque c'est la décroissance du nombre de relocalisations ( $nbr\_Shifting$ ) après quelques jours, et la décroissance de  $nbr\_Mp/N$  et de  $nbr\_PmA/nbr\_PnV$  dans les trois applications, ce qui montre l'efficacité de la stratégie de remplissage que nous avons proposé sous forme d'un algorithme heuristique. La troisième remarque c'est la croissance de  $nbr\_R/nbr\_Cp$  après quelques jours pour les deux premières applications (45%, 65%), mais pas pour la troisième application (95%) à cause de la densité de son état initial qui est très élevée.

La dernière remarque concerne les autres indicateurs de performance de chaque application, dont la valeur prouve le succès de la solution proposée, comme le taux de satisfaction des demandes est supérieure à 65% et le taux de récupération est supérieur au taux de relocalisation pour toutes les applications (65% > 35%, 52% > 48%, 51% > 49% respectivement).

**Tableau de bord d'évaluation de la première application**

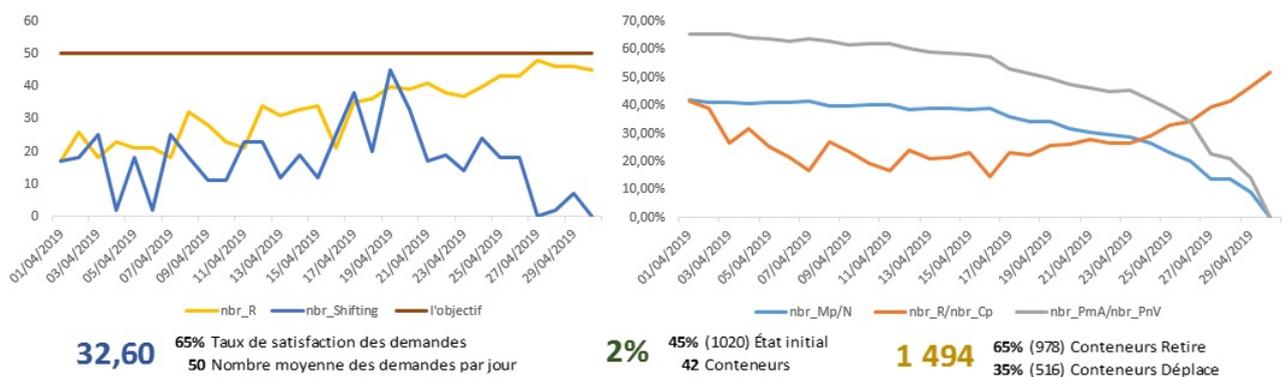


FIGURE 4.14: Tableau de bord d'évaluation de la première application

**Tableau de bord d'évaluation de la deuxième application**

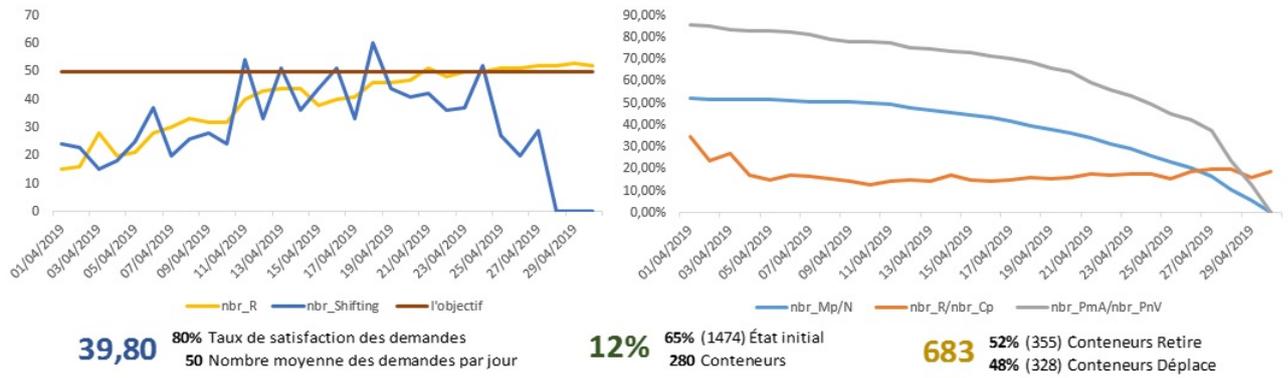


FIGURE 4.15: Tableau de bord d'évaluation de la deuxième application

Tableau de bord d'évaluation de la troisième application

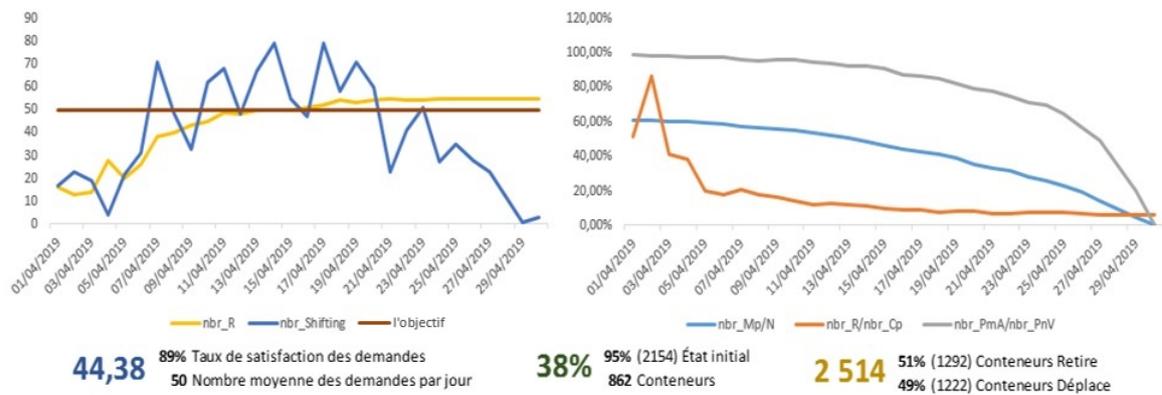


FIGURE 4.16: Tableau de bord d'évaluation de la troisième application

## Conclusion

Dans ce chapitre nous avons présenté l'outil informatique que nous avons développé et utilisé pour simuler les trois applications, afin de valider la solution que nous avons proposé dans ce mémoire.

# Conclusion générale et perspective

## Conclusion générale

L'objet de notre projet de recherche concerne les problèmes d'optimisation dans la gestion des conteneurs chez *BMT*. après d'un diagnostic profond du fonctionnement de l'entreprise dans les deux cotés administratif et opérationnel durant la période de notre stage, nous avons proposé une solution composée de deux modèles l'un est linéaire et l'autre non-linéaire en nombre entier, de très grandes tailles, qui rendent leur résolution difficile avec les méthodes exactes.

Notre travail s'est attaché dans un premier temps à montrer la complexité du problème de gestion des conteneurs dans un terminal maritime et modélisé mathématiquement ce problème dans le cas de *BMT*. Et dans un second temps nous résoudrons ce problème et en avons évalué l'impact en termes de faisabilité et de performance.

Pour aide les dirigeants de l'entreprise *BMT* a mise à l'exploitation et valide notre solution, nous avons développé un prototype de système d'aide à la décision pour évaluations des performantes de la gestion des conteneurs et valide notre solution.

## Perspective

Néanmoins, les résultats de ce modeste travail constituent les bases d'un travail à poursuivre et à améliorer pour une étude beaucoup plus approfondie qui pourra faire l'objet d'une thèse de doctorat et plusieurs projets de master.

Ainsi, les perspectives futures sont dans un premier temps l'estimation des dates de départ de chaque conteneur, c-t-d l'estimation fonctionnelle, une fonction qui estime les dates de départ de chaque conteneur à partir de ces informations.

Pour réaliser cette estimation fonctionnaire, il est nécessaire de disposer d'informations statistiques fiables sur l'historique des clients et leurs marchandises, afin d'en connaître les caractéristiques de leurs réclamations et de leurs fiabilités pour passer ensuite à la modélisation.

Ces informations statistiques doivent être synthétisées par un système d'information fiable qu'il collecte des informations en temps réel avec un système de nettoyage des données sur plus de trois ans en allant jusqu'à plus de 10 ans. Ensuite, il faut une étude sérieusement et rigoureuse en gestion des coûts au niveau de département finance.

Après ces deux projets, il y a plusieurs thématiques de recherche opérationnelle, comme l'affectation des conteneurs à des piles bien arrangées au cout minimum, l'ordonnancement des tâches, l'affectation des camions à des blocs ... etc.

Nous espérons avoir posé la première pierre dans le domaine de la gestion des conteneurs à *BMT* ce qui donnera une impulsion à la recherche locale dans cet axe, en vue d'une prise de conscience de « **La Recherche Opérationnelle** », de l'exploitation des données et de l'importance d'information.

# Bibliographie

- [1] C. Boughani. Cours d'optimisation combinatoire. Université de Béjaia, 2016.
- [2] A. Boulghorba and S. Larbi. L'impact de l'extension du port sur les performances du terminal à conteneurs BMT via les RdP. Master's thesis, Université de Béjaia, 2016.
- [3] A. Bouzekri and K. Fellah. Analyse du fonctionnement et évaluation des performances de la nouvelle zone IOB de BMT. Master's thesis, Université de Béjaia, 2017.
- [4] F. Broeze. The globalisation of the oceans : Containerisation from the 1950s to the present. St. Johns, Canada : International Maritime Economic History Association, 2002.
- [5] K. Chebil. *Optimisation des mouvements des conteneurs dans un terminal maritime*. PhD thesis, Université de Montréal, 2011.
- [6] B. J. Cudahy. *Box Boats : How container ships changed the world*. Fordham UP, New York, 2006.
- [7] C. Exposito-Izquierdo, B. Melian-Batista, and J. Moreno-Vega. Optimization model and heuristic approach for blocks retrieval processes in warehouses. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling*, pages 111–119, 2014.
- [8] M. Gen and R. Cheng. Genetic algorithms and engineering optimization. *Industrial Engineering / Manufacturing*, 2007.
- [9] G. Hans-Otto and K. H. Kim, editors. *Container Terminals and Automated Transport Systems : Logistics Control Issues and Quantitative Decision Support*. Springer, 2005.
- [10] J. H. Holland. Adaptation in natural and artificial systems. *University of Michigan Press, Computers*, 1975.
- [11] X. Jiang, L. H. Lee, Y. Han, and K. C. Tan. A container yard strategy for improving land utilization and operation efficiency in a transshipment hub port. *European journal of operational research*, 221 :64–73.
- [12] L. Lagarde. La science du management quantitatif. In *La Recherche Opérationnelle en France*. ROADEF, France, Octobre 2011.
- [13] M. Levinson. *The box : How the shipping container made the world smaller and the world economy bigger*. 2006.

- [14] F. Meisel. *Seaside Operations Planning in Container Terminals*. Contributions to Management Science. Physica Verlag, 2009.
- [15] A. Mesrour and R. Tebri. Contribution à la modélisation et l'évaluation des performances du terminal à conteneurs bmt. Master's thesis, Université de Béjaia, 2016.
- [16] R. Moussi. *Genetic Algorithms and Engineering Optimization*. PhD thesis, Université du Havre, 2012.
- [17] N. Ndiaye. *Algorithmes d'optimisation pour la résolution du problème de stockage de conteneurs dans un terminal portuaire*. PhD thesis, Université de Havre, 2015.
- [18] M. Sakarovitch. *Optimisation combinatoire : programmation discrète*. Hermann, 1984.
- [19] D. Sculli and C. F. Hui. Three dimensional stacking of containers. *Omega*, 16 :585–594, 1988.
- [20] B. Slack. Shipping and ports in the twenty-first century : Globalization, technical change and the environment. In *Corporate realignment and the global imperatives of container shipping*, pages 25–39. Routledge, London, 2004.
- [21] M. Stopford. *Maritime Economics*. Routledge, New York, 2009.
- [22] BMT Support. Bmt bejaia mediterranean terminal, avril 2019.
- [23] Microsoft Support. <https://support.office.com/fr-001/excel>, avril 2019.
- [24] L. Wolsey. *Integer Programming*. Series in Discrete Mathematics and Optimization. Wiley, 1998.
- [25] K. Yachba. *Vers une contribution dans le transport maritime de marchandises : Optimisation de placement des conteneurs dans un port maritime*. PhD thesis, Université de Tlemcen, 2017.

---

## Résumé

*La demande du parcs à conteneurs de l'entreprise BMT croît considérablement plus rapidement que l'offre de terrains disponibles ; par conséquent, les conteneurs sont généralement empilés haut afin de mieux utiliser l'espace disponible dans les parcs à conteneurs. Toutefois, lors du processus de récupération des conteneurs, il peut être nécessaire de redistribuer d'une manière non productive pour déplacer les conteneurs empilés au-dessus d'un conteneur cible. La récupération des conteneurs est directement liée à l'efficacité opérationnelle des terminaux. Étant donné que le secteur est devenu de plus en plus concurrentiel, il est devenu indispensable d'adopter une approche systématique de la récupération des conteneurs. Dans cette étude, Nous développons deux algorithmes (Génétique et Heuristique) adapté avec la stratégie de stockage chez BMT capable de générer des plans de travail réalisables pour les équipements installés dans les parcs à conteneurs, afin de maximiser le nombre de conteneurs à récupérer et minimiser le nombre de remaniement des conteneurs tout en tenant compte du coût des opérations. La méthodologie prend en compte le cas où les conteneurs sont regroupés en fonction de leur ordre de récupération. Des expériences numériques complètes révèlent que la méthode est plus rapide.*

**Mots clefs :** La conteneurisation, Algorithme Génétique, Algorithme Heuristique, la stratégie de stockage des conteneurs, BMT.

## Abstract

*In the storing surface of a Containers parc of the BMT society increases exponentially, they are generally stacked higher in order to better use the available surface of the of the containers park. However, during the picking process, of containers, it is necessary to redistribute them in an unproductive way in order to move the stacked containers over the target one. The operation of containers retrieval is tightly related to the operational efficiency of terminals. Since the domain has become more and more competitive, it is urgent to adopt a systematic approach for the problem of containers retrieval. In the present work, we have developed two algorithms (genetic and heuristic) adapted to storing strategy at BMT able to generate feasible work plans for the used equipments, and maximize the number of container to pick and minimize the number of containers relocations taking into account the work cost of equipments. The methodology considers the case where the containers are grouped according to their picking order. numerical experiments that the method is faster and improves some performance indices of the parc.*

**Key words :** conteneurization, genetic algorithms, heuristic algorithms, containers storing strategy, BMT.

---



