

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION  
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES  
--T L E M C E N--



المدرسة العليا في العلوم التطبيقية  
École Supérieure en  
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية  
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme d'Ingénieur

Filière : Génie industriel

Spécialité : Management industriel et logistique

Présenté par :

BOUREBA Ismail & BENHAMLA Mohamed Abd Eldjalil

Thème

**Analyse et amélioration de système de  
stockage dans un terminal portuaire**

Soutenu le 13 juillet 2021 devant le jury composé de :

M. Fouad MALIKI	MCB	Président	ESSA. Tlemcen
M. Mohamed BENNEKROUF	MCB	Directeur de Mémoire	ESSA. Tlemcen
M. Mostafa BRAHAMI	MCB	Examinateur	ESSA. Tlemcen
Mme. Amina OUHOUD	MCB	Examinateur	ESSA. Tlemcen

Année universitaire : 2020 /2021

بِسْمِ اللَّهِ

الرَّحْمَنِ

الرَّحِيمِ

# **REMERCIEMENT**

*Avant de commencer la présentation de ce travail*

*Nous remercions en premier lieu DIEU le tout puissant de nous avoir donné le courage, la volonté de réaliser et terminer ce travail.*

*Nous adressons toutes nos gratitudees à notre encadreur M.BENNEKROUF Mohammed, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter notre réflexion, malgré ses lourdes et bien nombreuses charges.*

*Un merci plein de respect, d'amour et d'honneur à M. Mohammed TOUNSSI et M. Rabie qui nous ont accompagnés durant notre stage, ils nous ont donné de leur temps pour nous faire bénéficier de toutes les informations se rapportant aux activités qui se font au niveau de terminal*

*Nous remercions le développeur BOUREBA Walid qui nous a aidés à la programmation. nous remercions notre cher ami BEDDAD Abderrahim pour l'aide qu'il nous a apportée et les encouragements qu'il nous a donné.*

*Ainsi à notre responsable de Filière M. Fouad MALIKI pour son soutien et son encouragement tout au long de notre parcours, ainsi que tous les professeurs qui nous ont enseigné durant notre cursus scolaire au niveau de l'ESSAT.*

## ***DEDICACE***

*Je dédie ce modeste travail à mes chers parents, à eux l'amour et le respect, le  
résultat et le sacrifice de plusieurs années d'étude.*

*A mes chers frère Mohamed Amine et Walid*

*A mes chers sœurs Wafaa et Amina*

*A mes chers neveux Ibrahim Khalil, Ahmed et Imran*

*A mon ami Abderrahim Beddad*

*Aux personnes qui m'ont toujours soutenu et encouragé.*

**Ismail**

## ***DEDICACE***

*Je dédie ce mémoire*

*A mes parents qui m'a soutenue et encouragé durant ces années d'études.*

*Qu'ils trouvent ici le témoignage de ma profonde reconnaissance.*

*A mes frères, ma grand-mère et Ceux qui ont partagé avec moi tous les moments*

*d'émotion lors de la réalisation de ce travail. Ils m'ont chaleureusement*

*supporté et encouragé tout au long de mon parcours.*

*A ma famille, mes proches et à ceux qui me donnent de l'amour et de la vivacité.*

*A tous mes amis qui m'ont toujours encouragé, et à qui je souhaite plus de*

*succès.*

**Mohamed**

**Table des matières**

**&**

**Liste des figures**

**&**

**Liste des tableaux**

## TABLE DES MATIERES

INTRODUCTION GENERALE.....	01
----------------------------	----

### Chapitre I : Conteneurisation, ports et terminaux à conteneur

I.A.	Conteneurisation .....	04
I.A.1.	Historique.....	04
I.A.2.	Les conteneurs.....	04
I.A.2.1	Description et propriété du conteneur .....	04
I.A.2.2.	Type des conteneurs .....	04
I.A.2.3.	Structure de conteneur.....	08
I.A.2.4.	Dimensions des conteneurs .....	09
I.A.3.	Avantage de la conteneurisation .....	10
I.B.	Ports.....	11
I.B.1.	Définition .....	11
I.B.2.	Types des ports.....	11
I.B.3.	Rôle des ports et leurs évolutions.....	12
I.C.	Terminal à conteneur.....	13
I.C.1.	Définition .....	13
I.C.2.	Rôle du terminal à conteneur .....	13
I.C.3.	Organisation du terminal à conteneur .....	13
I.C.4.	Equipements du terminal à conteneur .....	14
I.C.5.	Configuration de la zone de stockage.....	16

### Chapitre II : Problème de stockage de conteneurs

II.1.	Introduction .....	19
II.2.	Définition du problème du stockage de conteneurs (PSC).....	20
II.3.	Les différentes stratégies de stockage.....	20
II.3.1.	Ségrégation et Non-ségrégation.....	20

## TABLE DES MATIERES

II.3.2.	Groupage et Dispersion .....	21
II.3.3.	Stockage direct et Stockage indirect .....	22
	II.3.4. Priorité aux déchargements et Priorité aux chargements .....	22
II.4.	Méthodes de résolution .....	24
	II.4.1. Études s'appuyant sur la simulation pour résoudre le problème de stockage de conteneurs .....	24
II.4.2.	Programmation dynamique .....	27
II.4.3.	Utilisation de la théorie des graphes .....	30
II.4.4.	Algorithmes génétiques .....	33
II.4.5.	L'algorithme des colonies de fourmis .....	37
Chapitre III : L'entreprise portuaire d'Oran		
III.1.	Introduction .....	42
III.2.	Présentation de l'entreprise .....	42
III.3.	Le terminal à conteneur du port d'Oran .....	43
III.4.	Les procédures administratives .....	44
III.4.1.	Démarche import .....	44
III.4.2.	La procédure export .....	47
Chapitre IV : Modélisation et résolution du problème de stockage de conteneurs		
IV.1.	Introduction .....	49
IV.2.	Description du problème .....	49
IV.3.	La complexité du problème .....	50
IV.4.	Approche de résolution .....	50
	IV.4.1. La résolution de PSC par l'utilisation d'un algorithme génétique .....	50
IV.4.2.	Le modèle mathématique .....	52
IV.4.3.	Le processus de résolution .....	54
IV.5.	La résolution de PSC par l'utilisation d'un algorithme « proposé » appliqué sur le programme Python .....	57

## TABLE DES MATIERES

IV.5.1. Processus de résolution .....	57
IV.5.2. Algorithme .....	58
IV.5.3 Description du code Python .....	62
IV.5.4. Exemple de simulation.....	65
IV.5.5. Conclusion .....	66
CONCLUSION GENERALE .....	67
BIBLIOGRAPHIE .....	68

---

**LISTE DES FIGURES**

Figure I.1 : Conteneur Dry standard.....	05
Figure I.2 : Conteneur Open Top .....	05
Figure I.3 : Conteneur REFFER.....	06
Figure I.4 : Conteneur Open Side.....	07
Figure I.5 : Conteneur Citerne.....	07
Figure I.6 : Conteneur Flat Rack .....	08
Figure I.7 : Les coins supérieurs et inférieurs de conteneur.....	09
Figure I.8 : Schématisation d'un terminal à conteneur .....	14
Figure I.9 : Grue sur rail.....	14
Figure I.10 : Grue sur pneu .....	14
Figure I.11 : Rech Stacker.....	15
Figure I.12 : Cavalier gerbeur .....	15
Figure I.13 : RTGc .....	15
Figure I.14 : RMGc .....	16
Figure I.15 : Un terminal qui utilise des RTGc.....	17
Figure I.16 : Un terminal qui utilise des RMGc.....	17
Figure I.17 : Un terminal qui utilise des cavaliers gerbeurs.....	17
Figure II.1 : Croisement en un point .....	34
Figure II.2 : Croisement en deux points.....	34
Figure III.1 : L'organigramme de l'EPO.....	43
Figure III.2 : La procedure import.....	46
Figure III.3 : Conteneurs vides.....	47
Figure III.4 : Conteneurs pleins.....	47
Figure IV.1 : Structure de l'AG.....	50
Figure IV.2 : L'organisation de la zone de stockage.....	51
Figure IV.3 : La creation de la de base de donnée .....	62

Figure IV.4 : La création de la fonction stocker .....	63
Figure IV.5 : La création de la fonction sortie .....	63
Figure IV.6 : Le choix (déstocker ou stocker) .....	64
Figure IV.7 : La position de conteneur à visiter .....	64
Figure IV.8 : La position où stocker le conteneur .....	64
Figure IV.9 : Les emplacements des conteneurs sur la base de données .....	64
Figure IV.10 : Le type de fichier créé .....	65
Figure IV.11 : Fiche de situation journalière .....	65
Figure IV.12 : La simulation des conteneurs .....	65

## LISTE DES TABLEAUX

Tableau I.1 : Dimensions extérieures (mm) .....	09
Tableau I.2 : Dimensions intérieures (mm).....	09
Tableau I.3 : Capacité de conteneur .....	10

# **Introduction générale**

## INTRODUCTION GENERALE

Le transport maritime a existé depuis longtemps et remonte à très loin, c'est le mode de transport de marchandise le plus adapté et le plus utilisé dans les échanges internationaux.

Il n'a cessé d'évoluer depuis le temps, mais sa véritable révolution n'est parvenue qu'au moment de l'apparition du conteneur ou conteneurisation.

Auparavant la marchandise transportée s'effectuait en vrac, dans des sacs, cartons ou caisses en bois, ce qui fait que le chargement/déchargement, tri et distribution de la marchandise fait perdre beaucoup de temps et nécessite une main d'œuvre excessive.

Avec l'avènement du conteneur les échanges internationaux par mode de transport maritime ont pris de l'ampleur.

En effet c'est en 1956 que le transporteur routier dénommée Malcolm Maclean a découvert le conteneur, en ayant l'idée de transporter ses remorques sans leur châssis sur un navire.

Et c'est dix ans plus tard en 1966 qu'un premier navire porte conteneur d'une capacité de 200 à 250 conteneurs a relié l'Amérique et l'Europe. Ses deux dates symbolisent l'apparition de la conteneurisation.

La conteneurisation a boosté la recherche de moyens adaptés au transport de conteneurs. D'où la construction de navires porte-conteneurs, ce qui a entraîné une progression significative dans les échanges de marchandises au niveau mondiale. De ce fait le temps de chargement/déchargement, la durée de trajet et les délais de livraison ce sont largement démunie.

Ainsi les autorités portuaires se sont penchées sur les problèmes d'espaces portuaires car les navires porte-conteneurs sont de plus en plus grands ce qui nécessite des ports plus dimensionnés et à quai plus grand, pour permettre a ces navires de stationner.

Par la suite des espaces été aménagé au niveau des ports au vu de stockage de conteneur et de leur manutention ; c'est ce qu'on appelle le terminal à conteneur.

Suite à l'évolution du trafic international des conteneurs, les terminaux portuaires ce sont trouvés face à des exigences qui les obligent à ; soit l'amélioration de leur capacité c'est-à-dire trouver plus de terrain et d'espace, soit à trouver des solutions technologiques et des systèmes d'information au vu d'une gestion convenable pour améliorer leur rendement. La première solution étant difficilement réalisable car il serait difficile et onéreux de trouver plus de terrain et d'espace. La deuxième solution serait la plus plausible au vu des progrès technologiques réalisés.

Dans cette optique notre contribution consiste à analyser le système de stockage des conteneurs et par la suite apporter des solutions pertinentes pour permettre une gestion rigoureuse au niveau de la zone de stockage.

Notre étude s'est portée sur la zone de stockage de conteneur car c'est le lieu le plus sensible en ce qui concerne la manutention, le remaniement et le mode de stockage qui peuvent influencer sur le rendement du terminal portuaire.

Notre stage s'est dérouler au port d'Oran et notre étude a concerné les différents travaux effectués au niveau du terminal à conteneur du dit port.

# **Chapitre I :**

## **Conteneurisation, ports et terminaux à conteneur**

## I.A. Conteneurisation :

### I.A.1. Historique :

Les échanges internationaux ont été tout le temps en évolution, pour ce faire le transport maritime a été la solution la plus privilégiée.

Ainsi pour pallier aux problèmes d'échanges de marchandises énormes et effectuer leur transport d'une manière efficace, une idée révolutionnaire est apparue, c'est la **conteneurisation**.

La conteneurisation permet le transport de marchandises diverses sur de longues distances a des couts plus bas et des quantités de marchandises plus grandes.

C'est en 1950 que le conteneur maritime a été créé par l'Américain Malcom Maclean transporteur routier. En une dizaine d'année, le système de conteneur s'est répandu dans le monde entier, car il permet l'optimisation de l'espace a bord du navire et un gain de temps aux opérations de manutention et de stockage.

C'est ainsi que dès 1961 apparait la mondialisation ISO du conteneur 20 pieds (6 m) et 40 pieds (12 m) comme dimension standard des conteneurs.

### I.A.2. Les conteneurs :

#### I.A.2.1 Description et propriété du conteneur :

Un conteneur ou container (en anglais) est une caisse métallique parallélépipédique conçu pour le transport de marchandises par différents modes de transport (maritime, routier, ferroviaire et fluvial). Ses dimensions sont normalisées au niveau international.

Déférents types de conteneurs existent et qui répondent aux normes ISO internationales qui régissent le transport des marchandises.

Les deux longueurs largement utilisées sont le 20 pieds (EVP) *l'équivalent vingt pieds* et le 40 pieds (2 EVP) avec une largeur commune de 8 pieds pour satisfaire les besoins et la demande.

#### I.A.2.2. Type des conteneurs :

##### ➤ Le conteneur DRY standard :

Il est le plus utilisé, car il convient à la majorité des marchandises ils peuvent être utilisés pour transporter tout élément sec, et sont fermés de tous les côtés.

Caractéristiques :

- Equipé d'un toit et aux parois latérales.
- Entièrement fermé et hermétique.
- Solidité à toute épreuve.
- Les portes sont situées aux extrémités.
- Etanche à la pluie.
- Plancher stable, fermé et résistant.

- Prise (anneaux) pour le transbordement.



Figure I.1 : Conteneur Dry standard

➤ **Conteneur Open Top :**

Il se caractérise par son toit ouvrant. Il a été conçu pour permettre de charger un objet encombrant qu'on ne peut pas démonter, ou des produits lourds par le haut à l'aide de grue.

Caractéristique du conteneur Open Top :

- Absence de toit en dur.
- Protection du haut par bâche ou couverture amovible.
- Grosses onces métalliques amovible.
- Portes montées sur des gonds.
- Poutres transversales détachables au-dessus de chaque porte.
- Les autres caractéristiques semblables au conteneur DRY.



Figure I.2 : Conteneur Open Top

➤ **Conteneur REEFER :**

Il est conçu pour pouvoir contrôler la température afin de conserver la qualité de la marchandise pendant le transport.

Il est équipé d'un moteur électrique afin de réguler la température intérieure.

C'est un conteneur frigorifique qui sert à maintenir la chaîne du froid pour les produits alimentaires ou périssables pendant les chaleurs.

Caractéristiques du conteneur REEFER :

- Conserver une température ambiante définie.
- Chauffer ou refroidir selon le besoin.
- Offrir une fonction isotherme telle une glacière.



Figure I.3 : Conteneur REEFER

➤ **Conteneur Open Side :**

Il se caractérise par une ouverture latérale sur toute sa longueur. Cela permet de charger des marchandises de longueurs imposantes par chariot élévateur, et ainsi un gain de temps.

Caractéristiques de conteneur Open Side :

- Toujours équipé de portes latérales ou doubles portes.
- Souvent muni de portes aux extrémités.
- Robuste et sécurisé.
- Résistant aux intempéries.
- Plancher en bois.
- Fermeture avec cache cadenas.
- Fermeture avec poignée verrou.



Figure I.4 : Conteneur Open Side

➤ **Conteneur citerne :**

Ce conteneur sert à transporter des produits liquides, gaz, poudre. Les marchandises peuvent être alimentaires ou produits dangereux (essence ou produits gazeux).

La citerne doit être mise dans un cadre pour faciliter le chargement et le déchargement.



Figure I.5 : Conteneur Citerne

➤ **Conteneur flat Rack :**

Il se caractérise par des extrémités rabattables sur le plancher. La base sert à la pose de la charge, les parois latérales servent à fixer la charge au moyen de ridelles ou fosses d'arrimage.



Figure I.6 : Conteneur Flat Rack

### I.A.2.3. Structure de conteneur :

➤ **Structure :**

L'élément porteur du conteneur est sa structure en acier composée de poutres longitudinales et transversales et de montants d'angles. Elle seule garantit la stabilité et la portance du conteneur, le reste, plancher, toiture, parois et portes n'est qu'un habillage selon le type de conteneur voulu.

L'élément clé et commun à tous les conteneurs ISO ce sont les huit coins ISO.

➤ **Les coins ISO du conteneur :**

Ce sont des éléments standardisés. Les dimensions de ces coins et leur écartement en largeur et en hauteur est défini au millimètre pris par la norme ISO 1161, car ils sont utilisés pour la manutention par des engins spécialisés sur tous les terminaux mondiaux.

On distingue :

- Les 4 coins supérieurs nécessaires à la manutention du conteneur par un spreader normalisé au niveau mondial.
- Les 4 coins inférieurs nécessaires au calage du conteneur sur tous les types de transports, navire, train, camion.

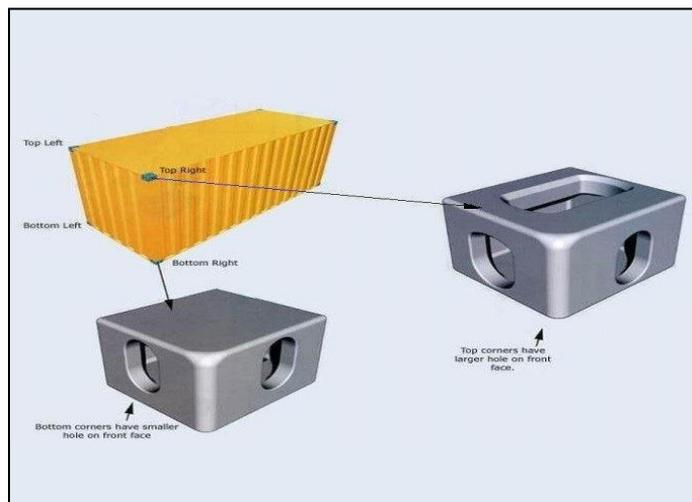


Figure I.7 : Les coins supérieurs et inférieurs de conteneur

➤ **Numéro d'identification :**

Le conteneur est repéré par un numéro d'identification propre à lui durant toute sa durée de vie. Le code a été normalisé par ISO 6346. Le numéro est constitué de :

- Un code de propriétaire de quatre lettres
- Un numéro de série de six chiffres.
- Un chiffre d'auto contrôle destiné à valider l'exactitude de l'enregistrement et la transmission de données. Exemple : **BMAT 453351 3**

**I.A.2.4. Dimensions des conteneurs :**

Les dimensions des conteneurs sont standardisées par la norme ISO à l'échelle internationale. Les conteneurs largement utilisés sont de norme 20 pieds et 40 pieds. L'unité de mesure de la conteneurisation est l'EVP.

Type Conteneur de	20 pieds DRY	20 pieds high Cube	40 pieds DRY	40 pieds high Cube
Longueur	6058	6058	12192	12192
Largeur	2438	2438	2438	2438
Hauteur	2591	2896	2591	2896

Tableau I.1 : Dimensions extérieures (mm).

Type Conteneur de	20 pieds DRY	20 pieds high Cube	40 pieds DRY	40 pieds high cube
Longueur	5898	5898	12032	12032
Largeur	2350	2350	2350	2350
Hauteur	2390	2698	2390	2698

Tableau I.2 : Dimensions intérieures (mm).

Type de Conteneur	20 pieds DRY	20 pieds high Cube	40 pieds DRY	40 pieds high Cube
Poids max (kg)	28550	28120	26000	26000
Volume (m <sup>3</sup> )	33,31	42,77	67	76,29
Tare (P. vide)	2230	2360	3680	3900

Tableau I.3 : Capacité de conteneur

### I.A.3. Avantage de la conteneurisation :

L'utilisation des conteneurs présente plusieurs avantages :

- Sécurité de la marchandise (limitation de vol, d'avarie, dommage et manquant).
- Réduction de temps et de cout de manutention.
- Réduction de temps d'escale de navire.
- Rapidité de livraison.
- L'imitation des ruptures de charge.
- Meilleure organisation portuaire.

Par contre l'inconvénient probable de la conteneurisation c'est que ça nécessite des investissements couteux et beaucoup d'espace. N'empêche que cela ne démunie en rien l'avantage et le rôle essentiel de la conteneurisation dans la pratique et l'évolution des échanges internationaux.

## I.B. Ports :

### I.B.1. Définition :

Le port est un abri naturel ou artificiel pour le navire aménagé pour l'embarquement ou débarquement du fret ou des passagers (suivant le petit Larousse 1996).

Seulement il n'existe pas une définition officielle et approuvée juridiquement par la communauté internationale.

Plusieurs auteurs ont chacun donné une définition à sa façon de voir.

Seule la convention de Genève du 09 Novembre 1923 portant statut du régime international des ports maritimes donne une définition du port comme suit : sont considérés comme des ports maritimes, les ports fréquentés par des navires de mer servant au commerce extérieur.

Mais cette définition aussi n'est que partielle parce qu'elle élimine tous les ports fluviaux ou lacustre (réf. CNUCED-Nations Unis-2012).

Selon l'économiste Bauchet, 1991, il définit le port comme lieu où s'abritent les navires mais aussi celui où passent les marchandises ou même sont transformées.

J.G. Baudelaire définit le port comme un ensemble d'installations conçues et exploitées en vue d'assurer le transfert de marchandises entre le navire de mer et les différents moyens de transports terrestres : Rails, routes, navigations intérieures, canalisations diverses. Mais il ajoute que la considération de la seule nature physique du port n'est pas satisfaisante.

### I.B.2. Types des ports :

Il existe quatre types de ports selon leurs activités.

- **Ports de commerces** : ils servent à accueillir les navires de marchandises et les navires de passagers.
- **Ports de pêche** : ils accueillent les bateaux de pêche. Leurs dimensions sont plus réduites que les ports de commerce.
- **Ports de plaisance** : Ils accueillent les bateaux de plaisance et de loisir. Les bateaux sont de petite taille et ne nécessitent pas de grands espaces.
- **Les ports militaires** : Ce sont des bases navales qui accueillent des navires de guerre. Ils peuvent inclure des arsenaux militaires, des chantiers de réparation et d'autres moyens de ravitaillement militaires. Ils sont interdits au public.

Selon leur localisation on peut classer les ports comme suit :

- **Ports maritimes** : Ils sont situés sur la côte d'une mer ou d'un océan .
- **Ports fluviaux** : Ils sont situés au bord d'un fleuve, d'une rivière ou d'un canal.
- **Ports lacustres** : Ils sont situés au bord d'un lac.
- **Ports secs** : Ils sont situés à l'intérieur des terres et destinés à grouper des marchandises pour les distribuer ultérieurement.

### I.B.3. Rôle des ports et leurs évolutions :

Le port joue un rôle primordial dans les échanges commerciaux entre un pays et le reste du monde. Il contribue grandement au développement économique et social d'un pays. C'est une interface reliant le transport maritime avec le transport terrestre.

Le rôle et les missions du port maritimes ont évoluer en fonction des progrès dans les échanges et le commerce internationaux.

Ainsi les ports constituent aujourd'hui de véritable poumon du commerce extérieur tant pour ce qui est pour l'approvisionnement de biens, que pour les exportations nécessaires au développement de l'économie d'un pays.

Au vu de cette évolution la CNUCED classe les ports en trois générations :

➤ **Port de la 1<sup>ère</sup> génération :**

Jusqu'aux années 1960 les ports jouaient essentiellement le rôle d'interface entre le transport terrestre de marchandises et le transport maritime.

Leur rôle étaient accès sur le chargement et déchargement des navires, le stockage, la réception et la livraison des marchandises.

➤ **Ports de 2<sup>ème</sup> génération :**

A partir des années 1970 le port sera considéré comme un centre de service dans le domaine de transport, de l'industrie et du commerce. La politique portuaire et les stratégies de développement sont élaborées de manière plus ambitieuse.

Le champ des activités du port s'est élargi aux services commerciaux et autres services tel que l'emballage et le marquage des produits. Ainsi des installations industrielles sont implantées à l'intérieur de l'enceinte portuaire.

➤ **Ports de la 3<sup>ème</sup> génération :**

Ils sont apparus en 1980, suite à l'extension au niveau mondial de la conteneurisation et l'inter modalisme, ainsi qu'aux exigences croissantes du commerce international.

La conception de l'exploitation et de l'aménagement des ports est alors différente des précédente. Le port est ainsi considéré comme un maillon dynamique dans le système de production et de distribution international. Ainsi grâce aux efforts fournis les ports ce sont transformés en centre de transport intégré et en plateforme logistique du commerce international.

Dans les années 2000 et à l'ère de la numérisation et la technologie de pointe certains ports sont entrés dans la quatrième génération.

Avec les progrès technologiques et numériques certain ports se sont portés dans la quatrième génération, qui se caractérise par une coopération internationale entre ces communautés portuaires, ainsi que par des réseaux EDI (Environnement de Développement Intégré) entre les places portuaires.

## **I.C.Terminal à conteneur :**

### **I.C.1. Définition :**

Le terminal à conteneur ne bénéficie d'aucune définition légale.

Le terme terminal est importé des états unis. Il se rapporte à la logistique et l'économie portuaire et ne fait l'objet d'aucune définition précise.

Néanmoins l'article 115-7III du code maritime qui évoque la convention d'exploitation du terminal, définit le terminal comme « lieu comprenant les terres pleines, les outillages et les aménagements nécessaires aux opérations de débarquements, d'embarquements, de manutention et de stockage liées au navire ».

### **I.C.2. Rôle du terminal à conteneur :**

Le rôle de terminal à conteneur est de permettre le transbordement entre les divers modes de transports (à savoir entre navire et transport routier). Pour se faire le terminal est doté de matériel de manutention adéquat et onéreux.

Le terminal doit être doté de matériel de levage et de transport inter afin d'exécuter les opérations nécessaires de chargement et déchargement, la manutention et le transfert au niveau de terminal.

Les opérations au niveau de terminal à conteneur peuvent être divisées en trois catégories :

- Chargement et déchargement des navires : s'exécute au niveau de la zone d'opération.
- Opérations de stockage et de manipulation de conteneur : s'exécute au niveau de la zone de stockage.
- Opérations concernant le transfert de conteneur vers les modes de transport terrestre.

### **I.C.3. Organisation du terminal à conteneur :**

Le terminal est organisé en trois zones.

#### ➤ **Zone d'opération portuaire :**

Dans cette zone sont effectués les opérations de chargement/déchargement des conteneurs du navire.

Elle comporte des grues de quai nécessaires à la manutention des conteneurs.

#### ➤ **La zone de stockage de conteneur :**

Cette zone concerne l'entreposage des conteneurs déchargés du navire ou en instance d'être chargés.

Cette zone est subdivisée en trois sous zones : une zone avant visite des conteneurs, une zone de visite et une zone après visite.

Des équipements spécifiques sont utilisés pour la manutention au niveau de cette zone.

➤ **La zone d'opération terrestre :**

Cette zone concerne les opérations qui sont effectuées pour la réception ou l'expédition des conteneurs par voie routière ou ferroviaire. Ici ce fait le contrôle des entrées et sorties des camions.

Représentation schématique des trois zones d'un terminal à conteneur, ci-dessous :

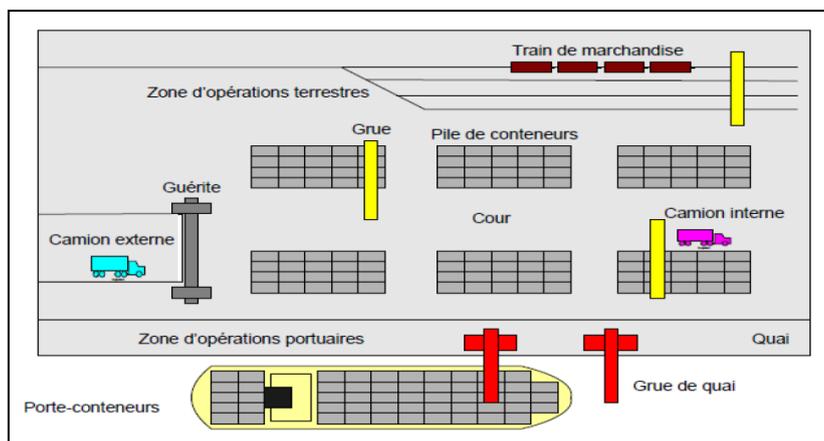


Figure I.8 : Schématisation d'un terminal à conteneur

#### I.C.4. Equipements du terminal à conteneur :

Les équipements dans un terminal à conteneur sont organisés selon les zones d'opérations. Ces Équipements sont adaptés aux besoins spécifiques des infrastructures portuaires.

Les principaux équipements utilisés sont :

- **Grue de quai :**

Utilisée pour le chargement/déchargements du navire. Existe sur pneu ou sur rail.



Figure I.9 : Grue sur rail



Figure I.10 : Grue sur pneu

- **Reach stacker :**

Il est utilisé au niveau de la zone d'opération pour la manutention des conteneurs ou au niveau de la zone de stockage.



Figure I.11 : Rech Stacker

- **Cavalier gerbeur :**

Il est à la fois véhicule de transport et engin de manutention. Son avantage est qu'il soit capable de soulevé, transporté et empiler le conteneur.



Figure I.12 : Cavalier gerbeur

- **Portique à conteneur sur pneu (RTGC) :**

Equipement utilisé pour le stockage de conteneur. Il se déplace sur des pneus, cela lui permet de circuler librement à l'intérieur de la cour de stockage



Figure I.13 : RTGc

- **Portique à conteneur sur rail (RMGC) :**

C'est un engin sur rail, il ne peut circuler que sur des voies ferrées, son avantage principal est qu'il est plus rapide que l'RTGC.



Figure I.14 : RMGc

Il existe d'autre équipement comme chariot élévateur, empileuse à conteneur vide camion mafi.

### **I.C.5. Configuration de la zone de stockage :**

La zone de stockage est la zone la plus importante de terminal car c'est là où s'opère le plus de manœuvre et de remaniement des conteneurs. Les espaces de stockage sont constitués de plusieurs blocs.

On distingue principalement deux types de configuration :

- Le modèle compact
- Le modèle linéaire

Dans le modèle compact il n'existe pas de séparation entre les adjacentes. Le bloc est constitué de plusieurs rangés collés les uns aux autres. Chaque rangé est composé de plusieurs travées qui contiennent à leur tour des piles dans lesquelles sont superposés des conteneurs.

La disposition des blocs diffère selon le genre de grue utilisée

- A savoir grue non automatisées RTGCs : les blocs sont disposés parallèlement au quai.
- Grue automatisées RMGCs : les blocs sont disposés perpendiculairement au quai.

Comme montré dans les deux schémas ci-dessous :

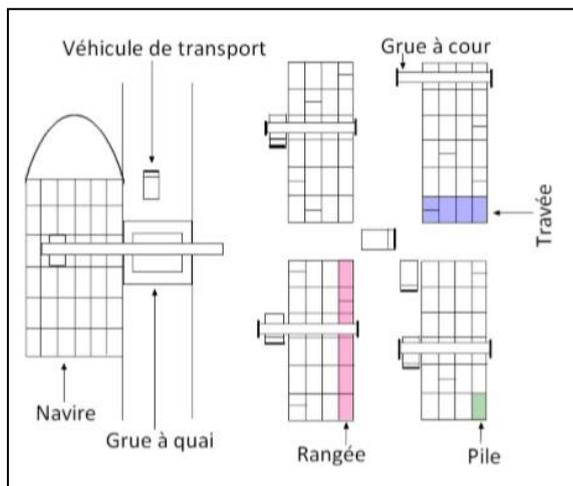


Figure I.15 : Un terminal qui utilise des RTGc

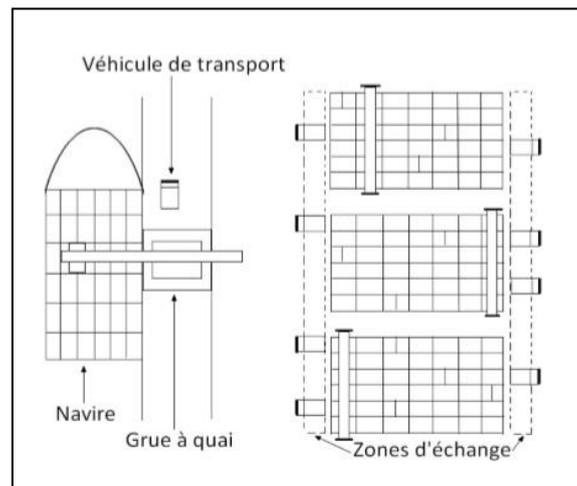


Figure I.16 : Un terminal qui utilise des RMGc

Dans le modèle linéaire les blocs sont constitués par des rangées espacées qui permettent la circulation des roues du cavalier gerbeur.

Chaque rangée est composée de plusieurs piles comme le montre la figure ci-dessous :

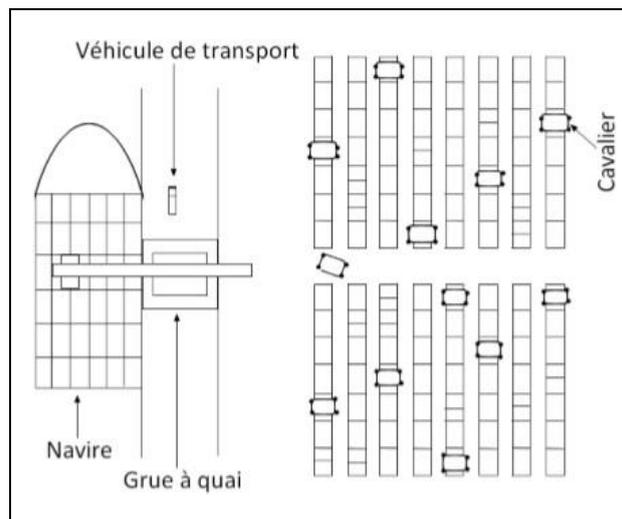


Figure I.17 : Un terminal qui utilise des cavaliers gerbeurs

Chacune de ces différentes configurations peuvent avoir une influence sur le rendement du terminal

Chu et Huang ont réalisé une étude mathématique sur les terminaux au port de Taiwan afin de déterminer la meilleure structure pour chaque type d'équipement. Ils ont conclu que les cavaliers gerbeurs sont plus appropriés aux terminaux qui utilisent un seul quai, tandis que les grus de cour sont plus avantageuse pour les grands terminaux.

# **Chapitre II**

## **Problème de stockage de conteneurs**

## II.1. Introduction :

Le parc de stockage peut être une ressource clé dans un terminal d'instrumentation. Sur Le degré d'efficacité de sa gestion se reflète dans la productivité globale du port. En raison de la réduction des prix du transport de marchandises et de produits induits par la conteneurisation, les taux de conteneurs ont considérablement augmenté dans les ports d'ailleurs.

En 2014, le trafic mondial de conteneurs s'élevait à 680 millions d'EVP (équivalent vingt pieds) représentant 90 % du trafic maritime de marchandises. La tendance est depuis 30 ans à la conteneurisation du transport des marchandises diverses, mais aussi de certaines marchandises traditionnellement transportées en vrac, avec un rythme de croissance de 7 à 8 % par an. Aujourd'hui, la plupart des lignes régulières sont des services conteneurisés.

En effet, la conteneurisation offre trois avantages décisifs : une très forte productivité lors des manutentions portuaires, des économies d'échelle permises par la taille croissante des navires<sup>1</sup>.

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 d'augmenter ces pour augmenter la surface des parcs de stockage, mais cette solution est inefficace car le nombre de conteneurs qui entrent en même temps dans un port ne cesse d'augmenter avec le développement des moyens de transport (en porte-conteneurs).

De plus, le coût du terrain est élevé. Une autre solution à ce problème est d'utiliser l'avantage de la standardisation des tailles des conteneurs, ce qui permet de les empiler les uns sur les autres. Pour être efficace, l'empilement des conteneurs doit également tenir compte des dates d'évacuation.

En effet, une disposition aléatoire des conteneurs dans une pile En effet, une disposition aléatoire des conteneurs dans une pile peut rendre certains conteneurs inaccessibles lors de leur évacuation.

Ainsi, il sera nécessaire d'effectuer des opérations de réarrangement des conteneurs juste avant les heures de départ de certains d'entre eux. Ces opérations sont appelées remaniements, et sont considérées comme des mouvements improductifs qui consomment du temps et ralentissent les opérations portuaires. Pour éviter ce genre de désagréments plusieurs travaux de recherche ont été menés au cours des dernières décennies pour proposer des méthodes de stockage et des systèmes d'aide à la décision efficaces.

Ces derniers font l'objet de ce chapitre, dans lequel des méthodes analytiques et numériques sont présentées.

## II.2. Définition du problème du stockage de conteneurs (PSC) :

À l'arrivée d'un conteneur en import ou en export, il s'agit de décider presque en temps réel de son emplacement exact parmi les emplacements vides de manière à rendre efficace son chargement sur un navire, un camion ou un train. En général, la détermination d'un emplacement doit s'effectuer de manière à minimiser le nombre des mouvements parasites ou improductifs pouvant avoir lieu lorsqu'on veut rapprocher un conteneur éloigné ou extraire un conteneur se trouvant en dessous d'autres au moment de son départ pour être chargé sur le navire, train ou camion associé<sup>2</sup>.

## II.3. Les différentes stratégies de stockage :

Important avantages de l'utilisation de conteneur c'est la possibilité de les faire superposés les uns sur les autres cependant solution a des limites soit dans la mesure où il peut causer des remaniements aussi la consommation importante d'espace en effet ce genre de manœuvre et surtout effectué lors de l'extraction des conteneurs qui sont en haut de la pile alors important pour chaque terminal à conteneur la stratégie de stockage alors on va voir ci-dessous les différentes stratégies Méthodes du stockage qui existe bon la littérature selon Saanen et al<sup>3</sup>.

### II.3.1. Ségrégation et Non-ségrégation :

#### ➤ Stockage par ségrégation :

Les zones de stockage sont préalablement séparées pour déterminer les emplacements qui sont réservés aux conteneurs importés et ceux qui sont destinés aux conteneurs en exportation. Cette répartition peut être faite de trois manières<sup>4</sup> :

- 1) Chaque bloc est réservé, soit aux conteneurs importés, soit aux conteneurs en exportation.
- 2) Diviser les rangées de chaque banque en deux parties, de ce fait, chaque moitié est réservée à l'une des deux catégories de conteneurs.
- 3) La seule différence entre cette méthode et la deuxième est le fait que les divisions se font en considérant les travées.

La ségrégation peut même aller jusqu'à subdiviser l'espace réservé à une catégorie de conteneur, par exemple en préconisant la partie réservée à chaque navire.

Cette stratégie est surtout utilisée pour le stockage des conteneurs qui vont être chargés sur des navires, d'ailleurs elle est utilisée dans <sup>5</sup> par Ibrahim et al.

#### ➤ Stockage par la méthode de non-ségrégation :

Elle ne prend pas en considération les catégories des conteneurs. Par conséquent, les conteneurs peuvent être superposés indépendamment de leurs destinations.

### II.3.2. Groupage et Dispersion :

#### ➤ La méthode de stockage par groupe :

On en fait attribuer à chaque ensemble de conteneurs qui ont les mêmes spécificités des emplacements adjacents (exemple : destination, dimension, contenu, frigorifié, vides, etc).

Les conteneurs qui appartiennent à un même groupe sont supposés être interchangeables, et par conséquent, ils peuvent être superposés d'une manière quelconque sans se soucier de leurs dates de départ <sup>6</sup>.

Cette méthode est surtout utilisée dans les terminaux à conteneurs qui utilisent des RTGs ; car, pour économiser de la main d'œuvre, les conteneurs sont regroupés autant que possible pour minimiser le nombre de grues de cour utilisées <sup>7</sup>.

Deux méthodes de réservation d'emplacements de stockage existent pour cette stratégie<sup>8</sup> :

##### ✓ Méthode unité pile :

Elle commence par réserver une pile vide pour chaque groupe de conteneurs, ensuite elle désigne une nouvelle pile vide à chaque fois que celle d'un groupe devient pleine.

##### ✓ Méthode unité travée :

Elle réserve, dès le début, toute une travée vide à chaque catégorie de conteneur. Et si la travée d'une catégorie devient pleine, elle réserve automatiquement une autre travée vide pour cette catégorie de conteneur.

#### ➤ La méthode du stockage dispersé :

N'essaie pas de regrouper les conteneurs. Ces derniers sont stockés indépendamment les uns des autres.

Une illustration de cette méthode est le stockage aléatoire, qui suppose une équiprobabilité de choix entre les places qui sont compatibles à chaque conteneur <sup>9</sup>.

On peut résumer cette méthode en 5 étapes :

- 1) Choisir aléatoirement une rangée.
- 2) Choisir un emplacement quelconque dans cette rangée.
- 3) Tester s'il est possible d'y placer le conteneur.
- 4) Si oui, alors effectuer le stockage.
- 5) Si non, alors recommencer avec la rangée suivante.

On peut dire de cette méthode est une méthode aléatoire puisqu' elle ne prend pas en considération les informations liées aux conteneurs et aux piles (par exemple : les dates de départ, les distances, etc.),

Elle ne fournit aucune garantie concernant la qualité des solutions. Mais, elle fournit des solutions réalisables.

### II.3.3. Stockage direct et Stockage indirect :

➤ **Stockage direct :**

Cette méthode est appliquée dans la plupart des terminaux portuaires où les conteneurs sont directement placés dans la cour de stockage, où ils vont rester jusqu'à leurs départs.

➤ **Stockage indirect :**

Les conteneurs sont d'abord placés dans une zone d'agencement avant d'être transférés à leur emplacement final dans la cour de stockage. L'objectif de ce procédé est de diminuer les temps d'attente des camions externes qui apportent des conteneurs, et aussi d'accélérer les activités des grues de cour en séparant les périodes de stockage et les périodes de retrait.

Les transferts de conteneurs de la zone d'agencement vers la cour de stockage se font pendant les temps libres des grues de cour. Ce type de procédé est surtout utile pour les ports qui ne disposent pas de toutes les informations nécessaires.

Un des points critiques en utilisant ce genre de procédé est de trouver la bonne division de l'espace de stockage :

- Dans <sup>10</sup>, Chen a étudié un compromis qui consiste à avoir une zone d'agencement très vaste et une cour de stockage qui a des piles très hautes.

- Dans <sup>11</sup> Castillo et Daganzo, qui, En décidant de ne pas séparer la zone d'agencement et la zone de stockage finale, ils considèrent un espace d'agencement à l'intérieur de chaque bloc de stockage.

- Dans <sup>12</sup> Taleb-Ibrahimi, Castillo et. Daganzo on retrouve également cette stratégie de stockage, où les auteurs considèrent que tous les conteneurs qui arrivent dans un bloc sont d'abord placés temporairement dans des piles intermédiaires avant d'être transférés dans les emplacements finales.

Les inconvénients de cette méthode c'est introduire trop de mouvement inutile et elle recueille des moyens matériels importants aussi de la main d'œuvre dans le cas des terminaux à conteneurs non automatisés.

### II.3.4. Priorité aux déchargements et Priorité aux chargements :

➤ **Les méthodes qui priorisent les déchargements :**

Ces méthodes ont pour but de maximiser les performances de toutes les activités liées aux opérations de stockage. Exemple :

✓ **La méthode de stockage par niveau :**

Elle stocke les conteneurs par couche, de telle sorte que tous les emplacements au sol soient occupés, avant de superposer les conteneurs. Cette stratégie fut proposée par Duinkerken et al dans <sup>13</sup>.

Elle est intuitive mais n'utilise pas la plupart des informations disponibles. Elle contient principalement quatre étapes :

- 1) Prendre une rangée quelconque qui a au moins un emplacement libre.
- 2) Chercher, dans cette rangée, un emplacement libre et adéquat qui est au contact du sol.
- 3) S'il est trouvé : y stocker le conteneur.
- 4) S'il n'est pas trouvé : chercher, dans la rangée, un emplacement libre et adéquat qui appartient au niveau le plus bas possible.

Avec le stockage par niveau, le risque de remaniement est moins important qu'avec la méthode aléatoire ; d'ailleurs il peut être inexistant lorsque toute la surface du terminal n'est pas occupée.

Cependant, son efficacité est remise en question lorsque le nombre de conteneurs à stocker est largement supérieur à celui des piles.

Un autre point faible de cette méthode, est le fait qu'elle puisse nécessiter de parcourir de longues distances dans la cour de stockage, ce qui peut augmenter le temps nécessaire pour effectuer les opérations de stockage ou de déstockage.

➤ **Les méthodes qui priorisent les chargements de conteneurs :**

Ont pour but de maximiser le rendement des opérations de retrait. C'est le cas de la méthode qui stocke les conteneurs suivant l'ordre décroissant de leurs dates de départ <sup>14</sup>.

Une version améliorée de cette méthode, appelée méthode de stockage par nivellement des dates de départ : proposée dans [9]15 par Borgman et al.

Leur idée est de stocker les conteneurs suivant l'ordre décroissant de leurs dates de départ, tout en essayant de minimiser la surface utilisée.

Ainsi, si plusieurs piles sont compatibles à un conteneur, alors on le stocke dans celle qui est la plus haute.

En plus de cela, le stockage se fait aussi en minimisant les écarts entre les dates de départ de deux conteneurs qui se succèdent dans une même pile. Le nivellement ne se fait donc pas par rapport au sol mais plutôt par rapport aux dates de départ.

Avec cette méthode, la recherche d'un emplacement de stockage pour un conteneur se fait en trois étapes qui se succèdent comme suit :

- 1) D'abord on cherche parmi les piles qui ne sont ni pleines, ni vides, celles qui ont, à leurs sommets, des conteneurs qui ont des dates de départ supérieures à celle du conteneur que l'on veut stocker. Si on en trouve, on calcule, pour chacune d'elles, la différence entre la date de départ du conteneur qui est à son sommet et celle du conteneur que l'on cherche à stocker. Ensuite on sélectionne la pile qui conduit à la plus petite différence.
- 2) Si de telles piles n'existent pas, alors on choisit parmi les piles vides, celle qui est plus proche de la sortie par laquelle le conteneur sera livré.
- 3) Si on n'a pas trouvé de pile qui appartient aux deux premiers cas, alors on stocke le conteneur dans la pile la plus haute parmi celles qui ne sont pas pleines ; an de minimiser les futurs remaniements.

Selon Borgman et al. <sup>16</sup>, cette méthode est plus efficace que le stockage aléatoire et le stockage par niveau, car le risque d'avoir des remaniements est nettement moins élevé.

Néanmoins, elle privilégie la minimisation de la surface de stockage utilisée au détriment de la rapidité du stockage ou du déstockage qui peuvent avoir des impacts sur la satisfaction des clients.

Des combinaisons entre différentes stratégies de stockage sont proposées dans la littérature, elles seront décrites dans la section suivante.

## II.4. Méthodes de résolution :

La résolution du problème de stockage de conteneurs se passe en deux étapes :

### - La première phase :

Des analyses sont effectuées pour rechercher la méthode de stockage adéquat. Certains papiers disponibles dans la littérature se sont uniquement focalisés sur l'étude analytique du problème, en proposant des méthodes de stockage pertinentes.

### - La deuxième phase :

La partie simulation permet de vérifier et de mesurer l'efficacité et les limites techniques de stockage qu'ils ont proposées dans la partie précédente.

Dans cette partie, nous allons parler des publications qui se limitent à l'analyse du problème. Après, nous parlons des différentes techniques de simulation et des algorithmes qui sont proposés dans la littérature pour la résolution du problème de stockage de conteneurs.

### II.4.1. Études s'appuyant sur la simulation pour résoudre le problème de stockage de conteneurs :

Grâce aux progrès technologiques constants et, l'ordinateur est plus complexe. L'une des principales applications de cette machine est sa capacité à terminer rapidement des travaux qui peuvent nécessiter beaucoup de travail à de temps en temps. La multiplicité des langages de programmation en fait également un outil apprécié pour les calculs, voire les simulations numériques. Ces moyens sont de plus en plus utilisés pour résoudre le problème du stockage des conteneurs dans un terminal portuaire.

Kap Hwan Kim <sup>20</sup> est parmi les premiers à avoir construit un programme informatique utilisant le langage C pour déterminer le nombre total d'ajustements d'extraction de conteneurs prévus pour un bloc.

Dans cette étude, l'auteur s'est uniquement intéressé au stockage des conteneurs fournis par les navires (c'est-à-dire les conteneurs importés par és) et a analysé le terminal à conteneurs utilisant des grues de cour. Il s'est néanmoins basé sur un mécanisme de ségrégation, qui ne permettait pas de mélanger des conteneurs provenant de navires de même longueur.

En outre, il prévoyait que les conteneurs déplacés pendant la rénovation seraient conservés à l'endroit où ils étaient placés à l'origine, de sorte que la grue cour n'aurait pas à se déplacer.

Les principaux contributeurs sont des tableaux et des formules qui déterminent la quantité de reprise causée lorsque les conteneurs sont extraits par ordre aléatoire par les grues de cour.

Toutefois, le degré d'exactitude de ces techniques de calcul doit être établi, car l'auteur pensait que tous les conteneurs avaient toujours la même possibilité de réclamation, ce qui ne correspond pas à la réalité.

Un autre avantage de la simulation est qu'elle rend les enquêtes quantitatives plus simples à réaliser. Sculli et Hui <sup>21</sup> l'ont utilisée pour explorer une situation spécifique du problème de stockage des conteneurs, où tous les conteneurs avaient la même taille.

Dans leur enquête, ils ont observé des oscillations dans quatre critères de performance : la vitesse de remplissage du parc de stockage, le nombre de mouvements improductifs, le nombre de conteneurs non stockés et aussi le nombre de places sans emploi.

Les auteurs ont estimé que chaque pile ne peut contenir plus de 3 conteneurs, ce qui signifie que le volume d'un bloc est égal au nombre d'emplacements au sol multiplié par 3. La quantité de mouvements improductifs est équivalente au nombre de mouvements de conteneurs non réclamés qui se produisent lorsqu'un conteneur est extrait sous d'autres conteneurs.

Les variations entre les instances utilisées dans leurs tests se situent aux niveaux suivants : l'espace de stockage maximal en question, la règle de stockage utilisée et le nombre de types de conteneurs. La recherche évalue deux stratégies de stockage : une approche aléatoire et une technique de catégorie de stockage.

La première consiste à placer chaque conteneur en première position pour commencer d'abord par le conteneur de niveau 1 (sur le sol), puis le conteneur de niveau 2 et le deuxième niveau, à moins qu'il n'y ait d'autres espaces dans les deux précédents.

La deuxième technique s'occupe des types de conteneurs, permettant de regrouper les conteneurs de même catégorie dans chaque pile. L'inconvénient du système est que, même si la cour de stockage n'est pas pleine, il peut arriver que trouver une place adéquate pour un conteneur particulier soit difficile ; dans ce cas, les auteurs proposent de stocker le conteneur de manière aléatoire.

Après plusieurs simulations sur 30 deux instances, Sculli et Hui ont observé que ces deux tactiques ne mobilisent pas le pourcentage de remplissage de la cour de stockage, tout en maintenant le nombre de mouvements improductifs. Selon eux, l'approche aléatoire conduit à des piles très élevées, de plus, elle crée aussi beaucoup de mouvements inutiles contrairement à l'autre voie.

L'absence de remaniements dans la deuxième stratégie de stock implique la réclamation simultanée de conteneurs du même type. Le nombre de types de conteneurs et la capacité maximale du parc de stockage ont également un impact sur le nombre de changements.

Une deuxième révélation de leur expérience est que le nombre de conteneurs non stockés et les espaces restants dépendent uniquement de leur capacité maximale et de l'espace de stockage global.

Il est vrai que Scully et Qui ont pris en considération des paramètres pertinents dans ce document pour résoudre le problème des conteneurs, mais ils ont également simplifié le problème en considérant que tous les conteneurs sont de taille identique et que tous les conteneurs du même type sortent en même temps, ce qui n'est pas toujours le cas dans les terminaux à conteneurs.

À <sup>22</sup>. Ma et Kim ont utilisé "eM plant" pour optimiser les performances de diverses techniques de stockage dans un logiciel d'optimisation. Ils ont considéré un terminal à conteneurs utilisant le GCMS et des grues automatisées et camions. Les blocs sont définis en liaison avec les quais.

Dans cette étude, la performance évaluée est le temps que les camions passent sur les blocs. La durée du trajet varie entre le quai et le parc de stockage des camions et le nombre de conteneurs sans espace de stockage. Les simulations décrites ici influencent à la fois les conteneurs importés et les conteneurs exportés. Les conteneurs ne sont toutefois pas mélangés à la cour.

Trois approches de ségrégation ont été examinées dans cette recherche :

- La première, "REBLOCK", associer un type de conteneur avec chaque bloc.
- La deuxième "ASROW" sépare chaque les rangés de chaque bloc en deux parties, l'une pour les conteneurs d'importation et l'autre pour les conteneurs d'exportation.
- La troisième option, appelée "RSBAY", divise la longueur de chaque bloc pour un type de conteneur en deux groupes. En outre, les auteurs ont choisi de stocker les conteneurs qui sont expédiés en groupes selon leur taille et les navires de destination.

Cependant, deux règles sont essayées pour attribuer l'espace de stockage pour ces groupes : l'une consiste à attribuer une pile par demande " unité pile " et l'autre qui réserve une travée par demande (unité travée).

Après de nombreuses simulations, Ma et Kim ont constaté que la procédure de stockage "REBLOCK" permettait aux camions de passer plus de temps dans les blocs.

Cette stratégie, ainsi que "RSBAY" et "RSROW", entraînent néanmoins des pénuries de places plus importantes que la technique de stockage aléatoire, qui ne limite pas le type de conteneur. L'approche " unité traves " a permis de créer plus d'espace que l'approche " pile unit ", mais l'avantage de la " pile unit " est de réduire le temps moyen passé en bloc par les camions.

Les auteurs ont également montré que le fait de privilégier les blocs à proximité des quais est moins bénéfique que le temps moyen passé sur une base aléatoire pour le prélèvement des blocs.

Dans <sup>23</sup>, Jiang et al recommandent un système C++ qui comprend à la fois un mécanisme de distribution des parcs de stockage et un mécanisme d'affectation des charges de travail aux grues de parc en deux phases successives, cette technique est spécialement destinée à traiter le problème des conteneurs stockés entre les navires à revenus.

Dans la première étape, cette procédure est appelée "Template génération" Zones de stockage pour chaque navire.

Dans la deuxième étape appelée "Space allocation and workload assignment", il détermine ensuite les espaces de stockage entre deux navires différents et la quantité de conteneurs attribuée à chaque grue de cour.

Pour exécuter ces deux phases, trois modèles mathématiques ont été présentés. Un pour la "Template génération" et deux pour la "Space allocation and workload assignment". Affectation des tâches ; L'objectif des auteurs est de fournir des espaces de stockage minimaux et adéquats pour les navires dans le modèle mathématique initial.

En ce qui concerne les deux modèles de la résolution sur "Space allocation and workload assignment", chacun illustre un certain nombre de stratégies de ségrégation.

Ces deux versions reposent sur le même principe fondamental, qui consiste à diviser deux zones successives allouées à deux navires distincts par l'espace de stockage partagé.

Cependant, la différence entre ces deux nouvelles stratégies est que les zones communes ont des mesures identiques dans le premier "Fixed shared space", tandis que les espaces communs ont des mesures différentes dans le second "variable shared space", qui sont déterminées par la charge d'exploitation des grues courtes, pour indiquer l'effet que les croisières sont impliquées dans cet article.

L'objectif du modèle proposé pour l'approche "fixed shared space" est de maximiser la taille globale des espaces communs, tandis que celui requis dans le modèle "Variable shared space" est de minimiser la quantité totale de l'espace stocké dans chaque bloc.

Jiang et al ont effectué des simulations numériques pour valider le système qu'ils ont présenté. Les deux ont montré que les deux nouvelles techniques sont supérieures à l'ancienne approche de séparation, qui préserve un espace dans lequel chaque vaisseau a des dimensions stables et inchangées, mais pas les mêmes résultats que la méthode de "variable shared space", avec les meilleurs résultats.

Cependant, Jiang et al ont souligné qu'ils n'ont pas pris en compte les incertitudes qui peuvent affecter les données qu'ils ont utilisées dans les modèles mathématiques et ont vivement recommandé de poursuivre leur étude à ce niveau.

Les techniques d'optimisation spécifiques ne figurent pas encore dans l'article. Les publications des dernières décennies ont toutefois montré l'application de la recherche opérationnelle pour résoudre le problème du stockage des conteneurs. Certaines des approches d'optimisation de la littérature offrent des résultats idéaux.

Cependant, la limite fondamentale de ces approches exactes est qu'elles nécessitent généralement beaucoup de temps informatique. Cela remédie aux algorithmes heuristiques et méta-heuristiques. Ces derniers nécessitent généralement un temps de calcul acceptable, bien que la réponse parfaite ne soit pas garantie.

Dans la littérature, les algorithmes heuristiques sont généralement propres à des problèmes particuliers. Les algorithmes méta-heuristiques les plus connus sont plus flexibles "L'approche 'tabou' a imité le recrutement des fourmis : algorithmes génétiques et algorithmes de colonies.

Dans les paragraphes suivants, nous traitons de diverses méthodologies d'optimisation, tout en notant les références utilisées pour aborder le problème du stockage des conteneurs.

### II.4.2. Programmation dynamique :

En 1957, Richard Bellman a développé la programmation dynamique <sup>24</sup>. Culioli <sup>25</sup> (page 295) affirme que cette approche d'optimisation dynamique est particulièrement adaptée aux problèmes d'optimisation séquentielle, pour lesquels on souhaite minimiser un coût séparable en temps, le long d'un chemin.

Cette méthode fait naturellement intervenir une variable temporelle, également appelée variable d'étape,  $t$  (qui peut être un nombre ou un intervalle compris dans  $[0, T]$ ), une variable d'état appelée  $x$  (avec  $x(t) \in R$ ), et une variable de décision appelée  $u$  (avec  $u \in R$ ). Une équation d'état, dont le format est  $x(t+1) = f(x(t); u(t), t)$  décrit l'évolution du système dynamique.

Par conséquent, la connaissance de l'état du système  $x(t)$  à un instant et une série de décisions  $u(t+k); k=0 \text{ et } t+k \leq T-1$ , conduit à la connaissance de tous les états successifs.

Ceci est en parfait accord avec le principe d'optimalité de Bellman, selon lequel dans un processus d'optimisation dynamique, une séquence de décisions sera optimale si, à tout état et moment considéré sur la trajectoire associée, les décisions ultérieures constituent une séquence optimale de choix pour le sous-problème dynamique qui se trouve dans cet état et ce moment condensé <sup>26</sup>.

Dans le cas continu, l'équation d'état devient  $\dot{X} = f(x(t); u(t); t)$ . Plusieurs articles ont utilisé la programmation dynamique pour résoudre le problème du stockage des conteneurs. Kim et Bae ont utilisé cette stratégie dans <sup>27</sup> pour traiter le problème du déplacement des conteneurs dans un terminal portuaire.

Les conteneurs qui seront chargés sur les mêmes navires ont été particulièrement intéressés. Ils affirment que le déplacement des conteneurs permet d'accélérer le chargement des navires. Ils partent du principe que la stratégie de chargement d'un navire n'est généralement connue qu'une fois que tous les conteneurs prévus ont été atteints et stockés.

Par conséquent, l'ordre de chargement de ces conteneurs sur les bateaux peut ne pas correspondre à l'ordre spécifié sur l'installation de stockage.

C'est pourquoi Kim et Bae ont mis au point une méthode qui permet de modifier une disposition particulière des conteneurs sous la forme d'un schéma nécessaire pour déplacer le moins de conteneurs possible et réduire la longueur des trajets.

À cette fin, ils ont séparé le problème en trois sous-problèmes : le problème de la correspondance entre les traves, l'ordre de déplacement et l'ordonnancement des tâches. Le défi de la correspondance entre les travées consiste à combiner chaque travée du motif original avec une travée différente dans la travée souhaitée. La quantité de conteneurs à transférer entre deux travées quelconques est établie pendant la phase de mise à plat des mouvements.

Le troisième sous-problème consiste à minimiser la durée totale du transfert. Les auteurs ont commencé par répondre en même temps au problème de la liaison entre l'intervalle et la question de l'appareil à déplacer. Ils emploient la programmation dynamique à chaque itération pour résoudre le premier sous-problème, puis utilisent les résultats acquis pour résoudre le second sous-problème en utilisant une approche de solution de transport.

Mais comme ils ont estimé qu'il était impossible de faire fonctionner deux grues de cour simultanément tant qu'une distance de sécurité égale à la largeur de deux longueurs ne les sépare pas, certaines correspondances entre les longueurs peuvent créer des problèmes et sont donc incluses dans une liste de restrictions.

La question des correspondances est abordée à nouveau en tenant compte des correspondances interdites. Les deux méthodes sont poursuivies jusqu'à ce que le mouvement correct soit acquis.

Le troisième sous-problème (c'est-à-dire le problème d'ordonnancement des tâches) est ensuite résolu par programmation dynamique sous la forme d'un problème de voyageur d'affaires, y compris les pré-relations, sous la forme de termes de satisfaction. Les auteurs ont remarqué que les techniques de programmation mathématique, en particulier pour l'ordonnancement des tâches, nécessitent beaucoup de temps de traitement et ont conseillé l'utilisation d'une méthode heuristique.

Kim et al dans <sup>28</sup> ont développé une méthodologie de programmation dynamique pour assigner des emplacements de stockage aux conteneurs d'exportation. Ils ont proposé un terminal de conteneurs équipé pour traiter les véhicules et les coursiers.

Cependant, dans la méthodologie qu'ils ont présentée pour résoudre le problème, les auteurs ont pris en compte les écarts de poids entre les conteneurs. Selon eux, les conteneurs les plus lourds doivent être placés près du fond pour garantir l'équilibre d'un navire.

Par conséquent, ils ont suggéré de disposer les conteneurs dans le parc de stockage de telle sorte que les conteneurs les plus lourds puissent être à la hauteur des batteries afin d'accélérer le chargement des navires.

Leur principal objectif est de minimiser le nombre de réorganisations du chargement des navires, mais ils pensaient auparavant que les conteneurs reliés à une traversée étaient du même groupe (même destination et même type). Ils prévoyaient néanmoins que les camions externes transportant des conteneurs seraient vidés selon le critère du premier arrivé, premier servi.

En outre, chaque conteneur ne peut être déplacé qu'une seule fois vers un autre lieu de stockage. Après avoir évalué l'approche de programmation dynamique qu'ils ont présentée pour traiter ce problème de stockage de conteneurs, Kim et al ont observé qu'elle prenait un temps de calcul très long et qu'elle ne permettait pas de prendre des décisions en temps réel.

Par conséquent, les auteurs ont créé un arbre de décision dont les nœuds sont des solutions obtenues à l'aide de leur modèle de planification dynamique, afin de réduire les temps de résolution.

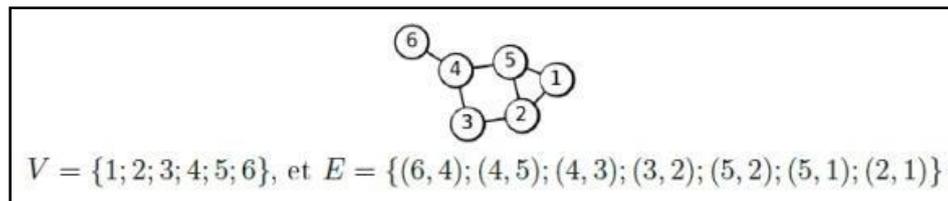
Par contre, l'arbre de recherche ne garantit pas la meilleure réponse puisque les auteurs ont utilisé des processus de classification dans le processus de classification de l'arbre de recherche qui comprennent des critères pour choisir les caractéristiques de connexion importantes, une règle d'impression et une technique de connexion simple.

En outre, les comparaisons entre les auteurs et leur programme dynamique démontrent que, selon l'ampleur de l'erreur autorisée lors des procédures de découpage, les solutions non optimales proposées par l'arbre de recherche varient de 1,0. à 5,5 pour cent.

### II.4.3. Utilisation de la théorie des graphes :

La théorie des graphes est appliquée de nombreuses façons pour résoudre de nombreux problèmes réels, notamment celui du stockage des conteneurs sur un terminal portuaire. Ce qui suit permet d'appréhender les différentes techniques de résolution de ce problème proposées dans la littérature et basées sur une notion graphique.

Le graphe est appelé une paire ordonnée  $G = (V;E)$ , comprenant une série de sommets notés  $V$  et un ensemble d'arêtes (dans le cas d'un graphe non orienté, ou d'arcs dans le cas d'un graphe orienté) appelé  $E$ . Chaque élément de  $E$  est une paire de  $V$  éléments, comme on peut le voir sur l'illustration.



Dans le cas d'un graphe orienté, la capacité de chaque arc peut être renseignée ; celle-ci est une fonction définie comme suit :

Le graphe orienté devient un réseau que l'on note  $G = (V;E;c)$ . En considérant un tel graphique, on peut définir la notion de flot. Flot dans un graphe : soit  $G = (V;E; c)$  un graphe ayant un seul sommet source  $s$ , et un seul sommet puits  $p$ . Un et déni de  $s$  à  $p$  est une fonction  $f : E \rightarrow \mathbb{R}$ , qui varie les deux conditions suivantes :

$$\sum_{(i,j) \in E} f(i,j) = \sum_{(j,k) \in E} f(j,k), \forall j \in V / \{s, p\}$$

pour assurer le respect des capacités

$$f(i,j) < c(i,j); \quad \forall (i,j) \in E$$

pour assurer la conservation de flux

Le cas où les arcs sont pondérés, on utilise une fonction de coût supplémentaire noté  $U$  :

$$u : E \rightarrow \mathbb{R} :$$

Problème du ot à coût maximal (ou minimal) : Il s'agit de déterminer les flux qui maximisent (ou minimisent) le coût global, tout en respectant les contraintes de capacité et de conservation des flux. Dans le cas de la maximisation, le problème est formulé comme suit.

$$\max \sum_{(i,j) \in E} c(i,j) \cdot f(i,j)$$

sous les contraintes de capacité et de flux

Dans <sup>29</sup>, Lee et Hsu ont converti la question de la relocalisation des conteneurs dans une carte orientée en une question de flot multiple dans un graphe orienté. Selon eux, la relocalisation des conteneurs du navire dans la séquence appropriée permet d'accélérer la cargaison du navire en minimisant le nombre de restructures prévues dans le même temps (voir même rendu nul).

Ils ont prévu que la manutention serait effectuée par des GMCC et que chaque conteneur déplacé serait réentreposé à son emplacement d'origine. Chaque nœud représente une installation de stockage dans le tableau présenté, les arcs correspondent aux possibilités de déplacement des conteneurs, et les flux de déplacement des conteneurs dans le graphique reflètent le mouvement spatio-temporel.

A partir de ce graphe, les auteurs ont proposé un modèle mathématique linéaire dans lequel tous les conteneurs considérés sont destinés à un seul comme un seul navire et sont placés dans une même travée.

En plus de cela, ils pensaient que seuls les conteneurs de même taille (dans chaque batterie de la travée) pouvaient être superposés, et que l'ordre dans le conteneur pouvait être entièrement connu avant le début du chargement (mais on ne savait pas encore quand ces conteneurs arrivaient au port).

Lee et Hsu ont converti ce modèle en C++ pour le résoudre à l'aide de CPLEX 9.0. En utilisant ce modèle, ils ont toutefois découvert que le temps d'exécution était plutôt long. Ils ont donc présenté une stratégie heuristique qui se déroule essentiellement en deux phases pour minimiser les temps de résolution :

- Dans la première partie, l'algorithme tente de localiser un ensemble de mouvements qui conduisent à une conjuration efficace de l'étendue.

- L'heuristique trie ces mouvements et ajoutant plus de mouvements élimine les cycles. Lee et Hsu ont déclaré que cette technique heuristique est très rapide et ont proposé une étude future dans le cas de nombreux navires.

Dans <sup>30</sup>, Yu et Qi ont utilisé la "Capacity scaling algorithm", suivant deux techniques de stockage alternatives, pour traiter le stockage de conteneurs importés (qui ont été livrés au port par des navires).

La première non ségrégative (c'est-à-dire qu'elle n'interdit pas le stockage de conteneurs provenant de navires méritants dans une même quai), contrairement à la deuxième qui est ségrégative et à courte vue car seuls les conteneurs arrivés pendant la période de stockage en cours sont pris en compte (cette méthode est appelée ségrégation à période unique).

Yu et Qi ont également proposé une troisième stratégie de stockage qu'ils ont appelé la ségrégation multi-période, dont la seule différence avec la deuxième stratégie est qu'elle prend en compte les conteneurs qui arriveront dans les futures périodes de planification.

Pour chacune de ces trois stratégies ils ont résolu le problème de stockage en deux étapes. Le nombre de conteneurs est décidé dans la première phase (sans spécifier l'emplacement réel de chaque conteneur).

Le problème du déplacement est également abordé dans la deuxième phase, en tenant compte d'une limite de manutention. Ces deux étapes visent à réduire les délais d'extraction des conteneurs à la demande de leurs propriétaires.

Cependant, étant donné que les auteurs supposent que les demandes de conteneurs sont totalement inconnues, pendant toute la même période, la durée de l'extraction du conteneur était basée sur la quantité de conteneurs dans la même travée.

Cela démontre que les nombres de remaniements prévus pour les extractions n'ont pas été déterminés. Ils ne définissent pas les conteneurs qui sont transférés d'une travée à l'autre dans la phase de relocalisation, mais modifient la partition des conteneurs en travées (c'est-à-dire qu'ils recalculent le nombre de conteneurs que doit contenir chaque travée).

Yu et Qi ont proposé trois modèles mathématiques, un pour chaque approche, pour compléter la phase de stockage initial. Ils ont souligné les fonctions objectives des modèles pour les deux stratégies initiales qui consistent en des coûts convexes indépendants et n'ont donc transformé ces problèmes en problèmes de flux minimum dans des graphes orientés avec des travées sur les nœuds.

Par la suite, ils ont résolu chacun de ces deux problèmes flot en utilisant l'approche de Capacity scaling algorithm. En ce qui concerne la troisième stratégie, les auteurs ont proposé une approche de programmation dynamique pour sa résolution.

Un seul modèle mathématique pour la question de la relocalisation a été proposé, indépendamment des trois stratégies différentes rapportées dans les lignes précédentes. Cependant, contrairement au problème d'allocations (dans la première phase), Yu et Qi n'ont pas résolu le problème de la relocalisation avec une approche exacte.

Ce dernier est divisé en plusieurs itérations dans lesquelles différentes opérations incluent le transport de conteneurs entre différentes travées afin de réduire le temps prévu pour l'extraction. Lorsque le temps de déplacement alloué est atteint, l'algorithme s'arrête. bien lorsqu'il n'est plus possible de diminuer les durées d'extraction prévues.

Les auteurs ont considéré que les manutentions sont effectuées par des grues automatisées (RMGCs), et que les opérations de relocalisation des conteneurs sont exécutées pendant la nuit. Après avoir exécuté des simulations numériques, en considérant un horizon de planification long de trois mois (qui est divisé en des périodes qui sont chacune équivalente à un jour), les auteurs ont remarqué que le multiple-période ségrégation procure une durée moyenne de retrait inférieure à celle qui est obtenue en utilisant le single-période ségrégation. On ne peut plus abaisser les durées d'extraction prévues.

Les auteurs ont considéré que des grues automatisées (RMGCs) seraient utilisées pour la manutention et que les déplacements de conteneurs auraient lieu la nuit. Suite à des simulations numériques, en considérant un horizon de planification long de trois mois les auteurs observent que le multiple-période ségrégation offre une période de retrait moyenne inférieure à le single-période ségrégation.

En outre, ils ont également observé que la méthode de non-ségrégation est plus adaptée au problème, car elle permet d'obtenir le temps d'extraction le plus faible et de réduire le risque de manque d'espace. D'un autre point de vue, Yu et Qi ont suggéré de considérer le cas d'un stockage où tous les conteneurs n'ont pas la même chance d'être retirés à tout moment.

#### **II.4.4. Algorithmes génétiques :**

Les algorithmes génétiques sont créés en 1975 par le chercheur américain Jonh Henry Holland, qui a publié, dans <sup>31</sup>, le premier modèle formel appelé the -canonical génétique algorithml. Ils sont basés sur des algorithmes évolutifs, inspirés des processus naturels de reproduction animale.

En fait, il est utile d'optimiser ce phénomène, car il peut empêcher la convergence rapide vers un optimum local. La mise en œuvre d'un algorithme génétique nécessite la détermination préalable de six paramètres qui sont détaillés ci-dessous.

##### **✓ Un principe de codage :**

Une façon de représenter une seule personne dans la population étudiée, en d'autres termes. Le codage binaire (qui comprend une série de 0 et de 1), le codage à caractères multiples et le codage arborescent sont les types les plus populaires dans la littérature (qui utilise une structure arborescente, d'où peuvent provenir un ou plusieurs 1). Il est essentiel, car cela a un impact sur l'efficacité de la méthode, que le problème étudié soit bien codé.

##### **✓ Une manière de générer la population initiale :**

Cela devrait permettre la formation de personnes non identiques. L'excentricité et l'imprécision de cette méthodologie se reflètent dans les résultats finaux de l'algorithme, car une population très pauvre pourrait conduire à une conversion prématurée vers un emplacement optimal, loin du l'optimum global. Il est donc très important d'avoir un début de population qui est largement répandu dans toute la région de définition du problème recherché.

##### **✓ Une fonction d'évaluation :**

Cela permet de calculer la performance de chaque personne dans la population (appelée fitness). Elle permet également de connaître la meilleure individuue (solution actuelle) dans une population, mais elle est aussi souvent utilisée pour choisir les individus à partir desquelles la génération suivante est générée.

✓ **Un opérateur de croisement :**

La création de la population d'origine, et le transfert de caractéristiques par les anciennes générations aux nouvelles générations sont également possibles. Le croisement en un point, le croisement en deux points, la méthode du couper-et-joindre, le croisement uniforme avec trois parents sont les méthodes de croisement les plus typiques découvertes dans la littérature.

- **Croisement en un point :**

Sélectionner un point quelconque, puis les parties droites des deux parents s'inversent. Ceci est montré dans la figure.

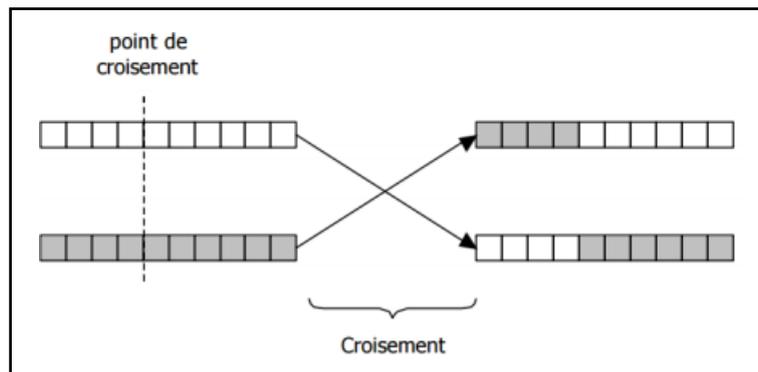


Figure II.1 : Croisement en un point

- **Croisement en deux points :**

Il est identique à un croisement à un point, sauf qu'au lieu d'une portion, deux emplacements sont choisis. Comme l'illustre la figure, les pièces entre les deux portions seront permutées.

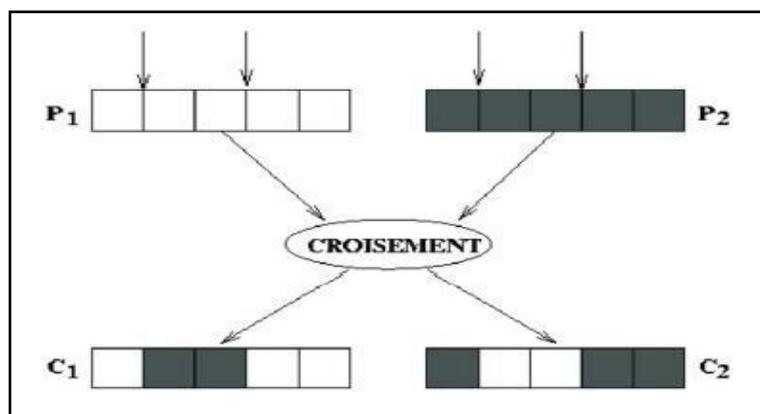


Figure II.2 : Croisement en deux points

Une forme de cette procédure est un croisement partiel proposé par Goldberg et Lingle<sup>32</sup>. Il fournit un mécanisme de réparation des chromosomes pour la résolution des violations des restrictions spéciales, ce qui est original pour cette technique. Ça progresse comme ça.

- 1) Sélectionner aléatoirement deux points de section.
- 2) Modifier les pièces qui s'y trouvent, pour créer deux nouveaux enfants.
- 3) Identifier un processus correctif.
- 4) Utiliser cette technique pour légaliser les enfants.

- **Croisement uniforme :**

Cette méthode est développée par Syswerda, c'est un codage spécifique aux chromosomes sous forme de chaînes de bits<sup>34</sup>.

Au départ, il génère aléatoirement un modèle sur lequel il s'appuie pour échanger les gènes parents. Ainsi, elle produit deux enfants qui ont les mêmes tailles que leurs parents. Une amélioration du croisement standard appelée croisement basé sur la position est proposée par Syswerda<sup>35</sup>.

Celui-ci ajoute une correction au croisement uniforme, et son déroulement est presque similaire à celui d'un ordre de croisement ; la seule division se situe au niveau du premier pas. L'algorithme de croisement basé sur la position est le suivant :

- 1) Sélectionnez un certain nombre d'emplacements dans un parent de façon aléatoire.
- 2) Créez et copiez l'enfant vide à l'endroit approprié.
- 3) Chaque deuxième symbole du parent fait partie du composant choisi du premier parent. Les autres sont les éléments nécessaires aux enfants.
- 4) Placez les autres composants du parent dans les cases vides, de gauche à droite, de l'enfant.

✓ **L'opérateur de mutation :**

En perturbant un ou plusieurs de ses gènes, exécute un certain désordre dans un chromosome. Elle maintient un certain nombre de populations, mais peut parfois augmenter la valeur d'un chromosome (performance).

En outre, elle permet d'éviter la convergence prématurée des algorithmes génétiques. Les méthodes de mutation sont connues de tous dans la littérature :

- **Le basculement d'un bit :**

Cela ne s'applique que dans les cas où les chromosomes sont codés en binaire, et il s'agit simplement de modifier le contenu d'un gène (0 devient 1, et 1 devient 0).

- **L'utilisation d'une borne :**

Il s'agit de remplacer un gène choisi au hasard par sa borne supérieure ou inférieure.

- **La méthode non-uniforme :**

Dans cette méthode, la probabilité de mutation change à travers l'algorithme génétique. Au début, elle reste faible pour faire stagner la population, mais vers la fin, elle devient élevée pour améliorer la valeur des solutions.

- **La méthode uniforme :**

Il remplace un gène par un nombre aléatoire entre ses bornes supérieure et inférieure.

- **La méthode gaussienne :**

Il sélectionne un gène et ajoute à son contenu une valeur choisie par la loi gaussienne. Le nouveau gène est écarté s'il est supérieur à la borne supérieure ou s'il est inférieur à la borne inférieure. Cette méthode et les trois précédentes sont spécifiques aux chromosomes, y compris les nombres entiers ou réels.

- **La méthode d'échange :**

Il s'agit juste de choisir arbitrairement deux gènes et d'invertir leur contenu.

- ✓ **La sélection :**

La sélection des individus à partir desquelles une nouvelle génération est créée peut se faire de plusieurs manières. Les méthodes de sélection les plus populaires sont : la méthode élitiste, la roue de la richesse, la sélection par rang, la sélection par tournoi et la sélection uniforme.

- **Méthode élitiste :**

Elle sélectionne les meilleurs individus en fonction de leur fitness. L'avantage de cette méthode est qu'elle ne permet pas de perdre les meilleurs individus au fil des générations, mais elle augmente aussi le risque d'une convergence prématurée vers un optimum local.

- **Méthode de la roue de la fortune :**

Chaque personne a la possibilité d'être choisie en fonction d'une probabilité qui dépend de ses performances. Ainsi, les personnes les plus aptes à résoudre le problème ont plus de chances d'être sélectionnées. Cette méthode répartit le nombre d'individus sur une roue de 360. Chacun d'entre eux a un pourcentage équivalent de fitness. Autrement dit, chaque élément de la population, qui a un fitness, a une portion mesurant degrés.

- **Sélection par rang :**

Elle consiste à trier les individus d'une population en fonction de leur fitness, puis à les numéroter par ordre décroissant. Ce qui signifie que celui qui a le plus petit fitness aura le numéro 1, et ainsi de suite.

Cette méthode fonctionne presque de la même manière que la roue de la fortune, à la seule différence que la part de la roue revenant à chaque individu dépend de son rang et non de son fitness. Avec cette méthode, chaque individu a la possibilité d'être sélectionnée mais la convergence vers la solution correcte est plus lente qu'avec la méthode de la roue de la fortune.

- **Sélection par tournoi :**

Il s'agit d'effectuer des tirages successifs avec remise de deux individus au sein de la communauté. A chaque tirage, une course permet de départager les deux candidats. Cette opération se fait indépendamment des autres personnes de la population.

En d'autres termes, la probabilité de victoire de chaque individu dépend de ses performances et de celles de son adversaire.

Cependant, la probabilité de victoire de l'individu qui a le fitness la plus élevée est déjà fixée entre 0,7 et 1, puisqu'elle est considérée comme un paramètre du processus génétique. Cette méthode permet de faire varier la pression de sélection en augmentant ou en diminuant la probabilité de victoire de la personne la plus intéressante.

- **Sélection uniforme :**

Elle se fait de manière aléatoire, uniforme, sans l'effet du fitness. Avec cette méthode, la sélection parmi les individus de la population est tout aussi probable.

#### **II.4.5. L'algorithme des colonies de fourmis :**

L'algorithme des colonies de fourmis est basé sur le comportement de ces animaux lorsqu'ils rassemblent de la nourriture.

Pendant ce temps, les études de ces animaux audacieux sont capables de trouver le chemin le plus court entre leur fourmilière et une zone où se trouve de la nourriture. Les fourmis communiquent entre elles de manière indirecte grâce à un produit chimique, appelé phéromone, qu'elles sécrètent continuellement et déposent le long des chemins qu'elles empruntent. Le but de cette opération est de marquer les chemins parcourus afin de les indiquer aux autres fourmis.

Cependant, la phéromone étant une matière qui s'évapore avec le temps, sa quantité est plus importante sur le chemin le plus utilisé. Ainsi, comme les fourmis sont non seulement capables de détecter la présence de phéromone mais aussi d'en mesurer l'ampleur, elles choisissent naturellement le chemin qui en contient le plus.

Les travaux de De Neubourg et al <sup>37 38</sup> ont contribué de manière significative à l'élucidation du comportement des fourmis naturelles. De Neubourg et al ont mis en relation une fourmi et une réserve de nourriture dans deux branches, pour étudier expérimentalement le lien entre les quantités de phéromones déposées et les comportements des fourmis. Ils ont fait de nombreuses expériences en faisant varier la diagonale entre les longueurs des deux branches.

Ainsi, ils ont vu que les fourmis commencent toujours par choisir librement leur chemin, mais dans la plupart des expériences, elles finissent toutes par utiliser la branche la plus courte. Ce résultat peut s'expliquer par le fait qu'au début d'une expérience, il n'y a pas de phéromone sur les branches.

Par conséquent, les fourmis n'ont aucune préférence. Elles choisissent donc avec la même probabilité n'importe laquelle des deux branches. Il se peut donc qu'une moitié des fourmis choisissent l'une des branches et l'autre moitié l'autre branche, mais le hasard peut aussi favoriser l'une des deux.

Cependant, comme l'une des branches est plus courte que l'autre, les fourmis choisies seront les premières à atteindre la source de nourriture et à entamer le chemin de retour.

Ainsi, au moment de choisir le chemin de retour, la phéromone sur la branche la plus courte les persuadera de la choisir et d'y placer encore plus de phéromone.

Par conséquent, la phéromone s'accumulera plus rapidement sur cette dernière, qui sera probablement utilisée par la majorité des fourmis. Dans le cas global, les chemins alternatifs sont représentés par un graphique dans lequel chaque fourmi se déplace pas à pas, en choisissant à chaque fois un arc à parcourir. Le premier algorithme de colonies de fourmis, nommé Ant System (AS), a été créé par Dorigo<sup>39 40</sup>, dans les années 1992.

Par la suite, des améliorations ont été apportées et d'autres variantes ont vu le jour. Les plus connues d'entre elles sont : la version rank-based (ASrank), le MAX-MIN Ant system (MMAS) et le Ant Colony System (ACS). L'algorithme de colonie de fourmis peut être décrit comme suit<sup>41</sup> :

**Pour**  $t = 1, \dots, t_{\max}$   
**Pour** chaque fourmi  $k = 1, \dots, m$   
**Choisir** une ville au hasard  
**Pour** chaque ville non visitée  $i$   
**Choisir** une ville  $j$ , dans la liste des villes restantes  
**Fin Pour**  
**Déposer** une quantité de phéromones sur le trajet  $T^k(t)$   
**Fin Pour**  
**Évaporer** les pistes  
**Fin Pour**

Lorsqu'une fourmi construit une solution, elle commence aléatoirement par parcourir un arc. Ensuite, elle choisit les prochains arcs, un par un, avec la probabilité :

$$P_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{i \in S_i^k} [\tau_{ii}(t)]^\alpha \cdot [\eta_{ii}]^\beta} \text{ if } j \in S_i^k, \text{ else } 0$$

- **Notations :**

- $\eta$  est à priori une information heuristique relative au problème étudié  $ij$  (Généralement, on a  $\eta_{ij} = 1/d_{ij}$ , où  $d_{ij}$  est la longueur de l'arc  $ij$ ).
- $\alpha$  et  $\beta$  sont deux paramètres qui déterminent respectivement l'influence des traces de phéromone et celle de l'information heuristique
- $S$  est l'ensemble des arcs qui ne sont pas encore parcourus par la fourmi  $k$  et qui sont  $i$   $k$  adjacents au sommet  $i$ .
- $\tau_{ij}$  La Quantité des phéromones sur l'arête  $(i,j)$   $ij$

✓ **Mise à jour des phéromones :**

Après un tour complet, chaque fourmi laisse une certaine quantité de phéromone sur l'ensemble de son parcours, quantité qui dépend de la qualité de la solution trouvée.

$$\Delta\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{si } (i, j) \in T^k(t) \\ 0 & \text{si } (i, j) \notin T^k(t) \end{cases}$$

- $T^k(t)$  est le trajet effectué par la fourmi  $k$  à l'itération  $t$ ,  $L^k(t)$  et la longueur de la tournée et  $Q$  un paramètre fixé.
- Le concept d'évaporation des pistes de phéromones est simulé à travers un paramètre appelé le taux d'évaporation  $0 < \rho < 1$
- Ne pas négliger toutes les mauvaises solutions obtenues

$$\tau_{ij}(t+1) \leftarrow (1 - \rho) \cdot \tau_{ij}(t) + \Delta\tau_{ij}(t)$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t)$$

- $t$  représente une itération donnée et  $m$  le nombre de fourmis Quant à l'algorithme ACS<sup>42 43 44</sup>, il améliore l'algorithme AS en suivant deux mécanismes :
  - Le premier consiste à ajouter de la phéromone uniquement aux arcs appartenant à la meilleure solution (qui peut être celle de l'itération courante, ou la meilleure solution globale).
  - Le second mécanisme concerne le choix des arcs. La méthode ACS est basée sur une règle dite "pseudo-random proportional" qui, selon une probabilité  $q_0$ , choisit le sommet  $j$  qui maximise le produit  $\Gamma(t) \cdot \eta$ , tandis qu'avec une probabilité  $1-q_0$  elle procède  $ij$   $\beta$  comme dans la méthode AS. Ainsi, si  $q_0 = 0$ , la méthode du "pseudo-random proportional" fonctionne de manière similaire à celle de l'algorithme AS.

Une autre différence entre l'algorithme ACS et AS est que l'évaporation des traces de phéromone se fait pendant la création des solutions. Ceci est fait pour réduire la probabilité que la même route soit suivie par toutes les fourmis (c'est-à-dire que cela facilite l'exploration en mettant en relation les effets des deux ajustements précédents, qui favorisent fortement le développement des connaissances acquises sur le problème étudié).

La différence ultime est que l'algorithme ACS utilise une méthode de recherche locale pour améliorer ses réponses en les transformant en optimums locaux. La 3ème version de l'algorithme des colonies de fourmis, connue sous le nom de MMAS<sup>45 46 47</sup>, introduit une limite supérieure et inférieure pour les valeurs des traces de phéromone, ainsi qu'une méthode d'initialisation de la phéromone qui est différente de celle utilisée dans les versions précédentes. En pratique, en utilisant l'algorithme MMAS, les valeurs des traces de phéromone sont incluses dans l'intervalle  $[\Gamma_{min}, \Gamma_{max}]$ .

En outre, les traces de phéromone sont initialisées à la valeur maximale, ce qui favorise une large investigation de l'espace de recherche, au début du processus.

Cependant, la mise à jour de la phéromone se fait comme dans la version ACS, en ajoutant de la phéromone uniquement sur les arcs qui appartiennent à la meilleure solution (qui peut être l'itération actuelle, ou celle trouvée au niveau global).

Une autre similitude entre l'algorithme MMAS et ACS est le fait qu'il utilise lui aussi des algorithmes de recherche locaux pour traduire ses solutions en optimums locaux. La méthode des colonies de fourmis a été créée à l'origine pour résoudre le problème du voyageur, mais elle est aujourd'hui utilisée pour résoudre d'autres problèmes.

Dans <sup>48</sup>, Yalaoui et al ont proposé une méthode de colonies de fourmis, appelée MOACO, pour résoudre le problème du stockage des conteneurs. L'accent a été mis sur le stockage des conteneurs exportés, afin de minimiser simultanément le coût global du stockage et la distance totale parcourue par les véhicules internes entre le parc de stockage et les plateformes.

Dans leur algorithme, ils ont considéré deux colonies qui effectuent chacune une tâche prédéfinie : l'une doit transporter le conteneur de chaque client dans un seul bloc, tandis que la seconde désigne une plate-forme pour chaque groupe de conteneurs appartenant à un seul client. Ils traitent donc le problème de manière séquentielle, en appliquant d'abord la première colonie, puis la seconde sur chaque groupe de conteneurs.

Cependant, les auteurs ont utilisé deux matrices de phéromone, chacune étant spécifique à la colonie. A chaque itération de leur méthode, ils mettent à jour uniquement les traces de phéromone appartenant aux solutions qui sont sur le front de Pareto. Yalaoui et al n'ont pas fourni de formulations mathématiques dans cette recherche, mais ils ont utilisé un algorithme de listage pour mesurer l'efficacité de leur technique de colonie de fourmis.

Les auteurs sont allés plus loin en renforçant leur algorithme de colonie de fourmis au moyen d'un algorithme de recherche locale (l'algorithme résultant est appelé MACO-LS). Pour ce faire, à chaque itération, les solutions non dominées sont modifiées en changeant les blocs et les quais qui sont alloués à des groupes de conteneurs particuliers tout en respectant les contraintes de capacité.

Après avoir effectué des simulations numériques, Yalaoui et al ont conclu que les deux algorithmes fournissent chacun de très bons résultats mais que MACO-S est plus puissant que MACO. L'approche proposée dans cet article semble intéressante ; cependant, les auteurs n'ont pas tenu compte du fait que l'allocation des navires aux postes à quai ne se fait pas en considérant uniquement les conteneurs qui sont exportés, mais surtout en tenant compte de la disponibilité des postes à quai et aussi de la longévité des navires.

Ainsi, pour minimiser la distance totale parcourue entre les blocs et les postes à quai, il serait intéressant de traiter le problème à l'envers ; en d'autres termes, étant donné une allocation de postes à quai, décider dans quels blocs affecter les conteneurs afin de minimiser simultanément les coûts de stockage et la distance totale à parcourir pour transporter ces conteneurs jusqu'au chantier naval.

# **Chapitre III**

## **L'entreprise portuaire d'Oran**

### III.1. Introduction :

Les travaux de construction du port d'Oran avaient commencé en 1792 par les Espagnols. A l'arrivée de français en 1930 ils ont repris des travaux qui ont commencé en 1948 et ne se terminent qu'en 1962 pour donner l'aspect actuel du port.

Actuellement le port d'Oran est considéré comme polyvalent car il traite les conteneurs, les marchandises diverses, les céréales... .

L'activité commerciale du port ne cesse d'évoluer d'année en année, ce qui fait de lui parmi les grands ports d'Algérie.

Le port de commerce est géré par l'Entreprise Portuaire d'Oran (EPO), sous la tutelle du groupe service portuaire (GSP) et du Ministère des travaux publics et des transports.

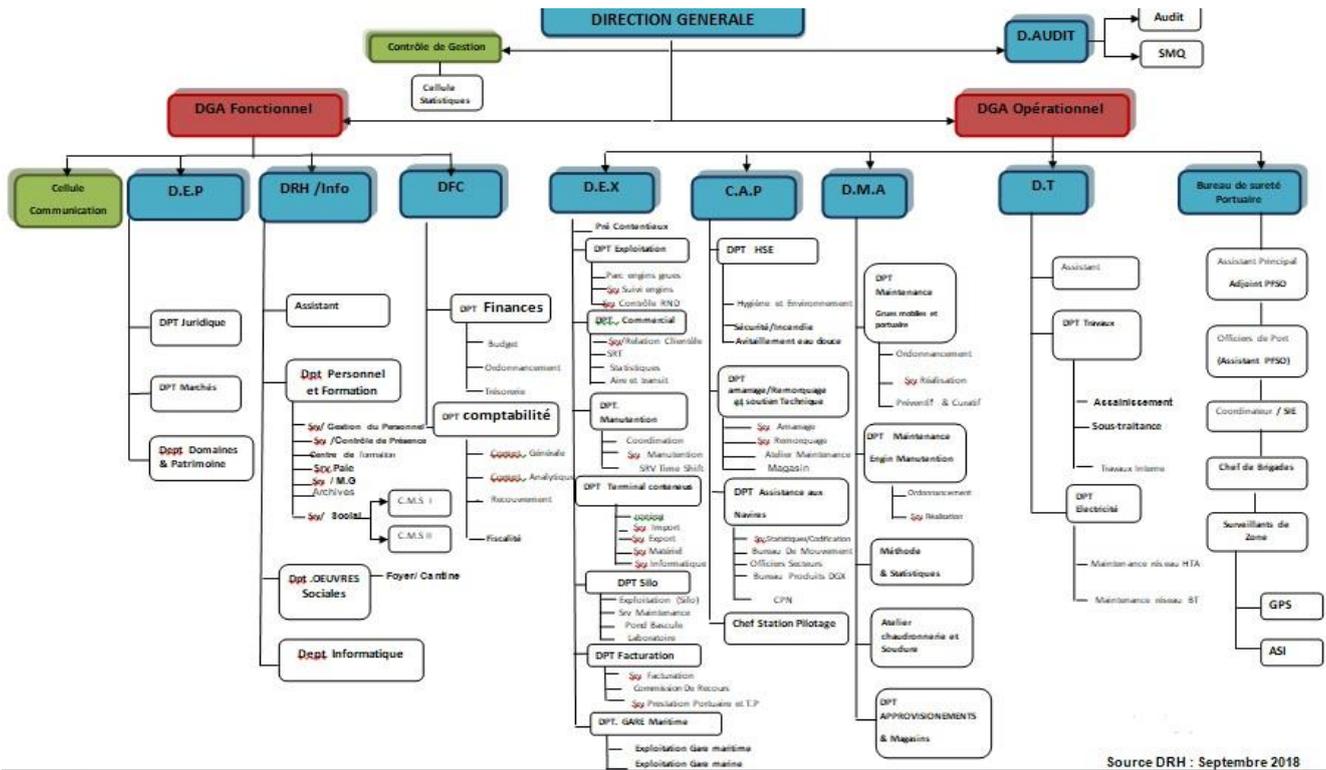
### III.2. Présentation de l'entreprise :

L'Entreprise Portuaire d'Oran "EPO", gère le port de commerce de la ville d'Oran, sous la tutelle du Groupe Services Portuaires et du Ministère Travaux Publics et des Transports, le port d'Oran considère comme un port polyvalent, traite les conteneurs, les marchandises diverses ainsi que les céréales de tout genre.

La superficie totale du port d'Oran a atteint **194** hectares, et il contient **7** bassins d'une superficie totale estimée à **122** hectares (plan d'eau). La longueur totale des quais du port atteint **4470** m et environ **23** quais dont **16** quais de travail et la longueur des quais varient entre **120** et **400** m et il y a un nouveau quai en phase d'achèvement au niveau du terminal à conteneurs, la profondeur de l'eau varie entre **8-13** mètres, et le port a une porte à l'est, **150** mètres de large et **25** mètres de profondeur, le tonnage maximal des navires est de **45000** tonnes et la longueur maximale des navires est de **225** m.

Le port d'Oran est divisé en plusieurs zones selon les installations spécialisées existante.

Organigramme de l'EPO



Source DRH : Septembre 2018

Figure III.1 : L'organigramme de l'EPO

III.3. Le terminal à conteneur du port d'Oran :

C'est une cour dans le port avec une série d'équipements et de secteurs qui vont ensemble dans un système harmonieux. Sa fonction est de charger, vider, ranger et stocker les conteneurs. Il se compose d'un quai et des zones de stockage.

Divise en plusieurs zones, le **Terminal à conteneurs** du port d'Oran est géré selon la zone, soit par **Cavaliers gerbeurs** pour le stockage des conteneurs de manière optimale, soit par **chariots élévateurs** pour les conteneurs frigorifiques ou le transfert des conteneurs sur les zones tampons.

Environ **200 000** conteneurs sont traités dans ce terminal au max, la superficie actuellement utilisée est de **12** hectares, et l'extension de **14** hectares n'est pas utilisée et la capacité de stockage actuelle est de **7 à 10** hectares, Il existe **05** quais affectés au terminal à conteneurs avec un tirant d'eau de **8 à 12** m, et il y a **10** magasins d'une surface de **22.063** m<sup>2</sup>.

➤ **Matériels de manutention et de stockage :**

- 32 Chariots élévateurs :  
  - Chariots élévateurs de 42 T.
  - Chariots élévateurs de 32 T.
  - Chariots élévateurs de 12 T.
  - Chariots élévateurs de 10 T.
- 08 Cavaliers Gerbeurs de 60 T.
- 06 Grues Portuaire de 63 T.

- 08 Grues Portuaires de 42 T.
- 33 Camions remorques.
- 26 Tracteurs (RO/RO – Rouliers).
- 20 Remorques 60 T.
- 5 Portiques à Grain.

### **III.4. Les procédures administratives :**

#### **III.4.1. Démarche import :**

Il représente le flux le plus important dans le port d'Oran vue l'importance des volumes du flux de conteneur entrées par rapport à l'expédié et c'est celui qui est en relation direct avec notre problème traité (stockage des conteneurs). C'est pour cela dans ce chapitre en vas bien détailler les étapes et les procédures que font partie de cette démarche.

##### **➤ Avant l'amarrage du navire :**

Après que le navire soit en rade une commission d'emplacement et mise en place. cette commission est constituée des représentants du capitainerie, direction acconage et manutention, l'agent maritime, la douane, la PAF. Le magasinier choisit le quai d'emplacement du navire et signe le bon de commande présenté par le capitaine maritime. Chaque direction reçoit une copie du PV de commission après l'amarrage du navire.

Le choix du quai est sous la responsabilité de la capitainerie. Ce choix est fait suivant la capacité du tirant d'eau vue que l'entreprise possède des quais de différente capacité. L'ordonnancement de la réception des navires est généralement fait suivant FIFO mais cette règle est flexible devant certaines situations pour optimiser le rendement de nos quais.

##### **➤ Après l'amarrage du navire dans le quai :**

Le capitaine maritime présente le manifeste de cargaison à la douane pour qu'il reçoit l'Autorisation d'accès au port.

Le manifeste de cargaison c'est une liste de tous les conteneurs qui vont être débarquer par le propriétaire.

Le capitaine maritime fait appel aux clients qui sont les récepteurs des conteneurs en leurs envoyant les avis d'arrivée. Le client reçoit le pli cartable qui est un dossier qui contient toutes les informations concernant la marchandise.

Il s'agit de la facture, le connaissance original (L'acte de propriété du marchandise), certificat de conformité, l'origine de la marchandise, les autorisations suivant la nature des marchandises. Ce dossier est envoyé par le fournisseur soit par DHL ou bien la poste, la banque, la compagnie maritime elle-même. Après la réception de ces documents le client (le transitaire) fait programmer la visite avec la douane.

**➤ Etape de débarquement :**

La direction acconage et manutention fait allouer les ressources nécessaires au processus du déchargement (les grues portique, les trieurs et les dockers), elle déclare un ODS (ordre de service) au chefs magasiniers et leurs pointeurs pour contrôler les conteneurs déchargés. le déchargement du navire est guidé à l'aide d'une liste de colisage fournie par le capitaine maritime ainsi que les instructions en temps réel pour assurer la stabilité du navire.

Chaque chef magasiniers possède 8 à 9 pointeurs, ils font partie de la direction terminale, ils travaillent suivant le système des shifts. Ils travaillent 3/4 shifts de 6h par jour (la 4<sup>ème</sup> brigade ne travaille pas le shift de nuit).

Ils sont repartis en deux, ceux qui sont chargés du navire et les autres chargés de la zone du stockage. Leur rôle est de contrôler les conteneurs débarqués et remplir la fiche de pointage et enregistrer toutes les données concernant le conteneur (numéro de conteneur, la marque, le type de conteneur (20 pieds / 40 pieds) numéro de scellé). En fin d'opération il établit un bordereau de réserve.

**➤ Étape fin de débarquement :**

Le chef magasiniers signe avec le commandant de bord le bordereau de réserve. il existe deux types : le premier type concerne le cas où il existe des conteneur abimés ou sans scellé, ainsi que des marchandises défectueuses. Le deuxième type (le différentielle) : il concerne deux cas :

Le premier cas représente les conteneurs manifestés et non débarqué, le deuxième cas représente les conteneurs débarqués et non manifestés.

Après que tout soit confirmé par le magasinier au sein de la direction, on fait la saisie numérique du manifeste et la codification des prestations faite par le service terminal pour établir la facturation. Finalement tous ses documents seront assemblé et archivé dans ce qu' on appelle un dossiers navire (manifeste de cargaison bon de commande liste de colisage le codifier et les facture).

**➤ Étape visite douanière :**

La durée du séjour des conteneurs dans le port est de 21 jours (article 205) dont le transitaire est sensé d'engager une déclaration et présenté tous les documents nécessaires, soit documents douaniers (la facture, certificat d'origine, l'euro) et les FAP (les formations administratives particulières) demandées par le gouvernement pour déterminer la position tarifaire de la marchandise et l'autorisation d'accès au territoire national. Une autre particularité c'est de faire conjointer la visite douanière et celle du service DCP pour prélèvement d'échantillon pour l'analyser afin d'avoir l'autorisation d'accès au territoire national. Une fois la déclaration est validée, elle passe finalement à la caisse.

Dans le cas du circuit vert l'étape de la visite douanière est éliminée. Pour le circuit rouge et orange en introduit la visite douanière : dans ce cas la durée de séjour du conteneur varie selon le nombre de conteneurs programmé par jours vue que le nombre des agents est limité (chaque agent peut visiter jusqu'aux 10 conteneurs par jours) ainsi que la

limitation des ressources disponibles par l'entreprise portuaire pour aménager les conteneurs dans la zone de visite. Les déclarations sont enregistrées dans le système de cotation. L'inspecteur ou l'agent responsable de la visite détermine le programme de visite (chaque jour à 14h).

Ça donne une durée de 24h à l'opérateur de l'entreprise portuaire pour préparer les conteneurs dédiés à la visite.

La douane établit un programme d'échantillonnage selon différents paramètres. (Type de marchandise, l'historique du client, port de chargement, des conventions signées avec le client et le reste passe par le scanner).

Contrairement à la marchandise de revente en état, dans ce cas le conteneur doit passer par la visite pendant 21 jours, Si le client ne se présente pas pendant cette période la loi oblige que le conteneur soit affecté au dépôt sous douane. Le tableau suivant représente un historique des conteneurs visités entre le 25-29 avril.

Le service douane estime une capacité maximale de 300 conteneurs visités par jour. On peut voir que cette limite n'était pas dépassée par le flux de conteneurs.

jour	nombre de conteneur sésité	nombre de conteneur enlevé
25	192	201
26	175	189
27	104	231
28	151	244
29	209	192

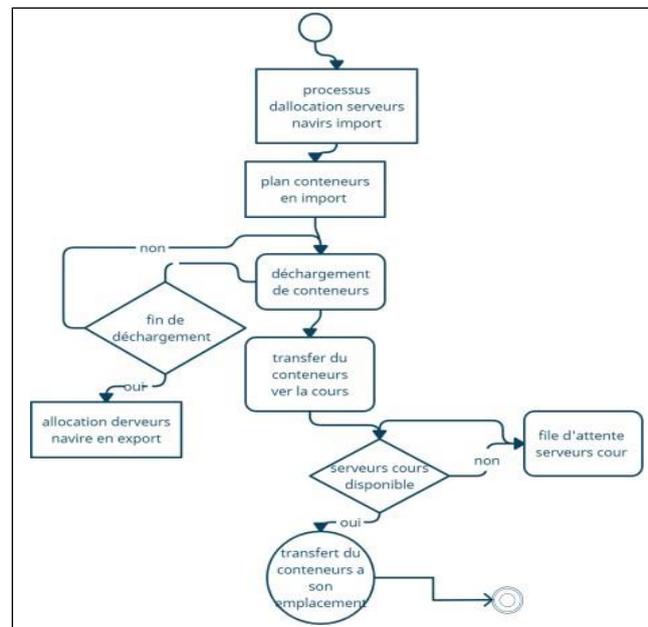


Figure III.2 : La procedure import

### III.4.2. La procédure export :

L'exportateur rapporte un conteneur vide ou plein, le conteneur en visite. Les conteneurs vides disposé d'une façon adjacente pris de quai ou se trouve le navire.

Pour les conteneurs pleins, ils sont stockés à part dans la zone de stockage en attendant la date d'exportation.

Les deux schémas ci-dessous représentent les démarches administratives de deux cas de conteneurs vides et pleins.

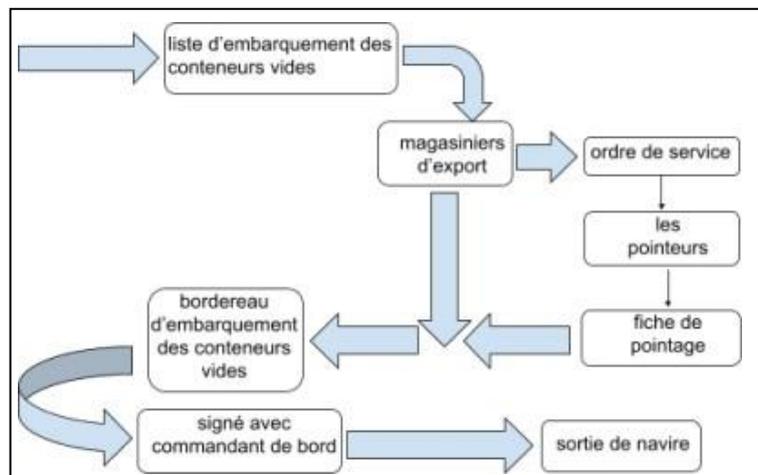


Figure III.3 : Conteneurs vides

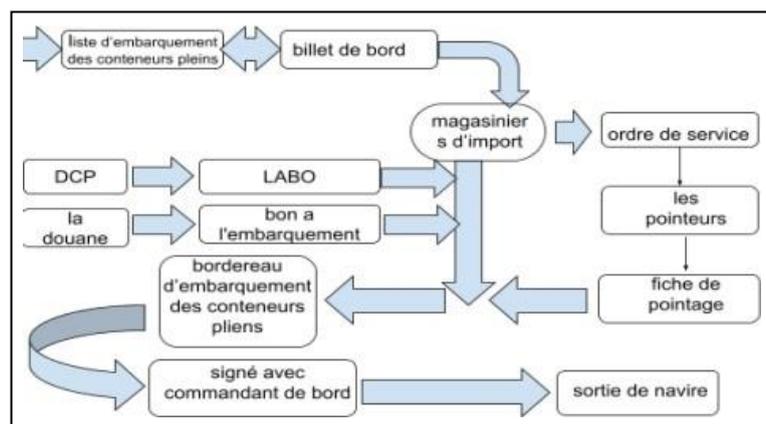


Figure III.4 : Conteneurs pleins

# **Chapitre IV**

## **Modélisation et résolution du problème de stockage du conteneur**

## IV.1. Introduction :

Le problème constaté au niveau du terminal d'Oran situe au niveau du remaniement des conteneurs, et dans la difficulté de localisation du conteneur à sortir, pour remédier a ces difficultés on doit recourir à l'utilisation de deux approche de résolution :

Une première approche qui utilise l'algorithme génétique, dans ce cas on a pris un exemple réduit pour résoudre le problème,

Une deuxième approche, on a proposé un algorithme satisfait aux conditions de travail dans le terminal d'Oran. Cet algorithme a été appliqué sur le programme Python.

## IV.2. Description du problème :

Nous considérant un terminal à conteneurs multimodal du port d'Oran, nous prenons en considération aussi les conteneurs entrants que sortants.

Les opérations de manutention et de transfert sont réalisées par :

- **Reach Stacker** : engin de manutention dans la zone d'opération.
- **Camion (mafi)** : engin de transférer le conteneur de la zone d'opération vers la zone de stockage.
- **Cavalier gerbeur** : engin de manutention dans la zone de stockage et transfert entre la zone de stockage et la zone de visite.

Dans notre problème nous tenons compte que les cavaliers gerber qui sont capable de transporter chacun un conteneur à la fois et d'empiler 3 conteneurs l'un sur l'autre.

On va s'intéresser qu'aux conteneurs entrants de type Dry, tandis que les autres conteneurs de différent type seront stocker ensemble dans un autre endroit.

Nous avons constaté qu'au niveau du terminal d'Oran il existe trop de mouvements improductifs qui font perdu beaucoup de temps, a cette effet notre objectif est d'essayer de minimiser aux maximum ces mouvements. Cela nous permet d'accélérer les opérations de livraison et de chargement des navires.

Le 2<sup>eme</sup> problème concerne la localisation du conteneur. A ce niveau le transitaire perd beaucoup de temps pour localiser son conteneur dans la zone de stockage, ce qui entraine un séjour plus important de celui-ci dans la cette zone, pour résoudre ce dernier problème, nous avons établi une base de données qui englobe les emplacements des conteneurs stockés.

Actuellement ce sont des ouvriers pointeurs qui font un recensement trois fois par semaine, l'application de notre base de données nous permet un recensement immédiat et s'en passer de ces ouvriers.

La résolution de PSC peut être traitée de façon statique ou dynamique. Dans le cas statique les conteneurs sont supposés tous arrivés au port avant le début des opérations de stockage, cependant dans le cas dynamique on prend en considération les arrivées et les départ imprévus et incertains des conteneurs après le début des opérations de stockage.

Dans le présent travail nous nous intéressons à la modélisation du cas dynamique du problème de stockage des conteneurs.

### IV.3. La complexité du problème :

Le PSC est classifié parmi les problèmes NP-difficile et NP-complets car c'est un problème de programmation linéaire en nombre entier, et comme celui-ci est NP-difficile alors le PSC est aussi NP-difficile<sup>49</sup>.

### IV.4. Approche de résolution

#### IV.4.1. La résolution de PSC par l'utilisation d'un algorithme génétique :

L'algorithme génétique est une méta-heuristique qui manipule une population de solution à la fois. C'est un algorithme de recherche basé sur les mécanismes de la sélection naturelle et de la génétique. Il combine une stratégie de « survie des plus forts » avec un échange d'information aléatoire mais structuré.

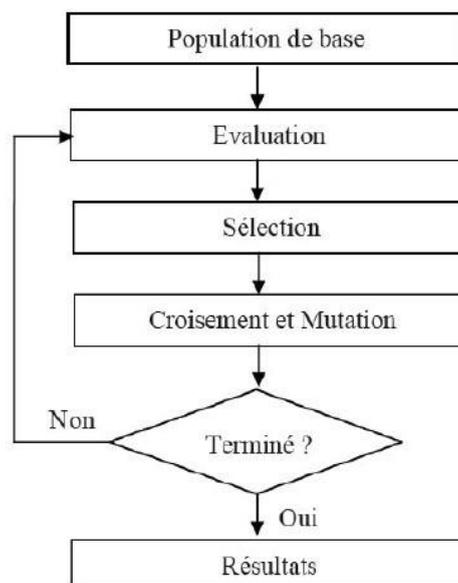


Figure IV.1 : Structure de l'AG

- Lors de l'arrivée des conteneurs à la zone de stockage on aborde le problème de stockage des conteneurs. La résolution finale c'est l'affectation de ces conteneurs à des emplacements disponibles dans les piles de façon optimale.

On suppose qu'on a une zone de stockage réduite comprenant 20 piles et on veut affecter 8 conteneurs à des emplacements libres dans ces piles d'une façon optimale, afin de minimiser le nombre de remaniement et la distance totale entre les emplacements affectés et le navire pour accélérer le déchargement de navire.

La zone de stockage réduite est représentée dans la figure ci-dessous :

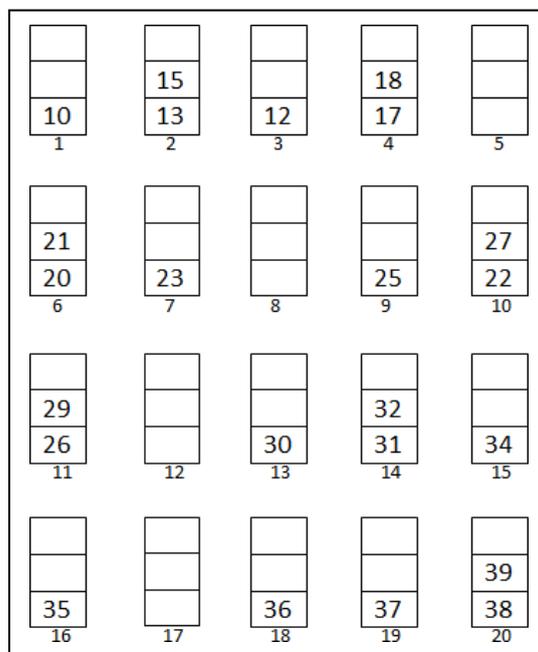


Figure IV.2 : L'organisation de la zone de stockage

Liste des conteneurs :

Id conteneur	1	2	3	4	5	6	7	8
dimension	20	20	40	20	40	40	20	40

Liste des piles :

Pile	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Dimension	20	20	40	20	20	20	40	40	20	40	20	40	40	20	20	40	20	40	40	20

Matrice des distances

6 10 20 18 22  
 (12 22 32 24 34)  
 18 34 44 30 40  
 30 40 56 42 46

Nombre d'emplacement libre dans chaque pile :

$$\begin{pmatrix} 2 & 1 & 2 & 1 & 3 \\ 1 & 2 & 3 & 2 & 1 \\ 1 & 3 & 2 & 1 & 2 \\ 2 & 3 & 2 & 2 & 1 \end{pmatrix}$$

Liste des conteneurs qu'on veut les visiter :

C : 13, 12, 32, 26, 37, 27, 20.

#### IV.4.2. Le modèle mathématique :

Pour la résolution du problème de stockage des conteneurs nous proposons le modèle mathématique suivant :

- **Les indices :**

C : indique le conteneur  $c \in \{1, 2, \dots, N_c\}$ .

P : indique la pile  $p \in \{1, 2, \dots, N_p\}$

I : indique l'emplacement dans la pile  $i \in \{1, 2, 3\}$

t : période de l'horizon de planification.

- **Données concernant les conteneurs :**

$A_0$  : L'ensemble des conteneurs qui sont déjà stockés dans la cour au début de shift.

$A_t$  : L'ensemble des conteneurs arrivés

$A_v$  : L'ensemble des conteneurs qui vont les visités dans cette période.

$R_c$ : la dimension des conteneurs (20, 40).

- **Données concernant les piles :**

$P_{\delta}^c$  : La pile qui contient le conteneur  $c \in A_0$ .

$I_{\delta}^c$ : La position du conteneur  $c \in A_0$  dans la pile ou il est stockée.

$C_p(t)$  : le nombre d'emplacement libre dans la pile à la fin de la période t.

$r_p$ : la dimension de la pile p.

- **Variables de décision :**

$$x_{pi}^c(t) = \begin{cases} 1, & \text{si le conteneur } c \text{ stocké dans la position } (p, j) \text{ dans la periode } t \\ 0, & \text{si non} \end{cases}$$

$$y_{pI}^c(t) = \begin{cases} 1, & \text{si le conteneur } c \text{ est appartient à la liste des conteneur à visité} \\ 0, & \text{si non} \end{cases}$$

$$z_{p_i, p'_{i'}}^c(t) = \begin{cases} 1, & \text{si le conteneur } c \text{ est déplacé de l'emplacement } (p_i) \text{ vers } (p'_{i'}) \\ 0 & \text{si non} \end{cases}$$

- **Fonction objective :**

$$\text{Min } \sum_{c=1}^{Nc} \sum_{p=1}^{Np} \sum_{i=1}^3 15 * z_{pi,p'i'}^c + \sum_{c=1}^{Nc} \sum_{p=1}^{Np} d_p x_p$$

- **Les contraintes :**

$$x_{p_0, p_0}^c(0) = 1. \quad \star c \in A_0 \quad (1)$$

Cette contrainte précise les emplacements de stockage qui sont déjà occupés au début de shift.

$$\sum_{p=1}^{Np} \sum_{i=1}^3 x_{pi}^c \quad \star c \in A_t \quad (2)$$

Cette contrainte assure que chaque conteneur est affecté à un seul emplacement de stockage a son arrivé.

$$\sum_{p=1}^{Np} \sum_{i=1}^3 x_{pi}^c(t) + \sum_{p=1}^{Np} \sum_{i=1}^3 y_{pi}^c(t) + \sum_{p=1}^{Np} \sum_{i=1}^3 z_{pj,pj}^c(t) \leq 1 \quad (3)$$

$\star c \in *1 \dots Nc+$

Plusieurs mouvements ne peuvent à la fois prendre place.

$$\sum_{c=1}^{Nc} \sum_{i=1}^3 x_{pi}^c \leq \phi \quad \star p \in *1 \dots Np+ \quad (4)$$

Le nombre des conteneurs affecté à une pile est inférieur ou égale à la somme d'emplacement libre dans cette dernière.

$$\sum_{p=1}^{Np} \sum_{i=1}^3 x_{pi}^c = 0 \quad \star c \in A, \text{ Tel que } R_c \geq r_p \quad (5)$$

Les conteneurs qui sont affectés à une pile ont les mêmes dimensions.

$$x_{pj}^c(t) \geq x_{pj+1}^c(t) \quad \star j \in *1,2,3+ , \star c \in A_t \quad (6)$$

Le remplissage de chaque pile se fait de bas en haut sans sauter aucun emplacement.

$$c_p(t) = c_p(t-1) - \sum_{c \in A_t} \sum_{i=1}^3 x_{pi}^c(t) + \sum_{c \in A_v} \sum_{i=1}^3 y_{pi}^c(t) + \sum_{c=1}^{Nc} \sum_{i=1}^3 z_{pi,pi}^c(t) - \sum_{c=1}^{Nc} \sum_{i=1}^3 z_{c,FF}^c(t) \quad (7)$$

$$\star p \in \{1 \dots N_p\}$$

Mis à jour de la capacité de chaque pile à la fin de chaque mouvement.

$$x_{pi}^c, y_{pi}^c, z_{pi}^c = *0,1+. \quad c_p \in N \quad d_p \in R$$

Cette contrainte précise la nature de chaque variable de décision.

#### IV.4.3. Le processus de résolution :

L'AG que nous propose pour la résolution du PSC comprend quatre étapes :

- La création d'une population initiale.
- La sélection d'une partie de cette population.
- Le croisement entre les individus sélectionnés.
- La mutation d'un individu.

##### - La création d'une population initiale :

La population initiale doit être suffisamment diversifiée et de taille assez importante pour que la recherche puisse parcourir l'espace d'état dans un temps limité.

Dans notre algorithme génétique, chaque individu de la population représente un solution valide du PSC, nous représentons chaque individu sous forme d'un tableau qui a deux lignes et dont le nombre de colonnes égal au nombre de conteneurs à stocké. Les conteneurs sont représentés dans la première ligne alors que les piles sont représentés dans la deuxième ligne.

(1)	1	2	3	4	5	6	7	8
	1	4	7	6	12	8	15	18

(2)	1	2	3	4	5	6	7	8
	4	17	3	11	8	8	20	10

(3)	1	3	4	5	8	2	6	7
	5	8	6	13	13	5	7	20

(4)	8	2	5	1	3	6	4	7
	7	4	10	1	19	19	11	40

(5)	1	7	6	5	2	3	4	8
	17	15	10	12	6	12	2	18

(6)	4	5	2	1	7	8	6	3
	9	16	2	11	1	16	8	8

**Calcule la fonction objective (Fitness):**

Après la création de la population on doit calculer la fonction objective qui est la somme des distances entre les piles concernées par l'affectation des conteneurs et la zone d'opération plus la somme du nombre de remaniement (on considère qu'un remaniement représente une distance de 15 mètre).

$$F_1 = 230$$

$$F_2 = 285$$

$$F_3 = 259$$

$$F_4 = 267$$

$$F_5 = 290$$

$$F_6 = 185$$

**- La sélection d'une partie de cette population :**

La sélection est la première phase de renouvellement d'une population. Ils existent différents méthodes de sélection, nous allons appliquer à notre algorithme génétique la sélection binaire par tournoi ; deux individus sont choisis au hasard, on compare leurs fonctions d'adaptation et le mieux adapté est sélectionné.

$$\begin{array}{l} \{(1,4) \rightarrow 1 \\ (2,5) \rightarrow 2 \end{array} \quad \begin{array}{l} \{(3,6) \rightarrow 6 \\ (2,3) \rightarrow 3 \end{array} \quad \begin{array}{l} \{(1,6) \rightarrow 6 \\ (5,1) \rightarrow 1 \end{array}$$

**- Le croisement entre les individus sélectionnés :**

Sert à créer de nouveaux individus, favorise l'exploration de l'espace de recherche et enrichit la diversité de la population. Il est fait avec deux parents pour génère deux enfants. Il existe différentes méthodes de croisement ; nous allons appliquer la méthode de croisement en un point.

## Croisement



Croisement entre(1,2)

1	2	3	4	5	6	7	8
1	4	7	6	8	8	20	10

1	2	3	4	5	6	7	8
4	17	3	11	12	8	15	18

Croisement entre (6,3)

4	5	2	1	8	2	6	7
9	16	2	11	13	5	7	20

1	3	4	5	7	8	6	3
5	8	6	13	1	16	8	18

Croisement entre (6,1)

4	5	2	1	5	6	7	8
9	16	2	11	12	8	15	18

1	2	3	4	7	8	6	3
1	4	7	6	1	16	8	8

Après le croisement on doit faire une correction par ce que certains individus ne satisfont pas à toutes les contraintes, (dans notre cas la contrainte c'était ; chaque conteneur est affecté à un seule emplacement).

La population après la correction :

- 9<sup>ème</sup> individu ; on remplace 2 par 3.
- 10<sup>ème</sup> individu ; on remplace 3 par 2.
- 11<sup>ème</sup> individu ; on remplace 5 par 3.
- 12<sup>ème</sup> individu ; on remplace 3 par 5.

(7)

1	2	3	4	5	6	7	8
1	4	7	6	8	8	20	10

(8)

1	2	3	4	5	6	7	8
4	17	3	11	12	8	15	18

(9)

4	5	2	1	8	3	6	7
9	16	2	11	13	5	7	20

(10)

1	3	4	5	7	8	6	2
5	8	6	13	1	16	8	18

(11)

4	5	2	1	3	6	7	8
9	16	2	11	12	8	15	18

(12)

1	2	3	4	7	8	6	5
1	4	7	6	1	12	8	8

#### - La mutation d'un individu

C'est un processus qui est effectué à la fin de chaque itération, où un changement mineur du code génétique est appliqué à un individu pour introduire de la diversité et ainsi d'éviter de tomber dans des optimums locaux. Pour ce faire, on choisit aléatoirement un individu de la population et un conteneur que l'on affecte à une autre pile qui est elle aussi choisie aléatoirement parmi celles qui ne sont pas pleines et qui satisfont toutes les contraintes.

(12)

1	2	3	4	7	8	6	5
1	4	7	6	1	12	8	8

Calcule la fonction objective :

$$F_7 = 232.$$

$$F_8 = 288.$$

$$F_9 = 246.$$

$$F_{10} = 249.$$

$$F_{11} = 274.$$

$$F_{12} = 177.$$

D'après la première itération, la meilleure affectation est représentée par le 12<sup>ème</sup> individu, car il nous donne la plus petite valeur de fonction objective.

- Dans notre cas au port d'Oran on peut pas appliqué L'algorithme génétique à cause des conditions et des manières de travail concernant le stockage et le déstockage :

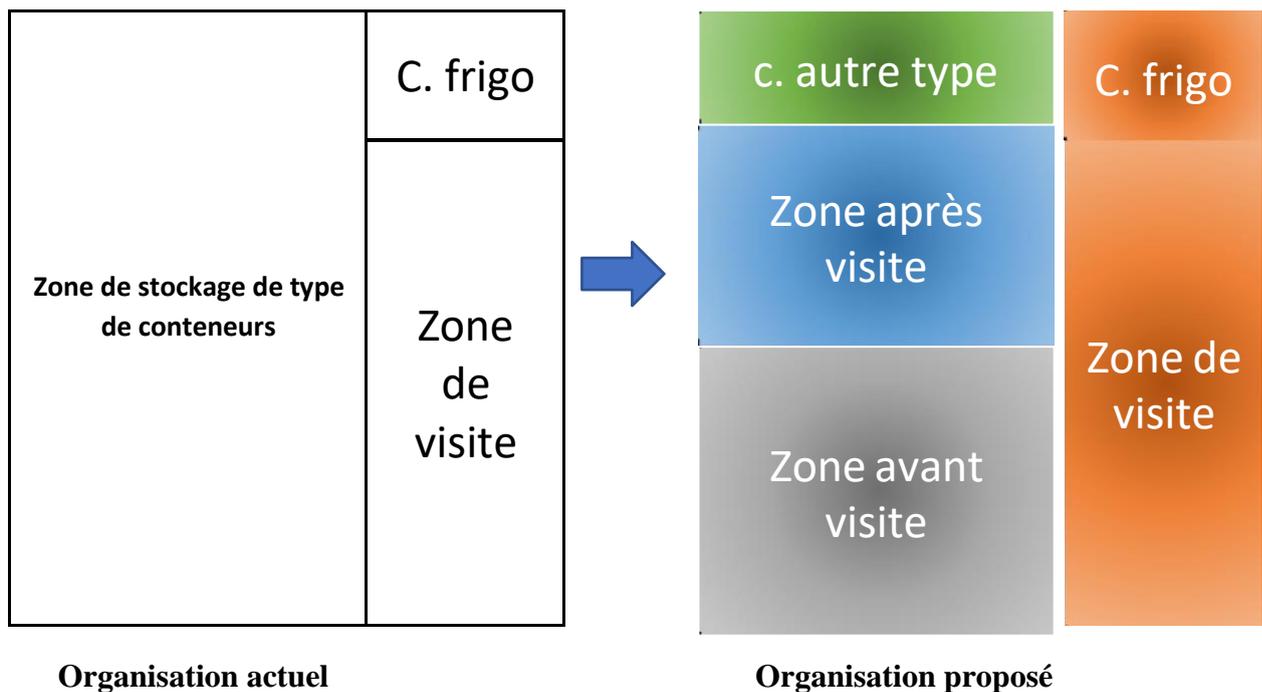
L'opération de stockage et du déstockage se fait en parallèle.

La livraison des conteneurs n'est pas organisée d'une façon chronologique.

Si pour ça on a choisi une autre approche de résolution et cela on proposant une procédure ou un algorithme qui respecte toutes les contraintes de travail réel de stockage et déstockage au niveau du terminal.

#### IV.5. La résolution de PSC par l'utilisation d'un algorithme « proposé » appliqué sur le programme Python :

##### IV.5.1 Organisation de la zone de stockage



##### IV.5.2. Processus de résolution :

Tout d'abord on doit connaître :

- L'ensemble des conteneurs qui doivent quitter pendant cette période (donnée par la douane).
- Les emplacements qui sont déjà occupés (donnée par notre base de données).
- Le conteneur qu'on veut le stocker ainsi que sa dimension (entrer par l'opérateur).

➤ **Stockage :**

Quand l'opérateur entre le nom de conteneur et sa dimension notre programme va affecter ce conteneur vers un emplacement dans une pile vide tout en respectons la contrainte de dimension.

S'il n'y a pas d'une pile vide alors on affecte ce conteneur vers une pile qu'a un seul conteneur stocker tel que ce conteneur n'appartient pas à la liste des conteneurs a visités.

Si non on l'affecte dans une pile qu'a deux conteneurs stockés tel que ces derniers n'appartient pas à la liste des conteneurs a visités.

➤ **Déstockage :**

On commence de retirer les conteneurs qui sont au sommet de la pile ça veut dire le dernier emplacement occupé (soit 3<sup>eme</sup>, 2<sup>eme</sup>, ou 1<sup>er</sup> position).

Si la position de notre conteneur se situe sous un conteneur non concerner, on enlève le conteneur non concerner et on se tire notre conteneur voulu.

Le conteneur non concerner sera déposé dans une pile (dans la même line) vide si non une pile a un seul conteneur si non dans une pile a deux conteneur tel que le conteneur de la 1<sup>ère</sup> position ne soit pas concerné par notre liste.

Si la position de notre conteneur se situe sous deux conteneurs non concernés, on enlève les deux conteneurs non concernés et on se tire notre conteneur voulu.

Les deux conteneurs non concernés leur dépose s'effectue de la même procédure faite précédemment.

#### IV.5.2. Algorithme

Les entrer :

Choix, Rc : entier

;names : char ;

**Ecrire** (« sort ou stock ½ ») ;

**lire** (choix) ;

**Si** choix == 1

fonction sorti ;

**Si non**

**Si** choix == 2

**Ecrire** (« Entrer le nom de conteneur : ») ;

**lire** (names) ;

**Ecrire** (« Entrer la dimension de » , names) ;

**lire** (Rc) ;

fonction insert\_ stock ;

**Fin si**

**Si non**

**Ecrire** (« please entrer 1 ou 2 ») ;

**Fin si**

#### La fonction de stockage

**Fonction** insert\_stock (names, dimension\_conteneur, list\_visita) ;

#nbr de pile dans chaque line.

c : tableau sur [0..20] d'entier =

[14,14,14,16,14,14,15,15,14,14,14,13,14,13,14,14,13,14,14,13,14]

#affecter une dimension pour chaque pile

**Pour** i allant de [21]

**Si** i == 0

**alors** dimension : tableau sur [0..NMax-1] d'entier =

[20,20,40,20,40,40,20,20,20,40,20,40,20,40]

**Si non**

**Si** i == 1

**alors** dimension : tableau sur [0..NMax-1] d'entier =

[20,40,40,20,20,40,40,20,20,20,40,20,20,40]

....

.....

.....

**Si non**

**Si** i == 20

**alors** dimension : tableau sur [0..NMax-1] d'entier = [20,20,40,20,40,40,20,20,20,40,20,40,20,40]

**Fin Si**

#affecter les conteneurs au premier emplacement

**Pour p allant de (c[i]**

**Si** emplacement1==0 & emplacement2==0 & emplacement3==0

**Si** dimension\_conteneur == dimension [p]

**alors** emplacement1 = names

emplacement2 = '0'

emplacement3 = '0'

**Ecrire** (names« est dans la »char(i+1),« line, »char(p+1)« pile, premier emplacement »)

**mis à jour** #mis à jour de la base de donné

**Fin Si**

**Pour i allant de [21]**

**Si** i == 0

**alors** dimension : tableau sur [0..NMax-1] d'entier = [20,20,40,20,40,40,20,20,20,40,20,40,20,40]

**Si non**

**Si** i == 1

**alors** dimension : tableau sur [0..NMax-1] d'entier = [20,40,40,20,20,40,40,20,20,20,40,20,20,40]

....

....

.....

**Si non**

**Si** i == 20

**alors** dimension : tableau sur [0..NMax-1] d'entier = [20,20,40,20,40,40,20,20,20,40,20,40,20,40]

**Fin Si**

#affecter les conteneurs au deuxième emplacement

**Pour p allant de (c[i]**

**Si** emplacement1==1 & emplacement2==0 & emplacement3==0

**Si** dimension\_conteneur == dimension [p] & emplacement1[0][0] **non** de list\_visita

**alors** emplacement1 = names

emplacement2 = names

emplacement3 = '0'

**Ecrire** (names « est dans la »char(i+1),« line, »char(p+1)« pile, 2<sup>eme</sup> emplacement »)

**mis à jour** #mis à jour de la base de donné

**Fin Si**

**Pour i allant de [21]**

**Si** i == 0

**alors** dimension : tableau sur [0..NMax-1] d'entier = [20,20,40,20,40,40,20,20,20,40,20,40,20,40]

**Si non**

**Si** i == 1

**alors** dimension : tableau sur [0..NMax-1] d'entier = [20,40,40,20,20,40,40,20,20,20,40,20,20,40]

....

....

.....

**Si non**

**Si** i == 20

```

alors dimension : tableau sur [0..NMax-1] d'entier = [20,20,40,20,40,40,20,20,20,40,20,40,20,40]
Fin Si
#affecter les conteneur au troisième emplacement
Pour p allant de (c[i])
  Si emplacement1==1 & emplacement2==1 & emplacement3==0
    Si dimension_conteneur == dimension [p] & emplacement1[0][0] non de list_visita &
    emplacement2[0][0] non de list_visita
      alors emplacement1 = names
      emplacement2 = names
      emplacement3 = names
    Ecrire (names« est dans la »char(i+1),« line, »char(p+1)« pile,3eme emplacement »)
    mis à jour #mis à jour de la base de donn e
  Fin Si

```

### La fonction de sortie (d stockage)

```

Fonction sorti (list_visita) :
#retirer les conteneurs qui sont au sommet
Pour Av allant de liste_visita
  Si Av in emplacement1 & emplacement2 ==‘0‘
    lire (Av, « est un sommet »)
    lire (Av, « est dans la line », str(i+1) « ,pile » str(p[0][0] « ,emplacement1 »)
    supprimer Av
    mis a jour #mis   jour de la base de donn es
  Si non
    Si Av in emplacement2 & emplacement3 ==‘0‘
      lire (Av, « est un sommet »)
      lire (Av, « est dans la line »,str (i+1) « ,pile »str (p[0][0] « ,emplacement2 »)
      supprimer Av
      mis   jour
    Si non
      Si Av in emplacement3
        lire (Av, « est un sommet »)
        lire (Av, « est dans la line »,str(i+1) « ,pile »str (p[0][0] « ,emplacement3 »)
        supprimer Av
        mis   jour

#retirer les conteneur qui sont au sommet-1
Pour Av allant de liste_visita
  Pour table allant de (21)
    Si Av in emplacement1 & emplacement2 != __0‘ & emplacement3 ==‘0‘
      lire (Av, « not sommet »)
      lire (Av, « est dans la line » str(table+1) « ,pile »str(p[0][0] « ,emplacement 1 »)
      Pour p allant de (table)
        Si emplacement1 == __0‘
          lire (deplacer«str(emplacement2[0][0]«vers line»str(table+1)«,pile»str(p[0][0] «,emplacement1»)

```

```

Si non
  Si emplacement1 == _'1' & emplacement2 == _'0'
    lire (deplacer«str(emplacement2[0][0]«vers line»str(table+1)«,pile»str(p[0][0]
«,emplacement2»)
  Si non
    Si emplacement1 =='1' & emplacement2 == _'1' & emplacement3 =='0'
      lire (deplacer«str(emplacement2[0][0]«vers line»str(table+1)«,pile»str(p[0][0]
«,emplacement3»)
    Fin Si
mis à jour

Si non
  Si Av in emplacement2 & emplacement3 !='0'
    lire (Av « not sommet »)
    lire (Av « est dans la line » str(table+1) « ,pile » str(p[0][0]) « ,emplacement2 »)
    Pour p allant de (table)
    Si emplacement1 == _'0'
      lire (deplacer«str(emplacement3[0][0]«vers line»str(table+1)«,pile»str(p[0][0] «,emplacement1»)
    Si non
      Si emplacement1 == _'1' & emplacement2 == _'0'
        lire (deplacer«str(emplacement3[0][0]«vers line»str(table+1)«,pile»str(p[0][0]
«,emplacement2»)
      Si non
        Si emplacement1 =='1' & emplacement2 == _'1' & emplacement3 =='0'
          lire (deplacer«str(emplacement3[0][0]«vers line»str(table+1)«,pile»str(p[0][0]
«,emplacement3»)
        Fin Si
mis à jour

#retirer les conteneur qui sont au sommet-1
Pour Av allant de liste_visita
  Pour table allant de (21)
    Si Av in emplacement1 & emplacement2 != '0' & emplacement3 != '0'
      lire (Av « ,not sommet »)
      lire (Av « est dans la line »str(table+1) « ,pile »str(p[0][0]) « ,emplacement1 »)
      Pour p allant de (table) ;
        Si emplacement1 == _'0'
          lire (deplacer«str(emplacement3[0][0]«vers line»str(table+1)«,pile»
str(p[0][0] «,emplacement1»)
          lire (deplacer«str(emplacement2[0][0]«vers line»str(table+1)«,pile»
str(p[0][0] «,emplacement1»)
        Si non
          Si emplacement1 == _'1' & emplacement2 == _'0'
            lire (deplacer«str(emplacement3[0][0]«vers line»str(table+1)«,pile»
str(p[0][0] «,emplacement2»)
            lire (deplacer«str(emplacement2[0][0]«vers line»str(table+1)«,pile»

```

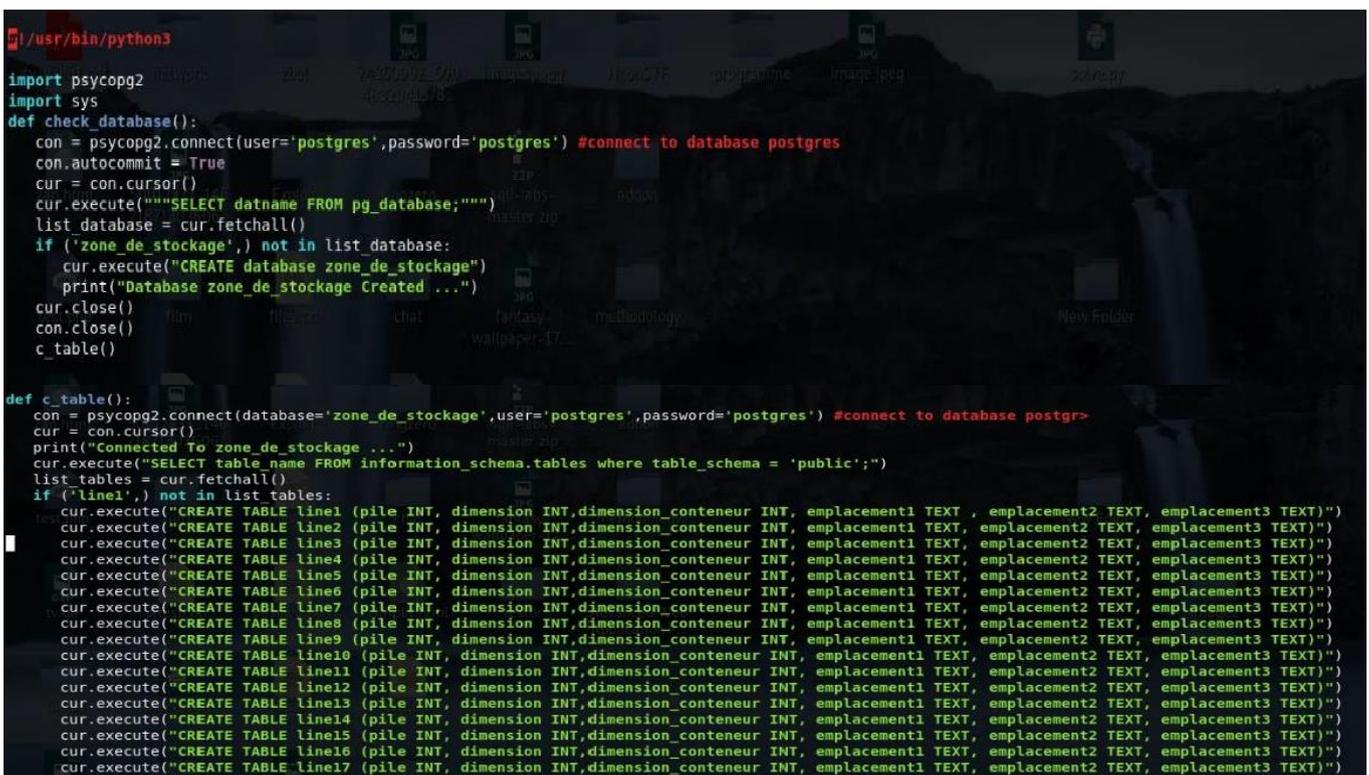
```

str(p[0][0] «,emplacement2»)
    Si non
        Si emplacement1 ==“1“ & emplacement2 == _‘1“ & emplacement3 ==“0“
            lire (deplacer«str(emplacement3[0][0]«vers line»str(table+1)«,pile»
str(p[0][0] «,emplacement3»)
            lire (deplacer«str(emplacement2[0][0]«vers line»str(table+1)«,pile»
str(p[0][0] «,emplacement2»)
    Fin Si
Si non
    pass
    mis à jour

```

### IV.5.3 Description du code Python

#### ➤ La création de la base de données



```

#!/usr/bin/python3
import psycopg2
import sys
def check_database():
    con = psycopg2.connect(user='postgres',password='postgres') #connect to database postgres
    con.autocommit = True
    cur = con.cursor()
    cur.execute("""SELECT datname FROM pg_database;""")
    list_database = cur.fetchall()
    if ('zone_de_stockage',) not in list_database:
        cur.execute("""CREATE database zone_de_stockage""")
        print("Database zone_de_stockage Created ...")
    cur.close()
    con.close()
    c_table()

def c_table():
    con = psycopg2.connect(database='zone_de_stockage',user='postgres',password='postgres') #connect to database postgres
    cur = con.cursor()
    print("Connected To zone_de_stockage ...")
    cur.execute("""SELECT table_name FROM information_schema.tables where table_schema = 'public';""")
    list_tables = cur.fetchall()
    if ('line1',) not in list_tables:
        cur.execute("""CREATE TABLE line1 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT , emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line2 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line3 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line4 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line5 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line6 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line7 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line8 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line9 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line10 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line11 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line12 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line13 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line14 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line15 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line16 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")
        cur.execute("""CREATE TABLE line17 (pile INT, dimension INT,dimension_conteneur INT, emplacement1 TEXT, emplacement2 TEXT, emplacement3 TEXT)""")

```

Figure IV.3 : La creation de la de base de donnée

### ➤ La fonction stocker

```
def insert_stockage(names,dimension,conteneur,list_visita):
    c = [14,14,14,16,14,14,15,15,14,14,14,13,14,13,14,14,13,14]
    con = psycopg2.connect(database='zone_de_stockage',user='postgres',password='postgres')
    cur = con.cursor()
    #dimension = 20
    for i in range(21):
        if i == 0:
            dimension = [20,20,40,20,40,40,20,20,20,40,20,40,20,40]
        elif i == 1:
            dimension = [20,40,40,20,20,40,40,20,20,20,40,20,20,40]
        elif i == 2:
            dimension = [40,20,20,40,20,20,20,20,20,40,20,40,40,40]
        elif i == 3:
            dimension = [20,20,40,20,20,40,20,40,20,20,20,20,40,20,20]
        elif i == 4:
            dimension = [40,20,20,40,20,20,40,40,40,20,20,20,40,20]
        elif i == 5:
            dimension = [40,20,40,20,40,20,20,20,40,40,20,20,20,40]
        elif i == 6:
            dimension = [20,20,40,40,20,20,40,20,20,20,20,40,20,40,20]
        elif i == 7:
            dimension = [40,40,20,20,20,20,20,40,20,40,20,20,20,40]
        elif i == 8:
            dimension = [20,40,20,40,40,20,40,20,20,40,20,20,40,20]
        elif i == 9:
            dimension = [20,20,40,40,20,40,20,40,40,20,20,40,20,20]
```

Figure IV.4 : La création de la fonction stocker

### ➤ La fonction sortie

```
def sorti(list_visita):
    con = psycopg2.connect(database='zone_de_stockage',user='postgres',password='postgres') #connect to database postgres
    cur = con.cursor()
    # list_visita = ['cx318']
    list_visit6 = []
    list_visit5 = []
    list_visit4 = []
    list_visit3 = []
    list_visit2 = []
    list_visit1 = []
    for Av in list_visita:
        for table in range(21):
            cur.execute("select emplacement3 from line"+str(table+1)+" where emplacement1 = '"+Av+"' or emplacement2 = '"+Av+"' or emplacement3 = '"+Av+"'")
            em3 = cur.fetchall()
            cur.execute("select emplacement2 from line"+str(table+1)+" where emplacement1 = '"+Av+"' or emplacement2 = '"+Av+"' or emplacement3 = '"+Av+"'")
            em2 = cur.fetchall()
            cur.execute("select emplacement1 from line"+str(table+1)+" where emplacement1 = '"+Av+"' or emplacement2 = '"+Av+"' or emplacement3 = '"+Av+"'")
            em1 = cur.fetchall()
            if (Av,) in em1 and ("0",) in em2 and ("0",) in em3:
                list_visit1.append(Av)
                break
            elif (Av,) in em2 and ("0",) in em3:
                list_visit2.append(Av)
                break
            elif (Av,) in em3:
                list_visit3.append(Av)
                break
```

Figure IV.5 : La création de la fonction sortie

Dans notre programme on a réalisé deux tâches : stocker et retirer

L'opérateur (cavalier gerbeur) s'occupe du retrait des conteneurs qui sont marqués dans la liste à visiter, quand le camion mafi ramène un conteneur de la zone d'opération, l'opérateur va stocker ce conteneur. Pour retirer il entre le numéro 1 et pour stocker le numéro 2.

```

root@befour:~# cd Desktop
root@befour:~/Desktop# cd programme/
root@befour:~/Desktop/programme# nano test2.py
root@befour:~/Desktop/programme# python3 test2.py
Connected To zone de stockage ...
Sort ou stock 1/2 : █

```

Figure IV.6 : Le choix (déstocker ou stocker)

- Par exemple s'il entre le numéro 1 :

Le programme va indiquer la position de conteneur pour le retirer

```

Sort ou stock 1/2 : 1
['ct455']
ct455 not sommet
ct455 est dans la line 11, pile 10, emplacement2
deplacer ct764 vers line11, pile 12, emplacement1
root@befour:~/Desktop/programme# █

```

Figure IV.7 : La position de conteneur à visiter

Le déstockage se fait suivant d'un ordonnancement optimal pour éviter les mouvements improductifs.

- Si l'opérateur entre le numéro 2 :

Le programme va demander le nom et la dimension de conteneur qu'on veut stocker, après il va nous donner un emplacement vide pour affecter ce conteneur.

```

Sort ou stock 1/2 : 2
Entrer le nom de conteneur : ct1189
Entrer la dimension de ct1189 : 40
ct1189 est dans la 1 line, 12 pile, premier emplacement
root@befour:~/Desktop/programme# █

```

Figure IV.8 : La position où stocker le conteneur

Le stockage se fait suivant un ordonnancement optimal pour éviter les mouvements improductifs lors le déstockage.

- La mise à jour de la base de données se fait immédiatement.

```

zone_de_stockage=# SELECT * FROM line21;

```

pile	dimension	dimension conteneur	emplacement1	emplacement2	emplacement3
6	20	20	ct281	ct908	0
5	20	20	ct278	0	0
1	40	40	ct285	ct593	ct898
2	40	40	ct287	ct594	ct899
8	40	40	ct289	ct596	ct904
9	20	20	ct286	ct585	ct912
11	40	40	ct291	ct910	0
7	20	20	ct907	ct907	0
14	20	20	ct294	ct907	0
13	20	20	ct293	ct579	0
10	40	40	ct290	0	0
3	20	20	ct911	ct900	0
4	40	40	ct288	0	0
12	20	20	ct292	ct583	ct909

```

(14 rows)
zone_de_stockage=#

```

Figure IV.9 : Les emplacements des conteneurs sur la base de données

### ➤ La création d'un fichier de situation journalière

Chaque début de journée notre programme crée un fichier pour nous indiquer toutes les opérations exécutées et leur horaire d'exécution « la situation ».

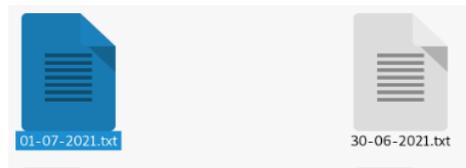


Figure IV.10 : Le type de fichier créé

```

*01-07-2021.txt
~/Desktop/programme

[00:38:59]ct1070 vers la 8 line, 5 pile, premier emplacement
[00:45:07]ct179 est dans la line 13, pile 9, emplacement1
[00:48:15]ct178 est dans la line 13, pile 7, emplacement1
[00:48:51]ct804 est dans la line 13, pile 13, emplacement3
[00:49:05]ct185 est dans la line 13, pile 13, emplacement1
[00:49:05] deplacer ct484 vers line13, pile 14, emplacement3
[01:49:08]ct503 est dans la line 15, pile 5, emplacement2
[01:51:44]ct837 est dans la line 15, pile 14, emplacement3
[01:51:48]ct831 est dans la line 15, pile 1, emplacement3
[01:51:51]ct512 est dans la line 15, pile 1, emplacement2

```

Figure IV.11 : Fiche de situation journalière

#### IV.5.4. Exemple de simulation :

Pour tester le rendement de notre programme on a simulé 20 conteneurs stockés et 13 conteneurs retirés tel que, le temps moyen de stockage est 1min et de déstockage 2min et pour le remaniement des conteneurs 0,6min.

```

root@befour:~/Desktop/programme# nano test.py
root@befour:~/Desktop/programme# python3 test.py
Connecté To zone_de_stockage ...
entrez le fichier qui contient les conteneurs et les dimensions pour le stockage
iliste stocker.txt
entrez la list de destockage : sortie.txt
ct1710 est un sommet
ct1710 est dans la line 4, pile 3, emplacement1
ct1722
ct1722 vers la 17 line, 12 pile, premier emplacement
ct1709 est un sommet
ct1709 est dans la line 6, pile 4, emplacement1
ct648 est un sommet
ct648 est dans la line 4, pile 6, emplacement3
ct1723
ct1723 vers la 4 line, 3 pile, premier emplacement
ct1724
ct1724 vers la 6 line, 4 pile, premier emplacement
ct1725
ct1725 vers la 18 line, 11 pile, premier emplacement
ct706 est un sommet
ct706 est dans la line 9, pile 2, emplacement3
ct1726
ct1726 vers la 18 line, 9 pile, premier emplacement

ct1733 vers la 6 line, 11 pile, premier emplacement
ct1734
ct1734 vers la 3 line, 4 pile, premier emplacement
ct1735
ct1735 vers la 1 line, 4 pile, deuxieme emplacement
ct1736
ct1736 vers la 9 line, 2 pile, premier emplacement
ct1737
ct1737 vers la 1 line, 7 pile, deuxieme emplacement
ct1738
ct1738 vers la 1 line, 13 pile, deuxieme emplacement
ct1739
ct1739 vers la 2 line, 5 pile, deuxieme emplacement
ct1740
ct1740 vers la 2 line, 10 pile, deuxieme emplacement
ct1741
ct1741 vers la 3 line, 3 pile, deuxieme emplacement
ct1742
ct1742 vers la 4 line, 12 pile, deuxieme emplacement
le stockage prit 21 min
le desockage prit 30 min
le deplacement des conteneurs ils ont pris 4.8 min

root@befour:~/Desktop/programme#

```

Figure IV.12 : La simulation des conteneurs

D'après les résultats de simulation on a constaté que le rendement d'un seul cavalier gerbeur dans une heure était :

- 20 conteneurs stockés.
- 13 conteneurs retirés.

Dans l'état actuel du port d'Oran le rendement d'un seul cavalier gerbeur dans une heure est approximativement :

13 conteneurs stockés.

6 conteneurs retirés.

#### **IV.5.5. Conclusion**

Notre travail s'est effectué en deux parties :

la première partie : création de la base de données de la zone de stockage, cela nous aide dans la localisation des conteneurs sans avoir recours au recensement et au transitaire.

la deuxième partie : création d'une méthode de stockage qui nous permet de minimiser le nombre de remaniement ce qui entraîne un gain de temps dans le déstockage.

La solution acquise d'après ce travail nous permet d'améliorer le rendement au niveau de la zone de stockage comparativement au travail s'effectue actuellement (approximativement, 450 conteneur stocké par jour et 200 conteneurs déstockés par jour).

à savoir un stockage de 700 conteneurs et un déstockage de 450 conteneurs (approximativement) par jour.

## CONCLUSION GENERALE

A l'ère de l'économie de marché, toute activité commerciale recherche à améliorer sa façon de faire pour atteindre une rentabilité maximale dans le domaine du transport maritime, le port et particulièrement le terminal à conteneur représente la pièce maitresse dans la gestion du commerce maritime pour atteindre un bon rendement dans cette activité.

Notre sujet étant choisi dans cette optique, car la problématique qui se pose à ce niveau, c'est les différentes opérations et manipulations des conteneurs au niveau du terminal, qui nécessitent du temps et des frais supplémentaires, qui si elles ne sont pas optimisées influeraient largement sur le rendement.

Des études bibliographiques ont été menées pour une bonne prise de la connaissance de ce domaine, ainsi qu'un stage pratique au port d'Oran pour voir de près l'exécution du travail au niveau du terminal et cerner tous les problèmes se rapportant aux manipulations inutiles et mouvements improductifs qui influent négativement sur le rendement.

Par la suite des solutions ont été proposées par l'utilisation des procédés mathématiques. En effet par l'utilisation de notre algorithme proposé, on est arrivé à solutionner le problème de remaniement des conteneurs lors du stockage et du déstockage, et ainsi éviter les mouvements improductifs.

Afin de résoudre ce problème on a dû créer une base de données. En plus du problème de remaniement cette base de données permet aussi de résoudre le problème de localisation des conteneurs dans la zone de stockage.

---

## BIBLIOGRAPHIE

- [1] "Conteneur, conteneurisation — Géoconfluences." 8 avr.. 2021, <http://geoconfluences.ens-lyon.fr/glossaire/conteneur-conteneurisation>. Date de consultation : 7 juin. 2021.
- [2] K. H. Kim, Y. M. Park, and K. R. Ryu. Deriving decision rules to locate export containers in container yards. *European Journal of Operational Research*, 124 :89–101, 2000
- [3] Y. A. Saanen et R. Dekker, Intelligent stacking as way out of congested yards ? part 1, *Port Technol Int*, vol. 31, pp. 87-92, 2007.
- [4] Y. Ma et K. H. Kim, A comparative Analysis : Various storage rules in container yards and their performances, *Industrial engineering & Management systems*, vol. 11, pp. 276-287, 2012.
- [5] M. Taleb-Ibrahimi, B. D. Castilho, et C. F. Daganzo, Storage space and handling work in container terminals, *Transportation Research Part B : Methodological*, vol. 27, pp. 13-32, 1993.
- [6] R. Dekker, P. Voogd, et E. V. Asperen, Advanced methods for container stacking, *Container Terminals and Cargo Systems*, pp. 131-154, 2007
- [7] Y. A. Saanen et R. Dekker, Intelligent stacking as way out of congested yards ? part 1, *Port Technol Int*, vol. 31, pp. 87-92, 2007.
- [8] Y. Ma et K. H. Kim, A comparative Analysis : Various storage rules in container yards and their performances, *Industrial engineering & Management systems*, vol. 11, pp. 276-287, 2012.
- [9] B. Borgman, E. V. Asperen, et R. Dekker, Online rules for container stacking, *OR Spectrum*, vol. 32, pp. 687-716, 2010.
- [10] T. Chen, Yard operations in the container terminal-a study in the "unproductive" moves, *Marit Policy Manage*, vol. 26, pp. 27-38, 1999.
- [11] B. Castilho et C. F. Daganzo, Handling strategies for import containers at marine terminals, *Transportation Research Part B : Methodological*, vol. 27, pp. 151-166, 1993
- [12] M. Taleb-Ibrahimi, B. D. Castilho, et C. F. Daganzo, Storage space and handling work in container terminals, *Transportation Research Part B : Methodological*, vol.

- [13] M. B. Duinkerken, J. J. M. Evers, et J. A. Ottjes, A Simulation Model For Integrating Quay Transport And Stacking Policies On Automated Container Terminals , Proceedings of the 15 th European Simulation Multiconference, Prague, 2001.
- [14] R. Dekker, P. Voogd, et E. V. Asperen, Advanced methods for container stacking, Container Terminals and Cargo Systems, pp. 131-154, 2007.
- [15] B. Borgman, E. V. Asperen, et R. Dekker, Online rules for container stacking, OR Spectrum, vol. 32, pp. 687-716, 2010.
- [16] B. Borgman, E. V. Asperen, et R. Dekker, Online rules for container stacking, OR Spectrum, vol. 32, pp. 687-716, 2010.
- [17] M. Taleb-Ibrahimi, B. D. Castilho, et C. F. Daganzo, Storage space vs handling work in container terminals, Transportation Research Part B : Methodological, vol. 27, pp. 13-32, 1993.
- [18] B. Castilho et C. F. Daganzo, Handling strategies for import containers at marine terminals, Transportation Research Part B : Methodological, vol. 27, pp. 151-166, 1993.
- [19] K. H. Kim et K. Y. Kim, Optimal price schedules for storage of inbound containers, Transportation research, Part B, vol. 41, pp. 892-905, 2007.
- [20] K. H. Kim, Evaluation of the number of rehandles in container yards, Computers & Industrial Engineering, vol. 32, pp. 701-711, 1997.
- [21] D. Sculli et C. F. Hui, Three dimensional stacking of containers, Omega, vol. 16, pp. 585-594, 1988.
- [22] Y. Ma et K. H. Kim, A comparative Analysis : Various storage rules in container yards and their performances, Industrial engineering & Management systems, vol. 11, pp. 276-287, 2012.
- [23] X. Jiang, L. H. Lee, Y. Han, et K. C. Tan, A container yard strategy for improving land utilization and operation eciency in a transshipment hub port , European journal of operational research, vol. 221, pp. 64-73
- [24] R. Bellman, Dynamic programming, Princeton university press, Princeton, NJ, 1957.
- [25] J. Culioli, Introduction à l'optimisation, Ellipses, 2e édition, 2012.
- [26] J. Culioli, Introduction à l'optimisation, Ellipses, 2e édition, 2012.

[27] K. H. Kim et J. W. Bae, Re-marshaling export containers in port container terminals, *Computers & Industrial Engineering*, vol. 35, pp. 655-658, 1998.

[28] K. H. Kim, Y. M. Park, et K. Ryu, Deriving decision rules to locate export containers in containers yards, *European journal of operational research*, vol. 124, pp. 89-101, 2000.

[29] Y. Lee et N. Hsu, An optimization model for the container pre-marshaling problem, *Computer & operations research*, vol. 34, pp. 3295-3313, 2007.

[30] M. Yu et X. Qi, Storage space allocation models for inbound containers in an automatic container terminal, *European journal of operational research*, vol. 226, pp. 32-45.

[31] J. H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Computers, 1975.

[32] Mitsuo Gen et Runwei Cheng, *Genetic Algorithms and Engineering Optimization*, Industrial Engineering / Manufacturing, 2007.

[33] Mitsuo Gen et Runwei Cheng, *Genetic Algorithms and Engineering Optimization*, Industrial Engineering / Manufacturing, 2007.

[34] Mitsuo Gen et Runwei Cheng, *Genetic Algorithms and Engineering Optimization*, Industrial Engineering / Manufacturing, 2007.

[35] Mitsuo Gen et Runwei Cheng, *Genetic Algorithms and Engineering Optimization*, Industrial Engineering / Manufacturing, 2007.

[36] Mitsuo Gen et Runwei Cheng, *Genetic Algorithms and Engineering Optimization*, Industrial Engineering / Manufacturing, 2007.

[37] J.-L. Deneubourg, S. Aron, S. Goss, et J.-M. Pasteels. The self-organizing exploratory pattern of the Argentine ant, *Journal of insect behavior*, vol. 3, pp. 159-168, 1990.

[38] S. Goss, S. Aron, J. L. Deneubourg, et J. M. Pasteels, Self-organized shortcuts in the Argentine ant, *Naturwissenschaften*, vol. 76, pp. 579-581, 1989.

[39] M. Dorigo, *Optimization, learning and natural algorithms (in Italian)*, PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Italy, pp. 140, 1992.

[40] M. Dorigo, V. Maniezzo, et A. Coloni. The ant system : optimization by a colony of cooperating agents, IEEE transactions on systems, man, and cybernetics - part B, vol. 26, pp. 29-41, 1996.

[41] cours 5<sup>ème</sup> année métaheuristiques chapitre 2 mr Fouad Maliki

[42] M. Dorigo et L. M. Gambardella, Ant colonies for the traveling salesman problem, BioSystems, vol. 43, pp. 73-81, 1997.

[43] M. Dorigo et L. M. Gambardella, Ant colony system : a cooperative learning approach to the traveling salesman problem, IEEE Transaction on evolutionary computation, vol. 1, pp. 53-66, 1997

[44] L. M. Gambardella et M. Dorigo, Solving symmetric and asymmetric TSPs by ant colonies. In proceedings of the 1996 IEEE International conference on evolutionary computation (ICEC'96), pp. 622-627, IEEE press, Piscataway, NJ, 1996.

[45] T. Stützle et H. H. Hoos, The MAX-MIN Ant System and local search for the traveling salesman problem, In T. Bäck, Z. Michalewicz, and X. Yao editors, Proceeding of the 1997 IEEE international conference on evolutionary computation (ICEC'97), pp. 309-314, IEEE Press, Piscataway, NJ, 1997.

[46] T. Stützle, Local search algorithms for combinatorial problems : analysis, improvements, and new applications, Inx, Sankt Augustin, Germany, 1999.

[47] T. Stützle et H. H. Hoos, MAX-MIN Ant System, Future generation computer systems, vol. 16, pp. 889-914, 2000.

[48] F. Belmecheri-Yalaoui, F. Yalaoui, et L. Amodeo, Multi-objective Ant Colony Optimization Method to Solve Container Terminal Problem, Applications of Multi-Criteria and Game Theory Approaches, part I, pp. 107-122, 2014.