PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC

**MOHAMED BOUDIAF UNIVERSITY - M'SILA**
**FACULTY OF MATHEMATICS AND COMPUTER SCIENCE**

**DEPARTEMENT OF COMPUTER SCIENCE**

# End of studies DISSERTATION

**Presented to obtain the MASTER'S Degree**

**Domain** : Mathematics and Computer Science

**Branch** : Computer Science

**Specialty** : Advanced Information Systems

**By** : BOUDIAF Fatima

**SUBJECT**

# Organization and resolution of the container storage problem in container terminal

**Publicly supported on : 06/01/2016 in front of the jury members :**

| | | |
|---|---|---|
| **Dr. S BOUAMAMA** | **University of M'sila** | **President** |
| **Prof. B BOUDERAH** | **University of M'sila** | **Supervisor** |
| **Dr. M BOUNIF** | **University of M'sila** | **Examiner** |

**Promotion : 2015 /2016**

PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC

**MOHAMED BOUDIAF UNIVERSITY - M'SILA**
**FACULTY OF MATHEMATICS AND COMPUTER SCIENCE**

**DEPARTEMENT OF COMPUTER SCIENCE**

**End of studies DISSERTATION**

**Presented to obtain the MASTER'S Degree**

**Domain** : Mathematics and Computer Science

**Branch** : Computer Science

**Specialty** : Advanced Information Systems

**By** : BOUDIAF Fatima

## SUBJECT

# Organization and resolution of the container storage problem in container terminal

**Publicly supported on : 06/01/2016 in front of the jury members :**

| | | |
|---|---|---|
| **Dr. S BOUAMAMA** | **University of M'sila** | **President** |
| **Prof. B BOUDERAH** | **University of M'sila** | **Supervisor** |
| **Dr. M BOUNIF** | **University of M'sila** | **Examiner** |

**Promotion : 2015 /2016**

# ACKNOWLEDGMENTS

First of all, I thank **ALLAH**, the almighty, for giving me the strength to carry on this project and for blessing me with many great people who have been my greatest support in both my personal and professional life.

Secondly, I would like to seize this opportunity to express my deepest regards and gratitude to my supervisor Dr Professor **BOUDERAH Brahim** for his dedication and sincere effort throughout the whole stages of bringing this project into light.

My sincere thanks go to my family who supported me during my studies, as well as for their help and their understanding.

My sincere thanks go also to the jury members for their interest in my research by agreeing to review my work and to enrich their proposals.

Finally, I would like to thank all the people who contributed in some way to the work described in this dissertation.

# TABLE OF CONTENTS



## CHAPTER 1:

## GENERALITY ABOUT CONTAINER TERMINAL

## CHAPTER 2:

## STATE OF THE ART REVIEW



## CHAPTER 3:

## THE METHOD PROPOSED FOR SOLVING THE OPTIMIZATION PROBLEM

# CHAPTER 4:

# PRESENTATION OF SOFTWARE PROPOSED

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

**ACC** : **A**lgerian **C**ode of **C**ommerce

**AGV** : **A**utomated **G**uided **V**ehicles.

**ALNS** : **A**daptive **L**arge **N**eighborhood **S**earch.

**ALV** : **A**utomated **L**ifting **V**ehicles.

**ANSI** : **A**merican **N**ational **S**tandards **I**nstitute.

**ASCs** : **A**utomated **S**tacking **C**ranes.

**CNAN** : « ***C**ompagnie **N**ationale **A**lgérienne de **N**avigation* » The Algerian National Navigation Company.

**CO** : **C**ombinatorial **O**ptimization

**CPE** : « ***C**onseil de **P**articipation de l'**E**tat* » the Participation Council of the State.

**CSP** : **C**ontainer **S**torage **P**roblem.

**CSS** : **C**ontainer **S**torage **S**ystem.

**CT** : **C**ontainer **T**erminal.

**GUI : G**raphical **U**ser **I**nterfaces

**HACG** : **H**ybrid **A**nt **C**olony and **G**enetic algorithm.

**ILP** : **I**nteger **L**inear **P**rogramming.

**ISO** : **I**nternational **O**rganization for **S**tandardization.

**LB** : **L**ower **B**ound

**LEM** : « ***L**aboratoire **D**'étude **M**aritime* » Maritime laboratory study.

**LP** : **L**inear **P**rogramming.

**M/V** : **M**erchant **V**essel.

**MILP** : **M**ixed **I**nteger **L**inear **P**rogramming.

**PAD** : « ***P**ort **A**utonome de **D**akar »* the Autonome Port of Dakar.

**QCs** : **Q**uay **C**ranes.

**RTG** : **R**ubber **T**ired **G**antry.

**SA** : **S**imulated **A**nnealing.

**SC** : **S**traddle **C**arriers.

**SPA** : « ***S**ociétés **P**ar **A**ctions* » Corporations.

**SSAP** : **S**torage **S**pace **A**llocation **P**roblem.

**TEU** : **T**wenty-foot **E**quivalent **U**nit.

**TSP** : **T**raveling **S**alesman **P**roblem

**UB** : **U**pper **B**ound

**USA :** **U**nited **S**tates of **A**merica

**VB** : **V**isual **B**asic

**VI** : **V**alid **I**nequalities.

# GENERAL INTRODUCTION

# GENERAL INTRODUCTION

Containerization has played an important role in the development of international networks transport intermodal. A major international transport network based on the use of a standard cargo size grew. This standardization has accelerated the transfer of containers from one mode of transport to another but at the same time it has made almost impossible the manual controls. Therefore, further efforts imposed themselves for the creation of means of technical controls.

Today transfer operations from one mode of transport to another remain the key element of an efficient transport system. Among these transfer points, port container terminals are usually identified as an important link in the global supply chain. Container terminals are essential intermodal interfaces for the global transportation network. They play an important role in modern logistics as a shipping center. The competitiveness of a terminal is mainly reflected by its efficiency of transfer due to charges paid by a boat which depend on the rotation time and the number of containers loaded and unloaded at a terminal. Following the considerable growth of container transport by sea, handling operations in container terminals have risen sharply in recent years. Indeed, efficient handling of containers in terminals.

## 1 Problematic:

The field of port terminals is very large; it consists of several sub-problems where each problem has its own criteria. Among the various problems that arise in this area are: storage containers problem in port terminal, operations in a marine container terminal, problem of container movements in marine terminal…etc.

Within this broad area of port terminals we have chosen to treat one of these sub-problems, which concern the containers storage problem in container terminal. In which we seek the best way to store containers with a good organization of storage.

## 2 Aim of study:

The container storage optimization in a port terminal is a major logistical problem that has attracted the attention of researchers for many decades. Two large storage optimization axes are studied: optimization of the storage time and optimizing storage space. These two problems are often treated separately. The objective of our study is to optimize the containers storage in container terminal with a good organization. Therefore, we chose the CSP as a field of application that seems to be effective methods of combinatorial optimization to solve our problem.

## 3 Methodology:

The methodology that we have used to achieve the objectives of this work it's like follows; at first, we have described the domain of port terminals in detail in order to understand its current operation and to identify the characteristics, objectives and constraints of this system. Subsequently, the analysis of this problem in order to highlight the CSP compared to other related problems in the same domain, particularly those who are included in the storage problems.

This problem is very complex, many research has already helped to formulate the problem in different ways, the majority of studies using the optimization algorithms for solving the problem of containers storage in port terminal, such as this thesis [27]. The amongst several problems of the CSP there are a lot of studies about the adaptation of the harmony search algorithm to solve the storage allocation problem for inbound and outbound containers like this thesis [15]. Zhang and col. [13] solved the (SSAP) using a rolling-horizon approach. Both outbound and inbound containers are considered. In [6] Kim and Park proposed a heuristic decision rule and a sub-gradient optimization technique to solve the storage space allocation for outbound containers. With proper study of existing researches, it has been possible to find interesting links to develop an approach for solving the particular problem of container storage problem. To make the approach more realistic resolution, incorporating specific constraints observed the storage problem.

At first, it was necessary to develop an approach to perform the resolution of the problem. This includes the establishment of an appropriate combinatorial algorithm taking into account the constraints specifically related to the problem, and finally the proposed simulation software. The software that has been developed for the resolution of SCP has done by the Branch and Cut algorithm. We have used the Visual Basic language for programming.

## 4 Layout:

This dissertation is organized into four chapters. In the first chapter, we introduce the concepts related to container terminal, in which we present the Algerian National Navigation Company. We give also a general idea about the port container terminals.

In the second chapter we speak about the history of containerization. We cite the different studies that are done already concerning the field of port container terminals. Then we present the container storage problem in general.

Our main work and our personal contribution bound in the third chapter, in which we describe our problem and we propose a mathematical model for the CSP, then we present the Branch and Cut algorithm and we justify our choice of algorithm after this we show how we adapted the Branch and Cut algorithm for the resolution of our problem. The implementation of our proposed algorithm with its technical results is presented in the last chapter. Finally, we close our modest dissertation by a general conclusion and future works.

# CHAPTER 1

# GENERALITY ABOUT CONTAINER TERMINAL

## Chapter Summary:

In this chapter we present the CNAN and we describe its history, and its role. In the second step, we specify some useful concepts to understand their main work for readers unfamiliar with the world of container terminal. In third step, we talk about the port container terminals its functions and its structure also the loading and discharging a ship at the end we address to the operations of stacking.

---



---

# 1 Presentation :

## 1.1 CNAN:

The Algerian National Navigation Company ( in French "*La Compagnie Nationale Algérienne de Navigation*" CNAN ) is a state company that proven social and economic efficiency in cargo business over the past couple of decades. The CNAN GROUP SPA is foreign company registered in the trade register has been active for 13 years. Based in Algeria, it was specialized in the sector of maritime and coastal passenger transport [30]

### 1.1.1 History:

The company CNAN NORD Spa is a subsidiary of CNAN GROUP Spa. It premiered January 02, 2005 and provides the maritime carriage of goods by regular lines from the ports of Antwerp (Belgium), Hamburg (Germany), Bilbao, Aviles and Castellone (Spain), Lisbon (Portugal), Houston (USA) and Gemlik and Kumport (Turkey). In accordance with the resolution of the CPE N° 06/71/16/09/2006 CNAN NORD Spa became owner as of 02/12/2006.

To achieve the goals and missions assigned to it, particularly in the main segment of business (maritime transport of goods), CNAN NORD Spa manages, in its quality of ship-owner and owner, a 05 types ship general cargo fleet, acquired as part of its development plan it is the following vessels:

- M/V SAOURA
- M/V STIDIA
- M/V SEDRATA
- M/V KHERRATA
- M/V CONSTANTINE

It employs, for this purpose, and full time 263 people including 167 sailors and 96 sedentary. All of this workforce are of Algerian nationality, the working languages used are:

- The French for all text messages.

- Dialectal Arabic and French for oral communications.

It is formed by the present between the owners of the shares currently created and those which may be subsequently created a company by shares, which will be governed by the ACC, supplemented and amended, as well as by the laws and regulations in force and the present articles [30].

### 1.1.2 Role of CNAN:

The company purpose is to:

- Maritime transport of goods in Eastern Mediterranean, Northern Europe, Americas, middle and far East and Inter-port,
- Brokering, transit and commission in customs,
- Ship chartering,
- Management and port warehouses under customs,
- In general, all operations legal, industrial, commercial or financial, securities and real estate that may relate directly or indirectly to the social or object which may facilitate the construction, extension or development [30].

# 2 Basic member:

## 2.1 Definitions:

**Container:**

A container is a rectangular metal box that is used to store items that must be transported from one place to another. Through standardization, the dimensions of the containers are regulated by the ISO 668:1995 standard. The unit of measurement of container is the equivalent to 20 feet (TEU), and there are containers of 40 feet (2 TEU).

Generally, there are several forms of container, of which the best known are: standard, high-cube, hard-top, open-top, flat-racks, plat, ventilated, insulated and refrigerated, bulk, and tank. **Figure 1.1** is an illustration of these various types of containers [27].


**Figure 1.1** Types of containers [27]

There are three kinds of containers:

- **General purpose containers**: also called dry containers. They are equipped with doors at the ends and intended for general and dry goods.
- **Containers for specific use:** in this category we find two types of containers, there are the open-top containers: the structure of this type of containers is identical to that of the dry, but the roof is mobile and is generally sheeted for

vertical potting (large pieces/indivisible), and there is also the platform container (Flats): they are open. There are two types of Flats; fixed ends-walled containers and other mobile end-walls. The Flats are the only two permits, under certain conditions, goods in excess of height/width.

- **Container for specific goods:** they are used for goods having a special thermal characteristic. We can find: the container vented; the surface of breakdown of this type of container is increased by the opening of ventilation holes in the side rails. It is used for the storage of certain fruits and vegetables, coffee in bags requiring traffic of air. And the temperature-controlled container equipped with a generator that can be connected to the electrical system of the wearer [23].

An intermodal container is the most significant technological development in freight transport during the last 50 years [11]. Intermodal containers exist in many types and a number of standardized sizes, the ISO specification for an intermodal container 2.438m high by 2.438m wide. The lengths of the containers do vary, but 5.8,12.2,14.6, and 16.15m are standard lengths used today [4], these containers can be used across different modes of transport – from ship to rail to truck – without unloading and reloading their cargo. Intermodal containers are primarily used to store and transport materials and products efficiently and securely in the global containerized intermodal freight transport system, but smaller numbers are in regional use as well [3].

**Containerization:**

Containerization or shipping containers is a system of intermodal freight transport using intermodal containers made of weathering steel. The containers have standardized dimensions. They can be loaded and unloaded, stacked, transported efficiently over long distances, and transferred from one mode of transport to another "container ships, rail transport flatcars, and semi-trailer trucks" without being opened. The handling system is completely mechanized so that all handling is done with cranes and special forklift trucks. All containers are numbered and tracked using computerized systems [8].

### 2.2 The general process of a container terminal:

It is described as a sequence of operations from the arrival of the container until the departure of the containers from the port or vice versa.

- The containers destined for export arrive at the port by truck, are distributed between blocks and stored in a storage area.

- After a certain period of time, containers are removed from the blocks with the porticos of Court (Yard skulls) and are transported by vehicles (Yard Truck) to the docks where they are collected by gantry Wharf (Quai skulls) and loaded on ships [24].

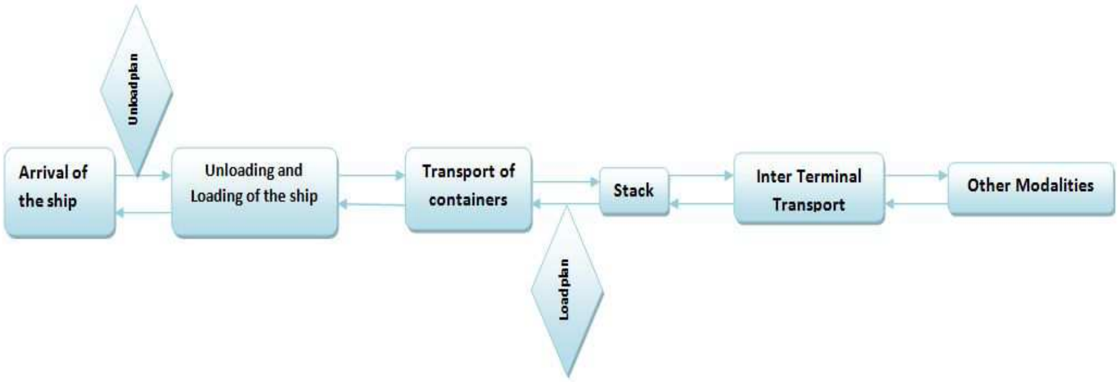



**Figure 1.2** The general process of a container terminal [38]

In the import situation, when a ship arrives at the terminal, imported containers must be unloaded by Dock Gate. Then they are placed on vehicles that will bring them to storage areas. After awhile, the containers leave storage areas and they will be transported by vehicles or other types of ground transportation (trains, lorries) [39].

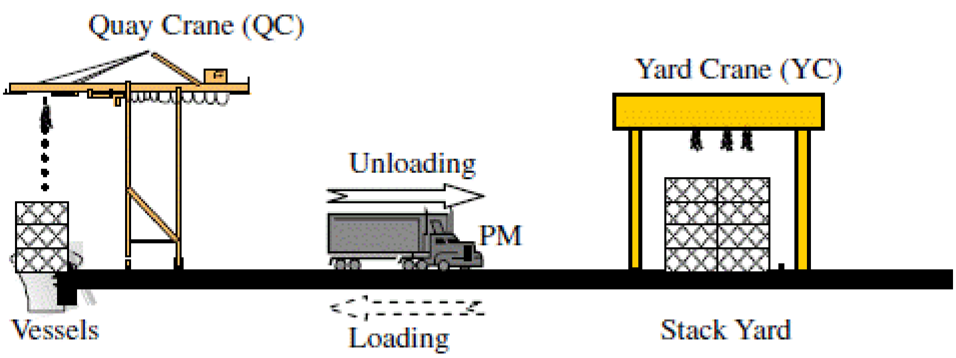



**Figure 1.3** Transportation and handling chain of a container [39]

2.2.1 The cycle of process of unloading of containers:

A *Lee and al.* (2009a) has been a description of cycle unloading of containers imported by trucks from the dock to the Court. The process of loading (unloading) of a container of its location in the storage area to the ship is referred to as a task. The time required to complete all tasks is called the completion time (makespan) [24].

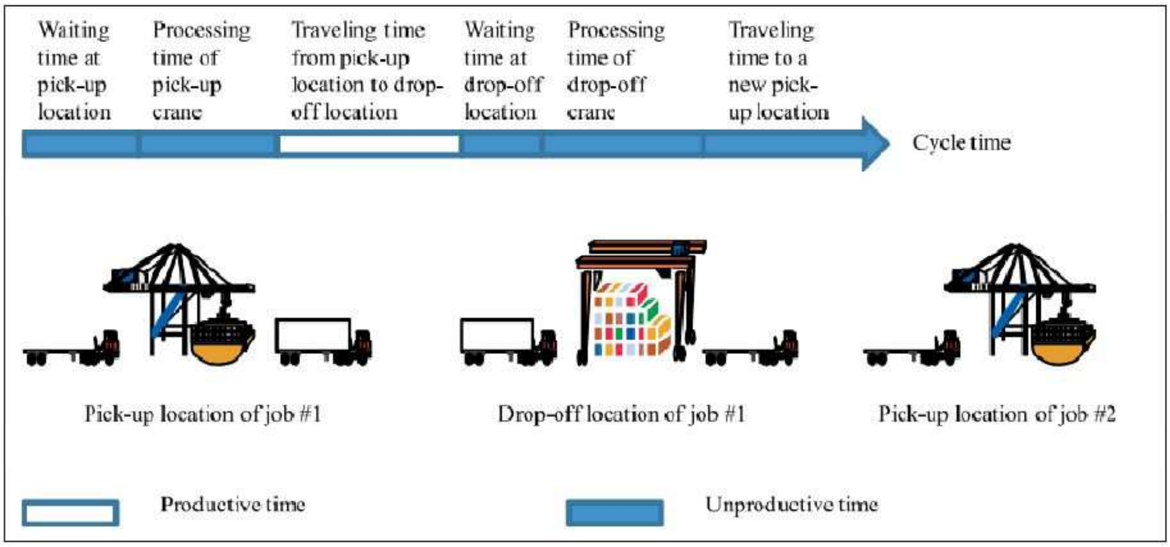



**Figure 1.4** Cycle of process of unloading of containers, *Lee and al.* (2009a) [24]

### 2.3 Container transport equipment:

Other essential equipment in a seaport is trucks. Beside the simple and ordinary trucks, there are several other forms of transport vehicles namely the AGV and the SC. The main function of these trucks is the link between the storage area and the docks by the transport of containers for export or import. The AGV require significant investments and have the ability to load a 40 container or well two 20 containers (Steenken and al., 2004). For the SC, the main characteristic is their ability to handle containers in the yard, in addition to their basic function which is the transport of containers from the storage area to docks or vice versa.

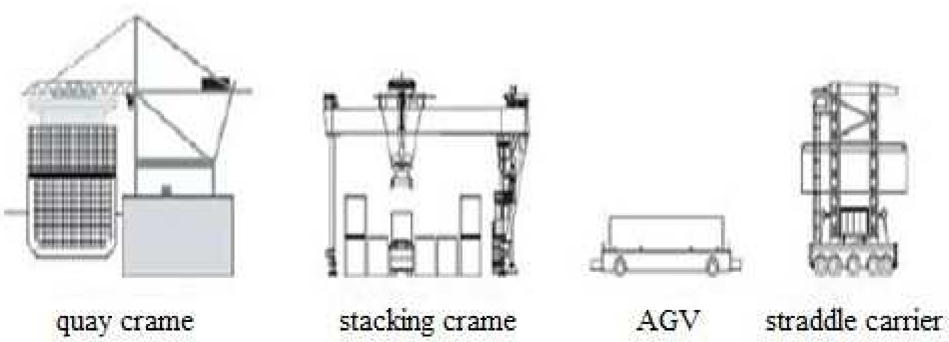



**Figure 1.5** Container transport equipment (Meersman, 2002) [24]

We can noted between the properties of these means of transport, they are very flexible and dynamic and they are led by "man driven" human except the AGV.

The SCs are capable of lifting containers to store them and that is why they are also called ALV [24].

2.3.1 The means of transport in a terminal:

**AGV:**

An automated guided vehicle or automatic guided vehicle is a mobile robot that follows markers or wires in the floor, or uses vision, magnets, or lasers for navigation. They are most often used in industrial applications to move materials around a manufacturing facility or warehouse. Application of the automatic guided vehicle has broadened during the late 20th century [39].

**Figure 1.6** AGV

**ALV:**

In automated container terminals, containers are transported from the marshalling yard to a ship and vice versa by automated vehicles. The automated vehicle type studied in this paper is an automated lifting vehicle that is capable of lifting a container from the ground by itself [39]. If a port uses AGVs, there exists significant unproductive and costly waiting both under the yard cranes and in the blocks.

A possible alternative solution to this problem is using ALVs, which can load and unload their own containers [37].


**Figure 1.7** ALV

**SC:**

A straddle carrier is a no road going vehicle for use in port terminals and intermodal yards used for stacking and moving ISO standard containers. Straddles pick and carry containers while straddling their load and connecting to the top lifting points via a container spreader. These machines have the ability to stack containers up to 4 high. These are capable of relatively low speeds ( up to 30 km/h ) with a laden container. The workers that use this machinery sit at the very top seated facing the middle as they can see behind them and in front of them. Straddle carriers can lift up to 60 t (59 long tons; 66 short tons) which equals up to 2 full containers [37].


**Figure 1.8** SC

**ASC:**

Automatic stacking cranes are breaking ground in all parts of the world. For medium-size as well as large terminals, even in areas with low labor cost, automation is often both economically and operationally the best alternative. Automated stacking cranes system enables the highest possible capacity and stacking density. The ASC terminal optimizes throughput and stack footprint. There are two types of ASCs: cantilever (side-loaded) cranes, where container transfer in and out of the stack is made alongside the gantry; and end-loaded ASCs [39]

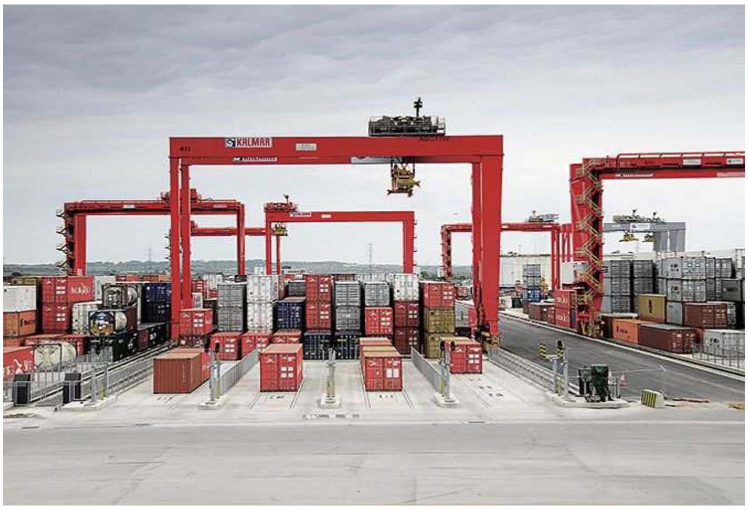


**Figure 1.9** ASC

**Gantry cranes:**

Gantry cranes are a type of crane built atop a gantry, which is a structure used to straddle an object or workspace. The terms gantry crane and overhead crane or bridge crane are often used interchangeably, as both types of crane straddle their workload. The usual distinction drawn between the two is that with gantry cranes, the entire structure (including gantry) is usually wheeled (often on rails). By contrast, the supporting structure of an overhead crane is fixed in location, often in the form of the walls or ceiling of a building, to which is attached a movable hoist running overhead along a rail or beam (which may itself move) [33].

Further confusing the issue is that gantry cranes may also incorporate a movable beam-mounted hoist in addition to the entire structure being wheeled, and some overhead cranes are suspended from a freestanding gantry. Crane is a machine for lifting, lowering, and moving a load with the hoisting mechanism a part of the machine [33]. The cranes can be:

• **Fixed Height Gantry Cranes** are available in spans up to 30 and capacities up to 5 ton.
• **Adjustable Height Gantry Cranes** provide the most flexible lifting solution of any style gantry crane. It can be easily moved and easily adjusted to provide different lifting heights.
• **Aluminum Gantry Cranes** are made from lightweight extruded aluminum track. They have an adjustable span and height and are collapsible for easy storage when not in use [35].

## 3 Container terminal:

A maritime container terminal is a place where containers arriving by ocean vessels are transferred to inland carriers, such as trucks, trains, or canal barges and vice verse.

Generally, a terminal is a facility where cargo containers are transshipped between different transport vehicles, for onward transportation.

The transshipment may be between ships and land vehicles, for example trains or trucks, in which case the terminal is described as a maritime terminal. Alternatively the transshipment may be between land vehicles, typically between train and truck, in which case the terminal is described as an inland terminal [32].

Maritime container terminals tend to be part of a larger port (**Figure 1.10)**, and the biggest terminals of this type can be found situated around major harbors. Inland terminals serving containers tend to be located in or near major cities, with good rail connections to maritime container terminals [28].

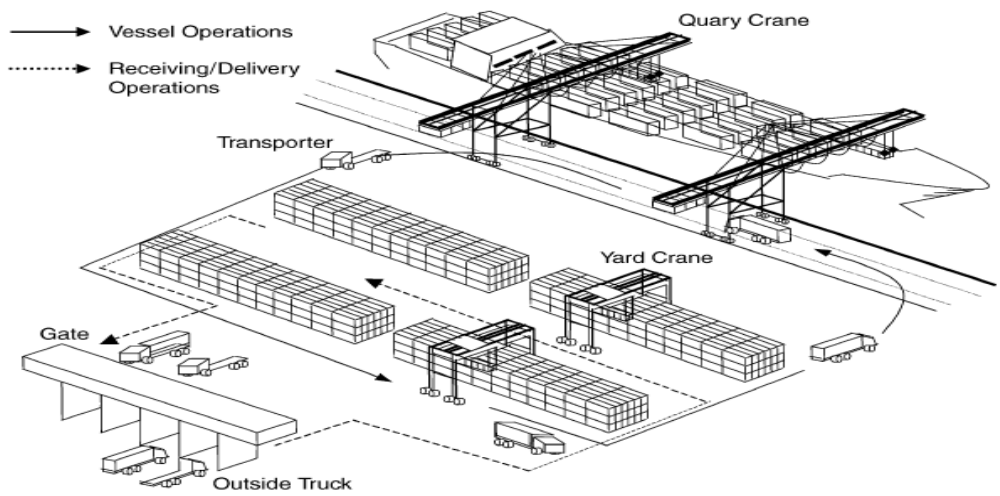



**Figure 1.10** A simplified three dimensional view of container terminal [28]

### 3.1 Functions:

Every maritime terminal performs four basic functions: receiving, storage, staging, and loading for both import (entering the terminal by sea and usually leaving by land modes) and export (usually entering the terminal by land and leaving by sea modes) containers [32].

- Receiving involves container arrival at the terminal, either as an import or export, recording its arrival, retrieving relevant logistics data and adding it to the current inventory.
- Storage is the function of placing the container in a known and recorded location so it may be retrieved when it is needed.
- Staging is the function of preparing a container to leave the terminal. In other words the containers that are to be exported are identified and organized so as to optimize the loading process. Import containers follow similar processes, although staging is not always performed. An exception is a group of containers leaving the terminal via rail.
- Finally, the loading function involves placing the correct container on the ship, truck, or other mode of transportation. In this work the emphasis will be put on internal logistics chain of container terminal (i.e. vessel-truck-yard and opposite direction respectively).

### 3.2 Structure:

In general logistics terms, container terminals can be described as open systems of material flow with two external interfaces. These interfaces are the quayside with loading and unloading of ships, and the landside where containers are loaded and unloaded on/off trucks and trains. Containers are stored in stacks thus facilitating the decoupling of quayside and landside operation [32].

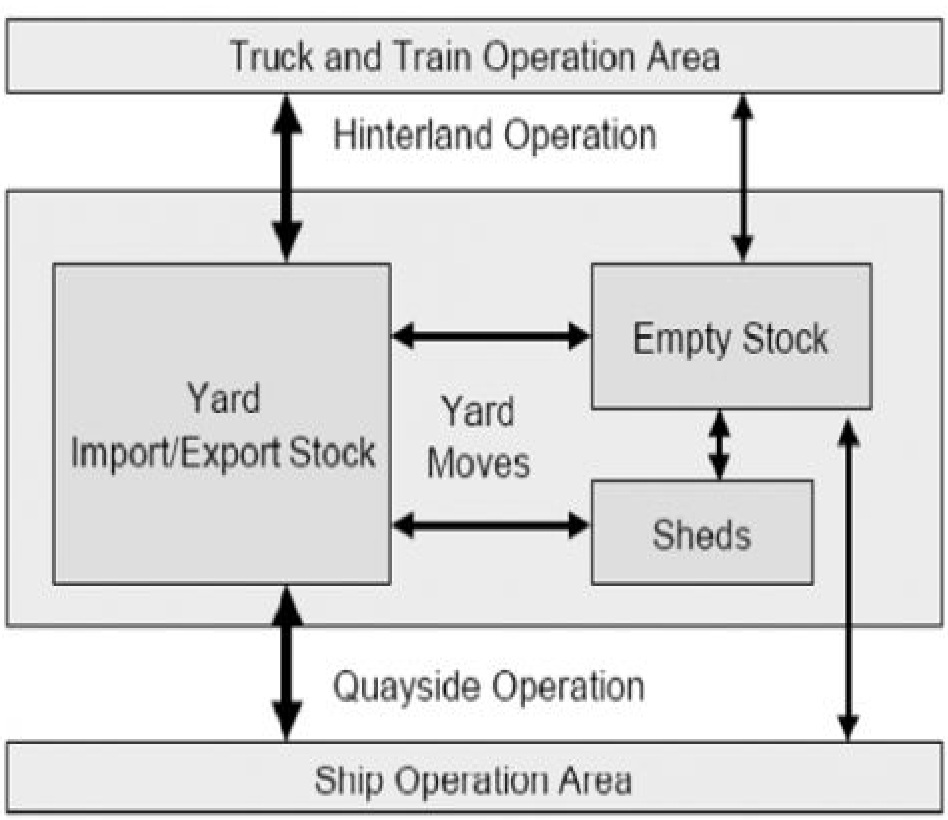



**Figure 1.11** Operation areas of a seaport container terminal and flow of transports [27]

### 3.3 Loading and discharging a ship:

When a ship arrives at the port, the containers have to be taken off the ship. This is done by manned QCs, which take the containers from the ship's hold and the deck.

Next, the QCs put the containers on vehicles, like AGVs. After receiving a container, the AGV moves to the stack. This stack consists of a number of lanes where containers can be stored for a certain period. These lanes are served by, for example, automatically controlled ASCs.

When an AGV arrives at a lane, the ASC takes the container off the AGV and stores it in the stack. After a certain period the containers are retrieved from the stack by the ASCs and transported by the AGVs to transportation modes such as barges, deep-sea ships, trucks or trains. This process is also being executed in reverse order, to load containers on a ship [32].

## 4 Containers storage (stacking):

The operation of storage is the most complicated part in a terminal since import and export containers are stacked simultaneously in the same storage space.

After the arrival of a ship, imported containers are unloaded and moved from the staging area to storage locations. The containers of the same length are normally stacked on each other, and are allowed to stay there for a few days for free.

The export process is fertilization that import. Before the arrival of the vessel, the port accepts containers destined for export.

Generally, their arrival is random. They are stored temporarily in a period which depends on the date of departure of the vessel. Before the loading of the containers in the boat, a loading plan is prepared taking into account the final destination, the type, the weight of each container and the maximum weight allowed for each cell, as well as the stability of the ship.

There are two ways of storage of containers; storage on chassis or stacking on ground. With the first system (chassis), each container is accessible directly and individual.

While the case of piling on ground access to a container, to unload all those who lie on him. Having regard to the limit of the space allocated to the storage, the use of the second system is the most common.

The storage area is a large space divided into blocks (**Figure 1.12**). Each block is composed of several bays each also includes a set of columns. The position of a container in a block is identified by the Bay, the column and floor [23].

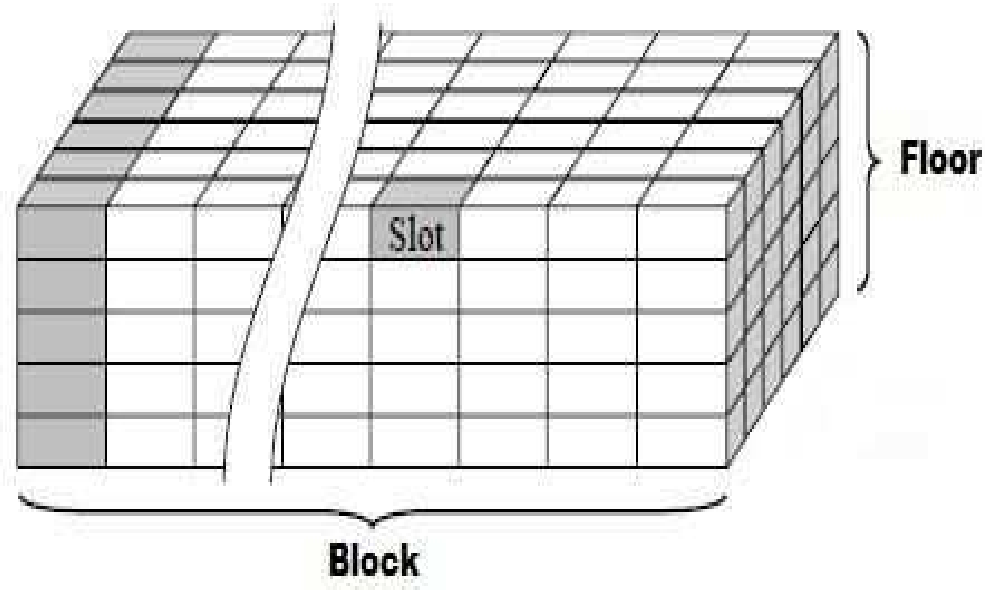



**Figure 1.12** Block [23]

It's the place where import and export containers can be stored for a certain period (**Figure 1.13**). It's divided into multiple blocks/lanes, each consisting of a number of rows. The height of stacking varies per terminal between two and eight containers high.

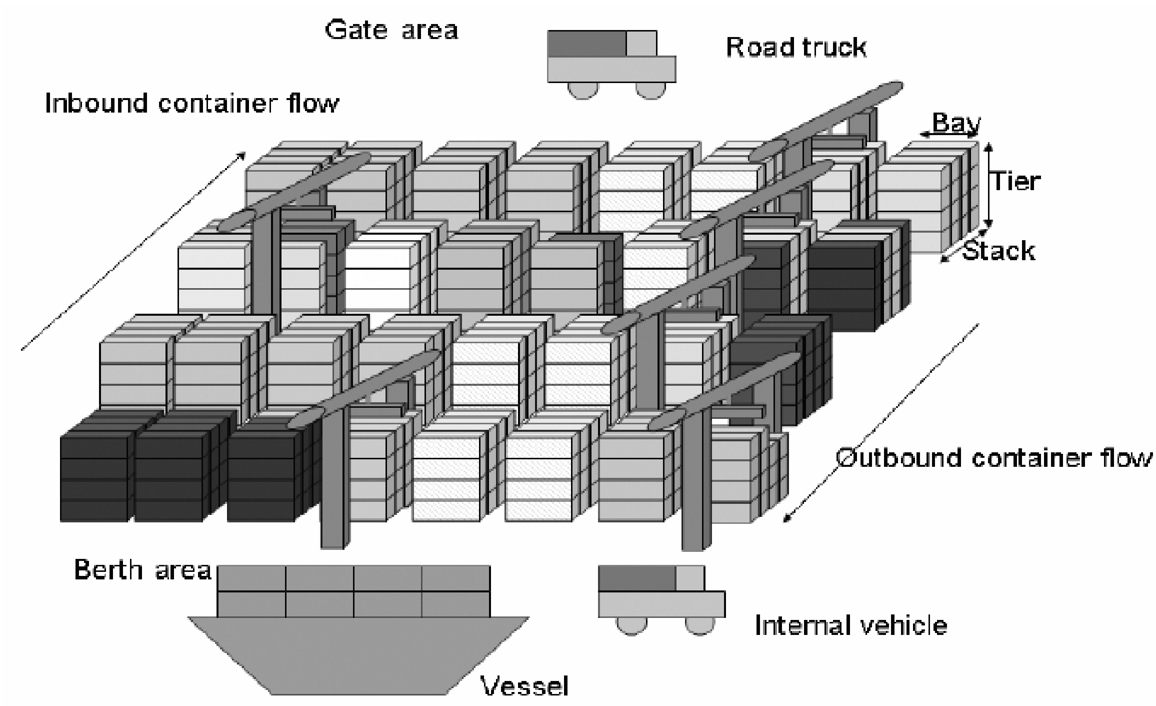



**Figure 1.13** The storage area [39]

At the end of each lane a transfer point might be situated. At this point the crane takes/places the container off/on the vehicle that transports the container. Empty containers are usually stored separately [39].

# CHAPTER 2

# STATE OF THE ART REVIEW

## Chapter Summary:

This chapter tells the history of containerization and its impact in the supply chain and globalization. This part allows us to get an idea about the different studies that are done already concerning the field of container terminals and it also allows us to position ourselves in relation to our problem and specified our orientations and choices that will serve as basis to our proposal. Then we present the CSP in which we speak about its complexity and the different methods which can solved it by the end we talk about CSS because it’s very important in the CSP.

---



---

# 1 The history of the container:

In May 2001, Malcolm P. McLean, the "Father of Containerization" used to say that he had the idea of rationalizing goods transport by avoiding the constant loading and unloading from one means of transport to another way back at the end of the 1930s at the port of Hoboken, when still operating as a small-scale hauler. To start with, McLean would load complete trucks onto ships, in order to transport them as close as possible to their destination. The development of standardized containers and trailers, moved by tractors, made it possible to ship just the trailers with the containers, so saving on space and costs. Later, the trailers were also left behind and the ships transported just the containers.

Ship-owners were more than a little skeptical about McLean's idea. This prompted him to become a ship owner himself and he appropriately named his company Sea-Land Inc. At the end of the 1990s, McLean sold his company to the Maersk shipping company, but his company name lives on in the name Maersk Sealand.

In the literature, the "Ideal X" is mentioned as the first container freighter. This ship left Newark on 26th April 1956 carrying fifty-eight containers, which it transported to Houston. The first ship designed to carry only containers is the "Maxton", a converted tanker, which could carry sixty containers as deck cargo. That was in 1956.

Another decade passed before the first container ship moored in Europe. The first container on German soil was set down by the "Fairland" at Bremer Überseehafen on 6th May 1966. The first containers used by SeaLand in Northern Europe were 35' ANSI containers, i.e. they were constructed to American standards. In other regions, 27' ANSI containers and other ANSI dimensions were often used. Ship owners in Europe and Japan quickly recognized the advantages of the container and also invested in the new transport technology.

Since American standards could only be applied with difficulty to conditions in Europe and other countries, an agreement was eventually reached with the Americans after painstaking negotiations.

The resulting ISO standards provided for lengths of 10', 20', 30' and 40'. The width was fixed at 8' and the height at 8' and 8' 6'. For land transport within Europe, agreement was reached on a 2.50 m wide inland container, which is mainly used in combined road/rail transport operations.


**Figure 2.1** The history of the container

The majority of containers used worldwide today comply with the ISO standard, with 20' and 40' long containers predominating. For some years, the ISO standard has come repeatedly under pressure. As stowage factors increase for most goods, many forwarders want longer, wider and higher containers, preferably all at once. Some ship-owners have given in to the pressure and containers of dimensions larger than provided for by the ISO standard are now encountered distinctly more frequently. "Jumbo" containers of 45' and 48' in length, widths of 8'6" (2.60 m) and heights of 9'6" (2.90 m) have been in existence for some years.

Efforts to build even larger containers, e.g. 24' (7.43 m) and 49' (14.40 m) boxes 2.60 m wide and 2.90 m high, are mostly confined to the USA. Even 53' long containers have been approved for use for some time throughout the USA, while some states will even allow 57'. In Europe and on other continents, narrower roads are a limiting factor.

Developing countries are understandably against changing the standards. More details are given in the section entitled "Container dimensions and weights" [31].

# 2 The studies concerning the field of container terminals:

Several studies were made in the field of the organization and resolution of CSP in container terminals, which related to the subject presented in this thesis, such as the problem of loading of the container operations planning which is formulated by joint linear program by the objective function that minimizes the time to completion of handling by the porticoes of court and based on several assumptions like this thesis [24]. There were also studies on the analysis of the PAD container handling operations, which affects two significant resignations: an operational dimension focusing on the productivity of the container terminal of Dakar, and size of container handling operations processes as in the thesis [26].

The majority of studies used the optimization algorithms for solving the CSP in container terminal, such as this thesis [27]. In [6] Kim and Park proposed a heuristic decision rule and a sub-gradient technical optimization to solve the storage space allocation for outbound containers, Kim [7] presented a technical to estimate the rehandlings number for the next pick-up and the total number of rehandles to pick up all inbound containers in a bay. Kim and Kim [5] proposed a cost model to estimate various cost components related to the import container handling and to determine subsequently the storage space and the number of transfer cranes required. Zhang and col. [13] solved the (SSAP) using a rolling-horizon approach. Both outbound and inbound containers are considered. There are various papers were proposed, treating different variants of the problem. I tried to present some of them in this section.

The most of studies was found in the field of container terminals they have used the application of agent which based on approaches to enhance container terminal operations, many dissertation are encompasses research addressing three critical decision making processes in marine terminal operations, the first, truck queuing at terminal gates, the second, inter-block scheduling of yard cranes and the third is the storage space allocation problem. All of these problems are concerned with resource optimization and share common objectives such as minimizing turn time of drayage trucks, reducing congestion and emission, and enhancing the productivity of terminals like dissertation [28]. The container storage space allocation is a critical decision in container terminals. It influences the productivity of the unloading process, either for inbound or outbound containers.

It's a complex operation since; it's highly inter-related with the routing of yard crane and truck [19], amongst the several problems of the CSP there are a lot of studies about the adaptation of the harmony search algorithm to solve the storage allocation problem for inbound and outbound containers like this thesis [15], which is studied the problem by considering multiple container type (tank, empty, refrigerated, open top, regular and open side), the variety of container type round the situation more complicated than others operations of containers storage. This [18] of a multi-agent system for the automation of a port container terminal presents a system architecture which is based on the multi-agent system paradigm for solving complex problems. This architecture is applied to solve the port container terminal management problem, and specifically to solve the automatic container allocation. The multi-agent systems paradigm seems to fit this problem due to its inherent complexity.

## 2.1 Why we need to see studies have done?

For well started in this thesis it's important to see the studies that were made previously in the field of port container terminals, and to make an overview of similar problems, and resolution methods that they are dedicated. For the purpose of a good understanding the field and to take a general idea about the different problems which arise as well the resolutions which were already proposed for them. This allows emphasizing the complexity, dynamics and openness characterizing the problem. The reasons to see the studies passed allow to position compared to our problem and specify our directions and our choices which will provide the foundations for our proposal.

The resolutions of the various problems in this field are based much more on heuristic approaches such as adaptive research in wide vicinity (ALNS). Its solve optimization problems in a terminal container [24], the effectiveness of the ALNS method is represented in the possibility of solving the problem regardless of its size.

The data are fictitious and many instances are built by varying the number of containers and the number of equipments in each time. The resolution of a series of optimization problems serves to improve the efficiency of port operations like the transport trucks travel planning, the porticos of court operations planning, the porticos of Wharf operations planning and the allocation of containers in storage areas. The containers are stored in the storage area which is separated into blocks.

A block usually consists of 6 lines (rows). Each line consists of 20 bays or more that can reach 4 to 5 height containers (floors) like the block of containers storage in **Figure 2.2** [24].

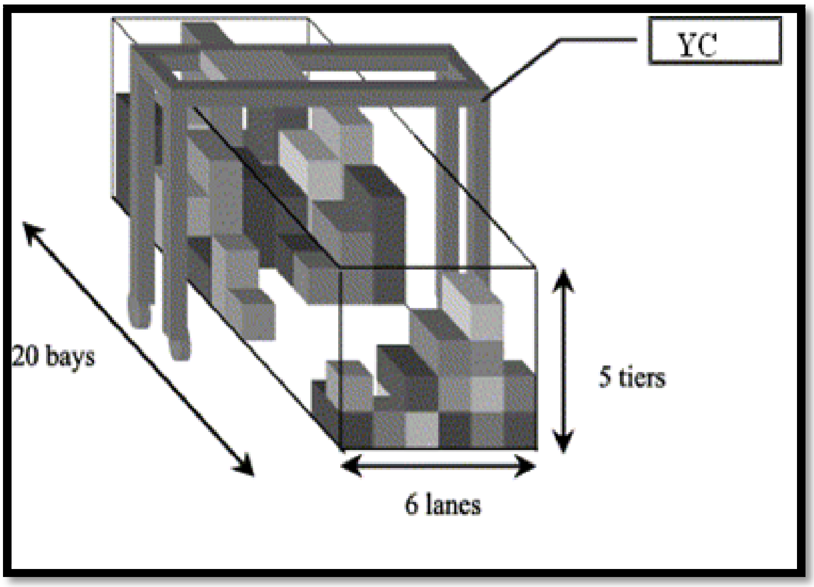



**Figure 2.2** Block of containers storage [24]

The containers are stored in stacks consisting of several called levels still floors. The position of a container in the storage area is characterized by a specific address block, Bay, row, floor. Maximum number of floors depends on released in the terminal handling equipment (*Steenken and al.*, 2004).

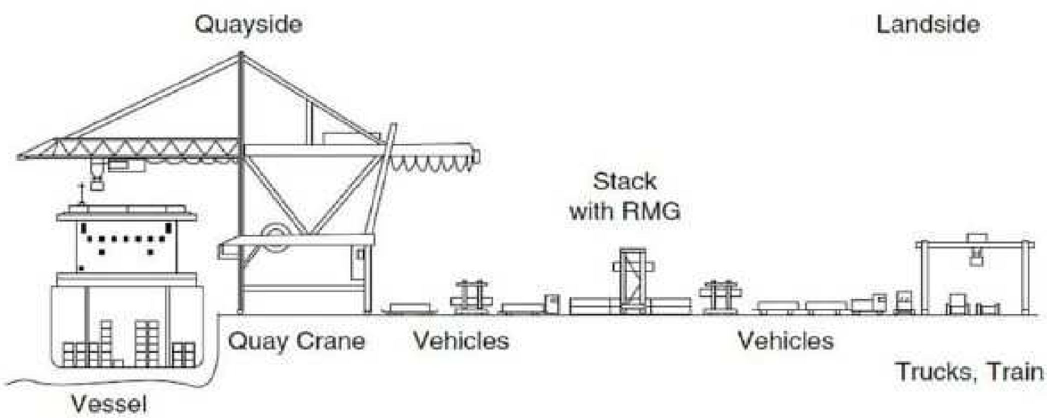



**Figure 2.3** Container terminal (CT) schematic side view (Steenken and al., 2004) [24]

### 2.2 The results and goals achieved in the previous studies:

Seaports have experienced major developments of the handling techniques with the evolution of the phenomenon of containerization, as there was a lot of work about the operations of the port facilities planning; namely the porticos of court and transport vehicles while trying to find out how the authors analyzed the operations relating to these devices and on what are they based to solve the problems arising there from. The general objective of their work is to plan handling of containers by the porticos of court. They are also interested in the synchronization of concurrent handling operations by the porticos of court and trucks taking into account the possibility of interference between the porticoes of court and non-productive movements as the thesis [24], where in which the problem is formulated by joint linear program then it is solved using a heuristic method (ALNS) in order to minimize the time of completion of container handling operations for export. Computational tests are made to evaluate the efficiency of the developed algorithm (ALNS) for that we used multiple strategies where we made different combinations of removal and insertion heuristics.

Each research has some aims to arrive to them, like [6] the objective was to find an arrangement of the containers that exploits efficiently the storage space and loading operations, [13] its aim was to minimize the total transportation distance of containers between blocks and vessel berthing locations. The objective of this thesis [15] is to find an optimal container arrangement which respects their departure dates, and minimize the re-handle operations of containers.

Finally, it performs tests on different instances to validate this method because only the results of tests that show us the quality of the solutions generated by a method.

## 3 Container storage problem (CSP):

The CSP in container terminal consists to effectively manage the storage space so as to increase the productivity of port, the most container storage problems especially with the considerable growth of the number of containers through seaports and limitation of storage areas in these ports. When a ship arrives, the inbound containers are unloaded by QC and then placed on quays. So, they are collected by SC. Each is able to store it in its storage location.

In order to find an optimal storage plan, the CSP falls into the category of NP hard and NP complete problems, it consists on finding the most suitable storage location for incoming containers that minimizes remanding operations of containers during their transfer to the ship, truck or train. In fact, the wait time of customer trucks, the transfer time of yard crane and the Ship turnaround time are advantageously reduced [20].

**3.1 The complexity of the problem:**

For studies it complexity of the CSP, the ideal case is considered, in which the Court of storage is large enough to allow the prohibition of metrication. In addition to this, the mono-objective version of the problem is considered also, in which only the total distance from the storage locations and the outputs is minimized.

As a first step, an instance any initially all of CSP, the stacks are empty, after that the case where all the stacks of court storage are not empty. The CSP can be defined as a problem of allocation of containers of different types and sizes arriving at a port, to empty locations within blocks of storage in the port, and it can be formulated as a problem of Bin-Packing 3-d where containers represent the objects and areas of storage boxes used for storing. It belongs to the category of problems NP-hard and NP-complete.

**3.2 The methods have used in CSP :**

The containers storage problem represents one of the main problems especially with the considerable growth of the number of containers through seaports and limitation of storage areas in these ports. The objective of solving problem of storage containers in container terminal is to find an optimal storage plan that specifies the location of ideal storage for each container and taking into account the actual storage constraints, for that the researchers have used several method like An exact method which is an approach that allows to generate the optimal solution to a problem.

Its principle generally consists of an exhaustive list of all possible solutions. As such, it will be difficult, even impossible, to find the optimal solution if the problem size becomes large. In addition, these technicals are very slow and require a very high execution time. However, there are some smarter methods which allow solving problems of more or less large sizes, minimizing the number of feasible solutions. They include the Branch and Bound, Branch and Cut algorithm or dynamic programming [23], where are also the colony algorithm and bees algorithm for CSP; in the case of a container terminal that is not a court of storage large enough to allow the operations of storage and retrieval for this colony of bees which was proposed for algorithm simultaneously minimizes the number of redesigns, distances to be travelled, and dispersions of containers that belong to the same category. It is a meta-heuristic algorithm which is applicable to various optimization problems. It was invented by Pham and al. who published in 2005 “The bees algorithm a novel tool for complex optimization problems” [7].

Where are also the genetic algorithms which are proposed for the resolution of the containers storage problem of includes mainly four steps which are: the creation of an initial population, the selection of a portion of this population, the application of crosses between selected individuals, and the mutation of an individual. We found an algorithm which is a combination of genetic algorithm and Ant Colony algorithm it called HACG Therefore, it contains all the procedures of these two algorithms. However, its main difference with the genetic algorithm is the fact that a part of the initial population is built by ants while the other part is built in the same way as with the genetic algorithm. The HACG algorithm begins by couples creating and initializing trace of pheromone. After that, each ant built a solution. There are thus a number of individuals added alternatives for an initial population to which one applies genetic algorithm. Subsequently, it updates the footsteps of pheromone, and then it starts a new iteration. On the other hand there is an Algorithm of SA is a meta-heuristic algorithm, which was founded in 1983 by Kirkpatrick [21]. They were inspired of a metallurgical process that aims to improve the appearance of a metal by heating to a high temperature and then cooling it gradually [4].

The simulated annealing algorithm is part of the class of local search algorithms; he explored the vicinity of an initial solution to find better solutions.

**3.3 Container Storage System (CSS):**

A container storage system is provided comprising a storage area, the storage area having a longitudinal axis with a shorter transverse axis. An overhead crane is mounted on the support framework, and at least one vehicle roadway extends substantially parallel to the transverse axis of the storage area. The overhead crane is moveable on the support framework to remove containers from a vehicle on the roadway and deposit the containers for storage in the storage area, and to subsequently replace containers on a vehicle after storage. The invention substantially reduces the amount of space taken up by roadways compared to known systems, in which the roadways extend longitudinally through storage areas [1].

A storage container is defined as a specific location in high-volume storage. It resembles a folder in a computer's file system; although there are some differences in the way files are handled. The allocation of these containers on the yard is a problem that directly affects the previous two systems. A bad container distribution forces the transtainer to make more movements and the GCs to be inactive more time, which increases the loading time.

A storage container can exist only at the highest level; it cannot have "sub containers" as a computer folder can have subfolders. A container can’t be deleted without removing all the files in it first. In addition, a container can’t be directly renamed. In order to change the name of a container, first all of its files must be removed, the empty container must be deleted, a new container must be created with the new name, and finally the files must be uploaded into the new container. The CSS is the way to reduce the useless transtainer movements is to increase the stacking density. Then, all the containers are allocated in close areas and the time dedicated to the movements is reduced.

A transtainer is a hoisting device for loading or unloading containers onto or from railway wagons. It is a gantry crane that usually travels on rails and is set up across a track bundle at a container terminal. The transtainer moves the containers alongside the track bundle, after which straddle carriers transport them to the storage area. Transtainers or yards gantry are sometimes also used in the storage areas of container terminals in order to stack containers in large blocks. In that case, the transtainer will have pneumatic tyres (**Figure 2.4**) [18].

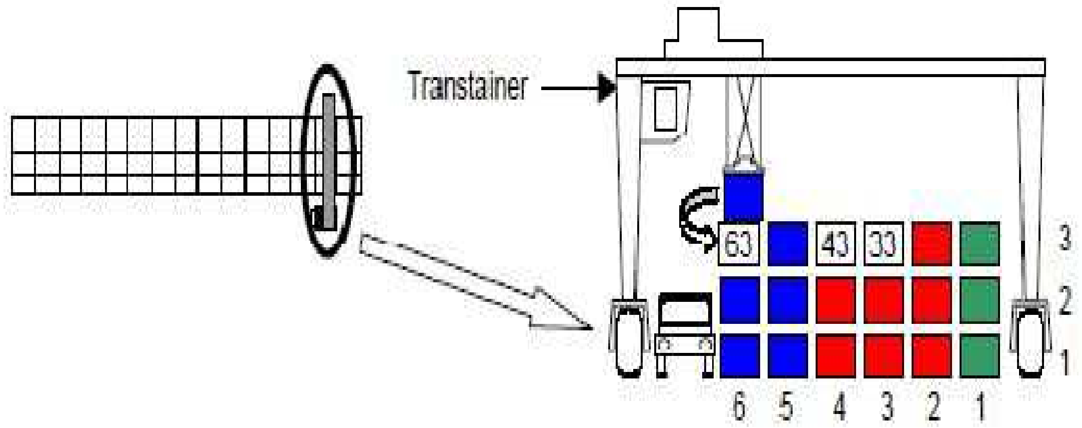



**Figure 2.4** Transtainer [18]

They are similar to ship-to-shore cranes in that, they attach their spreaders to containers for their lifting and lowering. Transtainers may lift and lower containers from and to rail cars, truck chassis, and container stacks and may be rail or rubber tired. If rubber tired, they are referred to as rubber-tired gantry cranes (RTGs). RTGs are more mobile than rail-tired gantry cranes in moving from one container storage bay to another. However, the greater stability of the rail-tired gantry crane allows ports to use them to stack containers higher, thereby increasing the density of container stacks in storage areas.

Like straddle carriers, transtainers lift and lower containers from overhead. Unlike straddle carriers, transtainers have the disadvantage of not being able to transport containers from one designated area of the port to another in a timely fashion; the transportation of transtainer containers is usually performed for their transportation.

Transtainers, however, have a selectivity-capability advantage over straddle carriers with respect to container stacks, i.e., they can generally pick and remove or relocate containers in the stacks faster than straddle carriers. These advantages increase with the value of port land. Also, transtainers can have an advantage over straddle carriers with respect to how high containers can be stacked. Usually, straddle carriers only stack containers three high (Bielli and al. 2006). Furthermore, transtainers require less space of stacked-container bays than straddle carriers and top loaders, i.e., transtainers allow for greater density in stacked-container storage than straddle carriers and top loaders (Bielli and al. 2006) [12].

The container allocation problem focuses on assigning a free allocation on the yard for the container to be stored until it is loaded into a ship (or, if it comes from a vessel, until a truck takes it away from the terminal). When a container arrives to the terminal, the system decides which allocation is the most appropiate depending on the assigned cargo ship, its destination port, its dimensions, its weight and, if it is available, the allocation into the bay of the ship [18].

# CHAPTER 3

# THE METHOD PROPOSED FOR SOLVING THE OPTIMIZATION PROBLEM

## Chapter Summary:

This chapter is devoted to the description of problem. In the second part we have mathematical model in which we have presented a mathematical formulation proposed for CSP. In the third part we have made a description of the Branch and Cut algorithm for solving the CSP, and we have justified the choice of Branch and Cut algorithm and we have described it, also its principle. By the end we have applied the Branch and Cut algorithm on the CSP in witch we have proposed an algorithm for our problem with a simple discretion for each function that has used in the main algorithm.

---



---

# 1 Description of problem:

The CSP has become one of the most difficult problems of storage in our day, especially with the increase in the number of incoming and outgoing containers as well as the storage space is limited.

The problem posed in this thesis is how we can organize containers in the operation of storage in such a way it optimizes the number of containers stores in a limited storage area. The best organization of storage saves more space, and that allow us to store much more containers.

Both cases are distinguishable in the operation of storage of containers: static and dynamic. The main difference between these two cases is the fact that in the static case, it considers that all containers have already arrived at the port before the start of storage operations; while in the dynamic case, we take into consideration the containers that will arrive after the beginning of storage operations.

In this thesis, we treat the static case of the containers storage problem, and we consider a terminal container, where containers come by ships. We take into account both the incoming containers than the outgoing these containers will go through the dock when they were unloading from ship, and between these containers some of them they will transport by camion, train,... etc, and the rest they will store in storage area.

To solve optimization problems, we distinguish two types of problems:

- Decision problem is a problem with two possible answers: yes or no.
- Optimization problem is detected a solution minimizing a certain objective function.

We have two families of approaches can be used; the exact methods and the meta-heuristics. The advantage of an exact approach is that it provides the optimal result for the problem. Among these methods, include separation technicals and evaluation algorithms with return back and other resolution methods exist to solve the containers storage problem. To do this, we adopt the combinatorial optimization (Annex A) in which we give a mathematical modeling and study the complexity of the problem and a numerical approach; by proposing a resolution by Branch and Cut algorithm.

## 2 Mathematical models:

For the resolution of the static case of the CSP, we propose a mathematical formulation of the problem in the form of an integer linear program (ILP) in which the parameters described in **table 3.1** are used [27].

| **Containers :** | |
|---|---|
| $C$ | : Set of containers |
| $q$ | : Index of the container |
| $N$ | : Number of containers |
| $Lq$ | : Location of $q$ in stack (number of $p$ where we find $q$ ) |
| $Ip_q$ | :Position of $q$ in stack (floor) |
| $T_q$ | : The type of container |
| **Stacks :** | |
| $p$ | : Stack |
| $Np$ | : Number of stacks |
| $C_P$ | : Number of free slots in stack |
| $t_P$ | : Type of container which can be placed in the stack $p$ |
| **Others :** | |
| $m_p^q$ | : Distance between containers and stacks which have the same type. |
| $x_p^q$ | : The decision variable |

**Table 3.1** Settings table

Resolution methods already proposed before, in these four following assumptions that they are considered; it takes into account the differences in size between the containers and are stored in each cell only containers of the same size, containers are of the same class if they have the same dimensions and are for a single train, or a same vessel, or even if they belong to the same customer in case containers travelling by road, it is assumed that the containers are numbered according to their order of arrival and unloading, since plans for unloading ships and trains are known in advance, considering that the stacks are numbered so that two adjacent cells that are in a same span have successive numbers, and if two spans are adjacent then the number of the last stack one succeeds the first stack of another number [25].

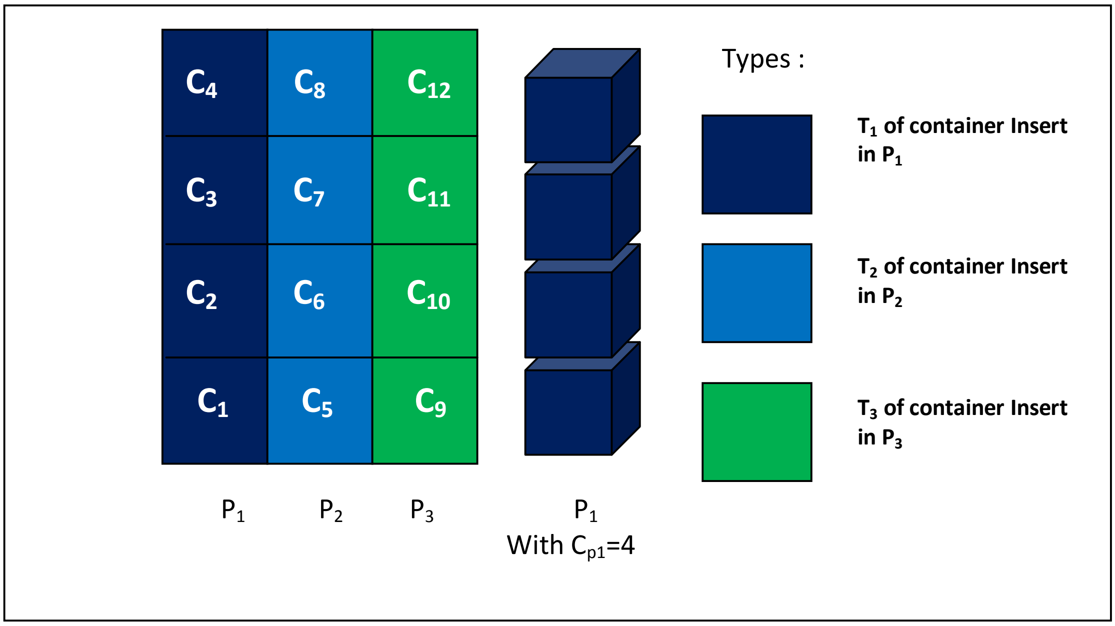



**Figure 3.1** Stacks

**Assumptions:**

To solve this problem, a number of assumptions must be considered:

- It takes into account all containers that have the same type $T_q$=$T_{q'}$, the same shape and the same weight, such as identical containers are classified in the same stack.
- A container can't be stored directly when it arrived from the ship, we put it in dock first then we store it in storage area.
- Storage blocks are initially empty.
- All containers will be stored in blocks of three dimensions.

The decision variables are defined as follows:

$$x_p^q = \begin{cases} 1 & \text{If container } q \text{ is assigned to stack } p \\ 0 & \text{Otherwise} \end{cases}$$

The parameters used in this model are described before in **table 3.1**. We can notice that the decision variable $x_p^q$ accurate the stack which is assigned each container, but it doesn't specify the exact location of a container in a stack. This poses no problem, because the way the graph is constructed ensures that the arrival and departure orders containers that are assigned to the same cell are compatible. The grouping of containers by category $T_q$ is essential when we assigned $q$ to stack $p$. The organization of storage containers solved by allocating to the stacks in which we respect that the type of container assigned is the same of stack $T_q = t_p$ , it's also essential to adopt a storage strategy (ANNEX B), so it stays to us just the minimization of the distance $m_p^q$ between containers and stacks which have the same type, therefore we consider the single objective version of the container storage problem. The following mathematical model was proposed:

$$Minimize\ f(m, x) = \sum_{q=1}^{N} \sum_{p=1}^{Np} m_p^q x_p^q \qquad (1)$$

The objective function (1) minimizes the total distance between containers and stacks to optimize the storage plan.

$$\sum_{p=1}^{Np} x_p^q = 1\ \forall\ q = 1, \dots, N\ \ and\ p = 1, \dots, Np \qquad (2)$$

Constraint (2) requires that each container is assigned to a single stack.

$$x_p^q + x_p^{q'} \leq 1, \forall (q, q') \in C, p = 1, \ldots, Np \qquad (3)$$

Constraint (3) ensures that incompatible containers (q and q′ which have different types) does not assign to a same stack.

$$\sum_{q=1}^{N} x_p^q \leq c_p \ , \forall\, p = 1, \ldots, Np \qquad (4)$$

Constraint (4) assures that the number of containers assigned to each stack is less than number of free slots.

$$\sum_{T_q \neq t_p} x_p^q = 0, \forall p = 1, \ldots, Np \;\; and \;\; 1 \leq q \leq N \qquad (5)$$

Constraint (5) secures the compatibility between containers and stacks.

$$x_p^q \in \{0,1\}\ \forall p = 1, \ldots, N_p, q = 1, \ldots, N \qquad (6)$$

Constraint (6) specifies that the decision variables are Boolean.

$$\sum_{q=1}^{N} \sum_{p=1}^{Np} p - Lq \qquad \text{, Where } T_q = t_p \qquad (7)$$

Function (7) calculated distance between location of container and stacks which have the same type with it.

**In case of import:** When a container arrives we search in which block we make on it when we find this block we search which stack from this block we can make our container.

**In case of export:** when we want to export a container we search in which position is it in, when we find it we unstuck the container which is in the top of stack because it has a priority more than the others containers $Ip_{q'} > Ip_q$.

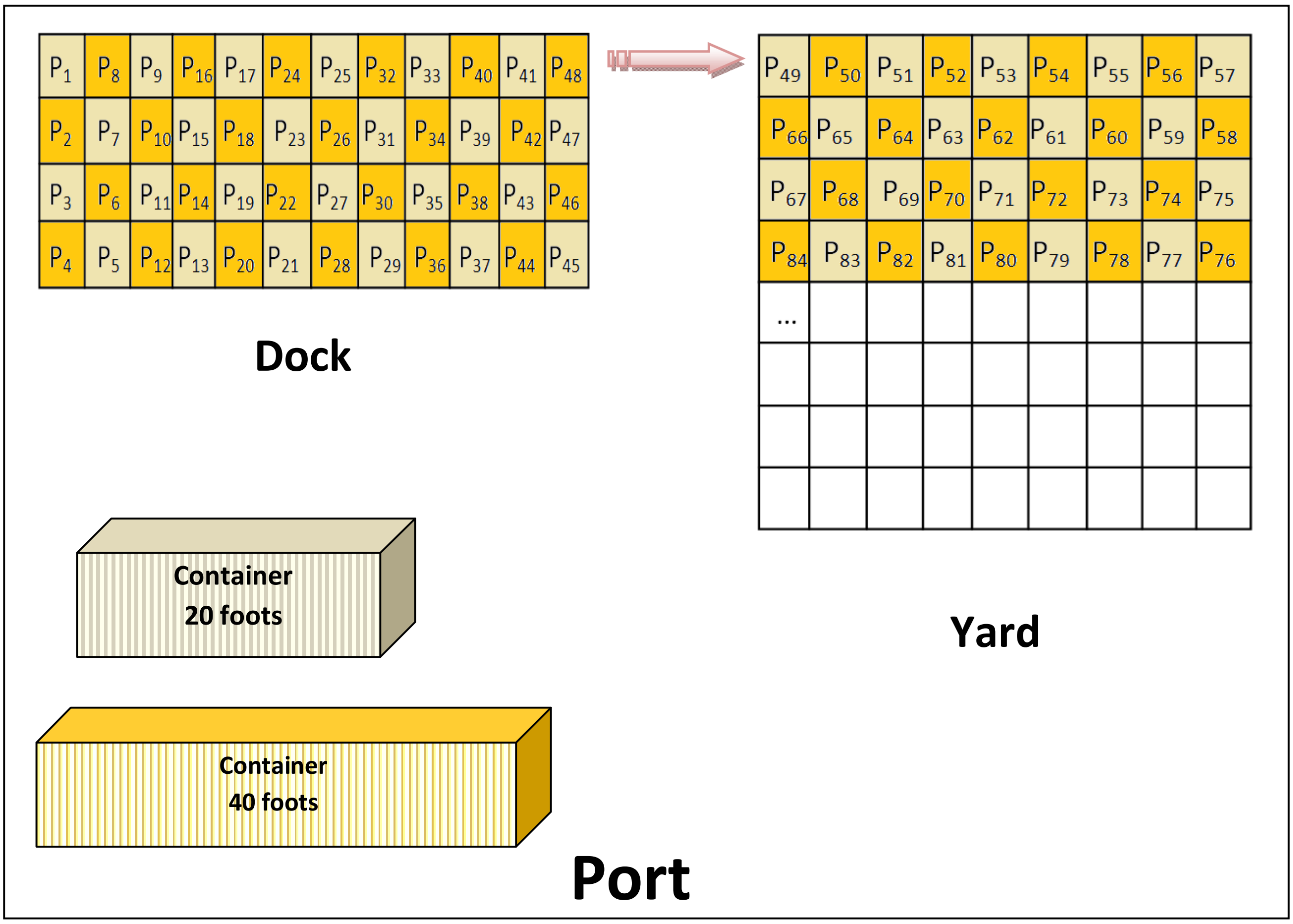



**Figure 3.2** Scheme summarizes the mathematical model of CSP

### 2.1 Complexity of problem solving:

To study the complexity of the container storage problem, we use graph theory concepts. Specifically we reduce the problem to a graph coloring problem.

We have the following preliminary notions in which we used the incompatibility graph G (V, E) : a graph, were V is the set of vertices and E the set of edges. Every vertex represents a container, and |V|=N.

An incompatibility graph is a graph in which each edge connects two incompatible peaks for a given relationship.

There is an edge between two vertices $v_q$ and $v_{q'}$ if and only if $T_q \neq T_{q'}$ , this means that container q and q′ can't be assigned to a same stack.

To study this problem we consider the ideal case in which the storage yard is large enough for the Prohibition of alterations possible. On top of that, we consider the single objective version of the problem, in which only the total distance between storage and stacks locations is minimized. We consider that in any instance of the container stacking problem (CSP), stacks of storage yard are not initially empty.

In the case where all stacks are not initially empty, the CSP is equivalent to a coloring problem with capacity, which generalizes the bounded coloring problem (so that is NP-hard).

In the coloration with capacity problem a capacity $C_p$ is fixed for each color p, and requires that each color is used at the $C_p$ once in a solution. Capacity $C_p$ can have different values. For an instance of CSP where each stack *p* has a capacity $C_p$, the CSP becomes a coloring problem with capacity in the graph G (Q,T,R) where each color can be used at most $C_p$ time.

## 3 Branch and Cut algorithm for solving the CSP:

In case the storage area is large enough for storage without any possible reshuffle, we propose a Branch and Cut algorithm for solving the container storage problem. To do this, we use the incompatibility graph, where vertices represent containers, and each edge connects two containers with different sizes or whose arrival and departure orders are not compatible. Since we know that two adjacent containers in the graph can't be stored together in the same stack, as we have already exploited this property in the mathematical model [27]. In the next section, we will explain why we have chosen the

Branch and Cut algorithm and we present this algorithm and its principle, then we describe the algorithm according to CSP.

### 3.1 Branch and Cut:

Branch and Cut is an exact method of combinatorial optimization for solving integer linear programs (ILPs), that is, linear programming (LP) problems where some or all the unknowns are restricted to integer values [17]. Branch and Cut involves running a Branch and Bound algorithm and using cutting planes to tighten the linear programming relaxations [16].

The Branch and Cut is a Branch and Bound with dynamic stress generation (often called cuts because they are used to break the current fractional points). We can be seen as the dual version of the branch and price but it is much easier to implement because the conventional binary connections on variables pose no problem under a Branch and Cut or the new dual variables the addition of new constraints.

There are two primary use cases of the Branch and Cut:

1. There is a MILP formulation and we want to use to avoid cuts maximum plug (it being very expensive) improving the relaxation value.
2. There is an ILP formulation with a very large number of constraints and we don't want (or we can't) enumerate them all [14].

### 3.2 Justification of choice:

There are many methods to solve the container storage problem, like Branch and Bound which is reliable but slow where is also Gomory Cutting Planes which is faster than Branch and Bound, but unreliable. So we have chosen the Branch and Cut because it combines the advantages from these two methods and improve the defects. It has proven to be a very successful approach for solving a wide variety of integer programming problems.

We can solve the MILP by taking some cutting planes before apply whole system to the Branch and Bound, Branch and Cut is not only reliable, but faster than Branch and Bound alone. Finally, we understand that using Branch and Cut is more efficient than using Branch and Bound.

### 3.3 The principle of the Branch and Cut algorithm:

Branch and Cut is also called programming method integer. Like any implicit enumerative method, the algorithm builds a tree structure named the shaft of "Branch and Cut", sub-problems that form the tree are called nodes.

There are three types of nodes in the tree of "Branch and Cut":

1- The current node that is being treated
2- The active nodes that are in the waiting list problems
3- The inactive nodes that were pruned during the progress of the algorithm.

The principle is to qualify from a full solution of the problem, and with the aid of the simplex for example, to go to another whole feasible solution to the optimum [23].

## 4 Description of the algorithm:

The Branch and Cut is an exact solution method that combines the Branch and Bound method and the cutting plane. Each of these methods works by solving a sequence of relaxations of the problem initial integers. The plan cutting methods improve relaxation of the problem in whole numbers, to find a better approximation, while the algorithms of Branch and Bound use sophisticated approach called divide for conquer.

The Branch and Cut algorithm uses a search tree whose root is a relaxation of the mixed integer problem solving. The other nodes of the search tree are created by partition of the solution search space, i.e. making connections. The major difference between the Branch and Cut and the Branch and Bound is that the first method uses valid inequalities to improve the solution found in each node of the search tree, before creating new branches. A valid inequality is an inequality which must be verified by every solution of mixed integer problem, but that is not necessarily satisfied by the solutions of relaxed sub problems.

To perform the Branch and Cut algorithm, it is crucial to have: one or more valid inequalities, a relaxation method of the studied problem, a technical of finding an upper bound, a branch rule, and valid inequalities research method are not met (also called separation method). We will, at first, detailing the last, before describing the sequence of steps of our Branch and Cut algorithm [27].

**Valid inequalities:**

For the creation of a valid inequality, by adding N($q$) the set of $q$ neighbors to constraint (3), we obtain the valid inequality (8), which is equivalent to (3) because for each $q'$ from N($q$) the $x_p^{q'}$ can equal to 0 or 1.

$$\sum_{q' \in N(q)}^{N} x_p^{q'} + |N(q)| x_p^q \leq |N(q)|, \forall (q, q') \in C, q = 1, \ldots, N; and\ p = 1, \ldots, Np \qquad (8)$$

$$1 \leq q \leq N$$
$$1 \leq p \leq Np$$

**Relaxation method of the studied problem:**

We apply the linear relaxation for the formulation of the problem to perform the Branch and Cut algorithm, so after the relaxation of integrity constraint (6), we find that the total number of constraints of the mathematical model remains great [20], we remove the adjacency constraint (3) and also eliminating the constraints of integrity of the decision variables from the model knowing, to decrease the number of constraints of the mathematical model. This does not affect the optimality of the solutions provided by our Branch and Cut algorithm, because the valid inequalities of the form (8) are used, each node of the search tree, to ensure the accuracy of solutions [27].

**Technique of finding an upper bound:**

The solution of the relaxed model gives us a lower bound of the solution of ILP. To find an upper bound, we solve the bounded problem on the incompatibility graph, so we search for each vertex number of its neighbors, the vertex which has the max neighbors it chooses among the admissible stack (i.e. that isn't attributed to the neighboring peaks of vertex, which have the correct size) and which is the nearest stack. The rest of vertices are assigned to the furthest stack which is compatible to them. Whenever a vertex is attributed, the number of empty slot of the stack corresponding to the used stack is reduced, for that, whenever there are updates that are performed on the capacity of the stacks as the algorithm progresses.

**Branch rule:**

When the solution provided by a node of the search tree is not full, we can improve it by making a connection for that we choose a most fractional decision variable i.e. the one which is halfway between the lower integer part and the upper integer part. For make a connection we use the classical branching rule, at each node of the search tree we create two branches (two sub-problems). Let $x_p^q$ is the largest fractional variable, when we put $x_p^q = 0$ in the first branch; it means that vertex (container $q$) will not be assigned to stack $p$ (in this branch). When, in the second branch, we put $x_p^q = 1$ which means that vertex (container $q$) will be strongly affected to stack p in this branch.

**Separation method:**

A separation method is used to detect valid inequalities which are not verified by a solution of a relaxed sub-problem of the whole problem, so at each node of the search tree, before creating branches, we look for neighborhood inequalities which are violated. If there is a violated inequality we add to the sub-problem a constraint to avoid the satisfaction of constraint (8).

# 5 Application of CSP by Branch and Cut algorithm:

## 5.1 The proposed algorithm:

The main algorithm composed of several stages of Branch and Cut algorithm in form of functions, each function has a set of instructions to do. The main algorithm is as follows:

```
Start
        Input: Insert List Containers arrived to Dock
    Select (Importing_Container) Or (Exporting_Container)
        If (Importing_Container) then
                Initialization();
            While (condition_is_verified=No) then
                Stopping_condition();
                Choosing_node();
                Relaxation();
                Adding_VI();
                Branch();
            EndWhile;
        Endif;
        If (Exporting_Container) then
            Unstack(Exporting_Container);
        Endif;
        Output:
        Print The nearest location of the importing container in yard
        Print State of yard (Number of free slots, number containers)
        Print State of Dock (Number of free slots, number containers)
End
```

### 5.2 Functions :

To fully understand the main algorithm before it is clear to detail each function by a simple algorithm. So we start function by function, we have:

**Explication of *Initialization()* function :**

At first, we create the root node $P^0$ of the search tree which represents the released problem, and initialize the list of active nodes in the search tree $T=\{P^0\}$, then we verify value of upper bound named *UB* which is calculated last times. If it's zero we put *UB* equal to infinity.

```
Initialization()
{
Start
    Declare P^0
    T ← {P^0}
        If UB=0 then
            UB=+∞;
        Endif;
End};
```

**Explication of *Stopping_condition ()* function :**

If *T* list is empty therefore the optimal solution is the one whose value equals to *UB* and if *UB* equal to infinity mean there is no solution.

```
Stopping_condition ()
{
Start
    If T=∅ then
        "The optimal solution is the one whose value =UB" ;
    Endif ;
    If UB=+∞ then
        "There is no solution " ;
    Endif ;
End};
```

**Explication of *Choosing_node ()* function :**

Choose from the list $T$ a node $P^l$ to explore, which will then be deleted next. It is possible to use one of the following methods to select a node to explore:
- The "opportunistic" method: selected from nodes that are not yet explored, one which has the smallest lower bound.
-The method of breadth first search: always trying to explore the nearest node to the root of the search tree. Thus, the root is the first node visit, followed by nodes that are derived from the root, etc.
-The method of depth first search: favors nodes that are farthest from the root, which are derived from already explored fathers.

**Explication of *Relaxation ()* function *:***

We solve the relaxation of $P^l$, if there is no solution we put $LB_l$ equal to infinity and going to step “*Penetration_and_pruning()*’’. Otherwise noted $S^{Rl}$ the optimal solution of $P^l$ and use its value to update $LB_l$.

**Explication of *Adding_VI ()* function :**

Search for valid inequalities *VI* are raped by $S^{Rl}$. if there has, we added them to the relaxation of $P^l$ and return to *Relaxation()*.

**Explication of *Penetration_and_pruning ()* function :**

If $LB_l$ is greater or equal than *UB* we return to step of *Stopping_condition()* and if $LB_l$ is less or equal than *UB* and $S^{Rl}$ is a feasible integer solution, we update the *UB* by putting the value of $LB_l$ in *UB*, then, we remove from the list $T$ any element $j$ whose lower bound is greater or equal than *UB* then return to *Stopping_condition()*.

**Explication of *Branch ()* function :**

If the solution $S^{Rl}$ is fractional and inferior to *UB* we make a connection to the most fractional variable, we obtain two sub-problems $P^{l,1}$ and $P^{l,2}$ which are added to $T$ list then we return to *Stopping_condition()*.

# CHAPTER 4

# PRESENTATION OF SOFTWARE PROPOSED

## Chapter Summary:

In this chapter, we illustrated an idea about how to store containers by computing distances and the computer tool used, and then we present the case of study, in which we present the port of Algiers and their data used, thereafter a description of the development tool used in programming, then we give a presentation of software and its interfaces. Finally, an illustration of a solution obtained by the software is performed.

# 1 Calculation Methods:

In our problem we have one criteria that must be minimized, which is the distance between dock in which we dispose the arrival containers at the first time and storage area of containers which is composed of blocks each block has a specific category of containers can stored. We proposed two methods to resolve our problem.

## 1.1 Distance Minimization Method:

In this method we used the stacks in which, the distance that we want to minimize is by number of stacks between locations of containers arrived at the first time in specific dock and stacks of storage area which had a same category with these containers. For this it’s important to know number of stacks for specific dock and the storage area of port so we brought the capacity of these both by $m^2$ and we divided by capacity of two types of containers (20 Foots and 40 Foots) to have number of stacks with two capacities. The number of slots is equal to the number of stacks multiplied by capacity of stack which is equal to 4 in our case.

## 1.2 Storage Method:

This method concerned the organization of containers storage it's for make containers which have the same category close to each with consideration of the depart dates, the container which has a nearest depart date we put it on the top of stack.

# 2 Presentation of the study area (Case Study: Port of Algiers):

This section is concerned simulation and resolution of container storage problem in the Algerian seaport: Port of Algiers. We begin by briefly presenting the port of Algiers. Next, we show experiments that helped to approve the expected performance of our solution into reality. In this illustration we presented the main concept regarding the study area that has chosen.

### 2.1 Presentation of port:

The Port of Algiers serves several States of the country. Are privileged hinterland covers the center and west center. Other parts of the country, including the south, can be considered part of the hinterland of the port of Algiers, and that, considering the flow of goods generated by the activities of oil companies.

Algiers port has a special geographical position in the Mediterranean and also nationwide making it the first commercial port in Algeria. The port city within the city covers an overall area of 126 hectares.

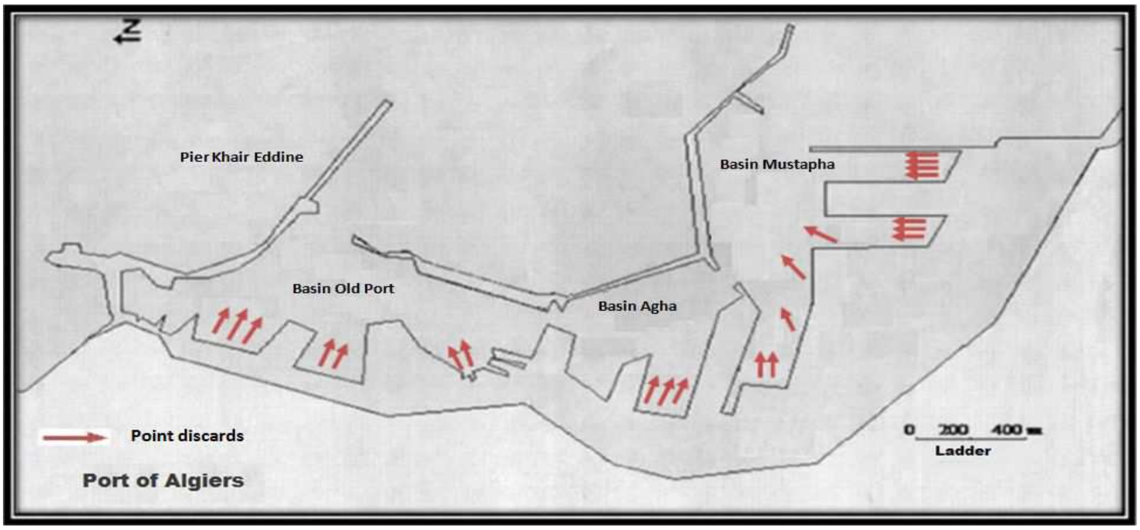



**Figure 4.1** Card of Algiers port LEM [22]

Algiers port has a total area of 282,000 m² warehousing, representing 24% of the total surface evenly distributed among the three geographical areas of the harbor, welcoming various goods:

- Full land of **232.000 m²**
- 12 stores of **50.000 m²**

This capability allows storage of 120,000 tons of goods, while the average monthly volume landed today is 800,000 tons [34].

It is a facility for maintenance work on the infrastructure of the port of Algiers, and is operated by the company Meditram. The block park is located the dock 26 (**Figure 4.2**) and occupies an area of 20 000 $m^2$. It has a dock work for berthing pontoons and barges.


**Figure 4.2** Dock N° 26 [36]

Built on an area of 30.3 ha, the container terminal has been approved in 1998 and has two positions each 300 ml wharf with a draft of between 9 and 11 meters. It includes a packing center and unloading of 4,800 m², a maintenance workshop of 2,000 m², a park gear 1.000 m², an administrative building of 2,400 m² and a checkpoint.

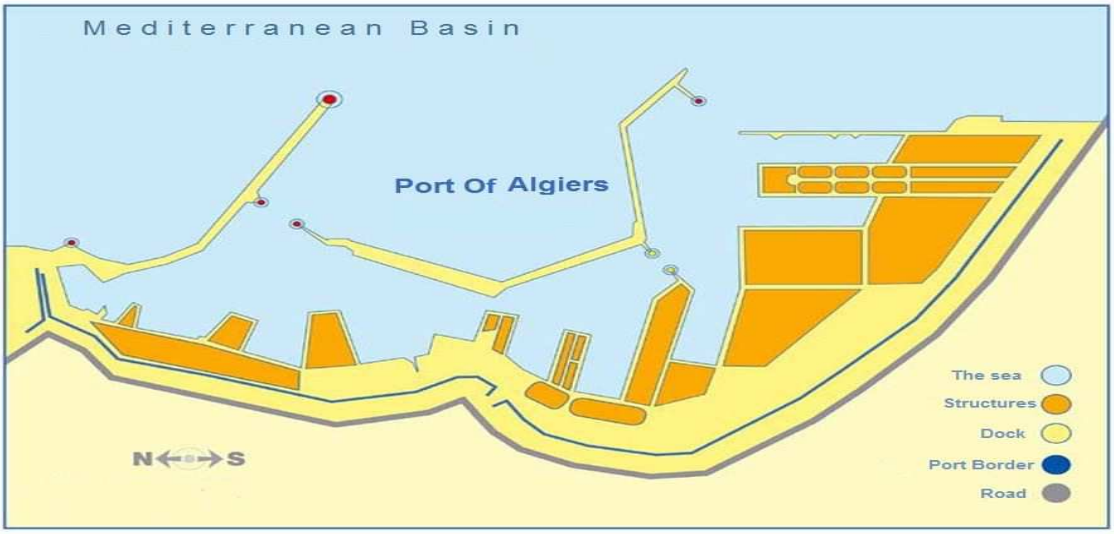



**Figure 4.3** Port of Algiers [34]

**2.2 Data used:**

There were several docks in the port of Algiers, so we have chosen a single dock for applied our study which is the dock N° 26 which has a specific categories of containers that can carry which are; " Bitume, exp/f, G.oil " [34], so starting from it we recovered arrivals containers for storing them in the storage space, the data used in this study are represented in the following table:

| | Dock N° 26 | Storage area |
|---|---|---|
| Surface ($m^2$) | 20 000 | 4800 |

**Table 4.1** Data of port concerning surface of dock and storage area

To determine number of stacks in dock and storage area we have calculated the surface of both sizes of containers exist in the port of Algiers (20 foots and 40 foots) **Table 4.2**.

| | 1 TEU | | 2 TEU | |
|---|---|---|---|---|
| | Length | Width | Length | Width |
| Foots (ft) | 20 | 8 | 40 | 8 |
| Miter(m) | 6,096 | 2,4384 | 12,192 | 2,4384 |
| Surface ($m^2$) | 14.864 | | 29.729 | |

**Table 4.2** Sizes of containers

And from the two possibilities we have divided the surface of each(dock and storage area), we get after division the number of stacks of each of them and to determine the number of slots, each stack has four floors, so the number of stacks multiply by the number of floor, as we are seeing in **Table 4.3.**

| | Number of stacks | Number of slots |
|---|---|---|
| Dock N° 26 | 834 | 3336 |
| Storage Area | 200 | 800 |
| Total | 1034 | 4136 |

**Table 4.3** Number of stacks and slots

## 3 Development tool used:

To develop the algorithms used in our optimization approach, we used the programming language Visual Basic object-oriented (VB). This language is the third generation oriented event programming language. Microsoft Corporation released Visual Basic in 1987. This was indeed the first visual development tool from Microsoft. Visual Basic was derived from the fast development in BASIC language, VB allows graphical user interfaces (GUI) development, access to databases, the creation of ActiveX controls and objects. The language not only allows Visual Basic programmers to create simple GUI applications, but also helps to develop complex applications. Visual Basic allows developers to target Windows, Web and mobile devices...

Our choice is based on the characteristics of the language. In visual basic a developer could easily translate an abstract idea into a program design that he can actually see on the screen. VB encourages developers to experiment, revise, correct, and design until the project meets the requirements requested. The Visual Basic Programmer uses the language in various fields such as education, business, accounting, marketing and sales. Visual Basic supports a number of buildings and common programming language elements. Once you understand the basics of the language, you can create powerful applications using Visual Basic. Visual Basic can create executables files, ActiveX controls. It is also used for systems of web database interface. This generation of Visual Basic continues the tradition of giving a quick and easy way to create applications. .NET Framework based Visual Basic also fully integrates the Framework.NET and the Common Language Runtime, which provide language interoperability, enhanced security, and versioning support [40].

## 4 Software Description:

The developed software is an application to solve the CSP. It treats the organization of storage area. This is a tool for decision support that helps to store a large number of containers with respecting various constraints associated with the problem. In this section, we give an explanation about the use of this software. The software is divided into three interfaces. The first one is for login, in which the user must enter his username and his password as shown in **Figure 4.4**:

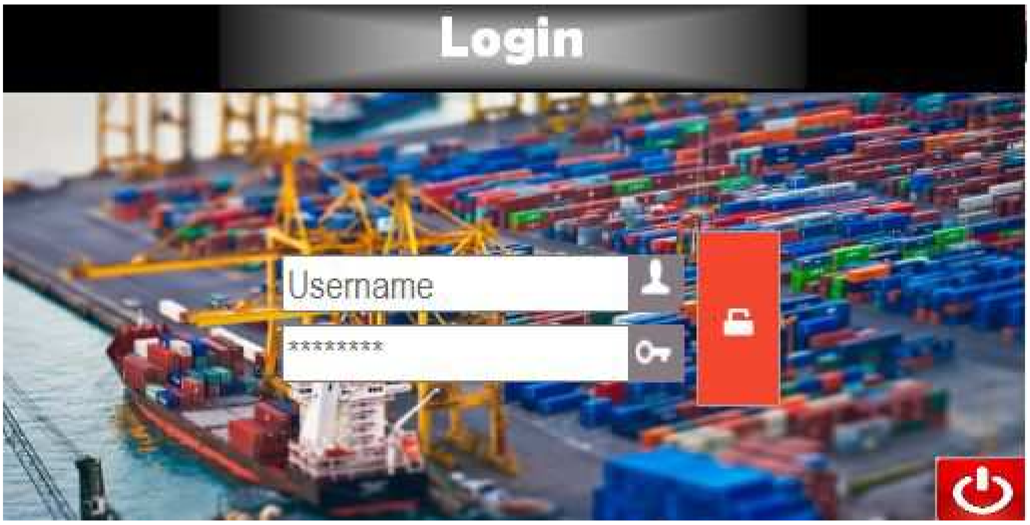



**Figure 4.4** Login page

Then the user clicks to Padlock button to login the main page, if one of username or password is incorrect an error message will be displayed.


**Figure 4.5** Main page

The main page is represented in **Figure 4.5**, it contained four tabs; Main, Storage, Exportation, Importation, the first tab showing in this page is Main tab in which we can insert the containers arrived at port to a specific dock by select number of dock (dock N°26) then the category of containers according to the categories of dock has chosen then we specified the number of containers arrived when we finished we click on Add Containers To Dock button (**Figure 4.6**).


**Figure 4.6** Main tab of dock

A page showing in **Figure 4.7** it will appear to chose type and date of arrive of container then we click to Add button for added it when we added all the containers this page will closes by itself.


**Figure 4.7** Insert containers in Dock page (insert type of 20Foots or 40Foots)

In **Figure 4.8** we have Storage tab in which we can see status of storage area or dock, imports and exports.

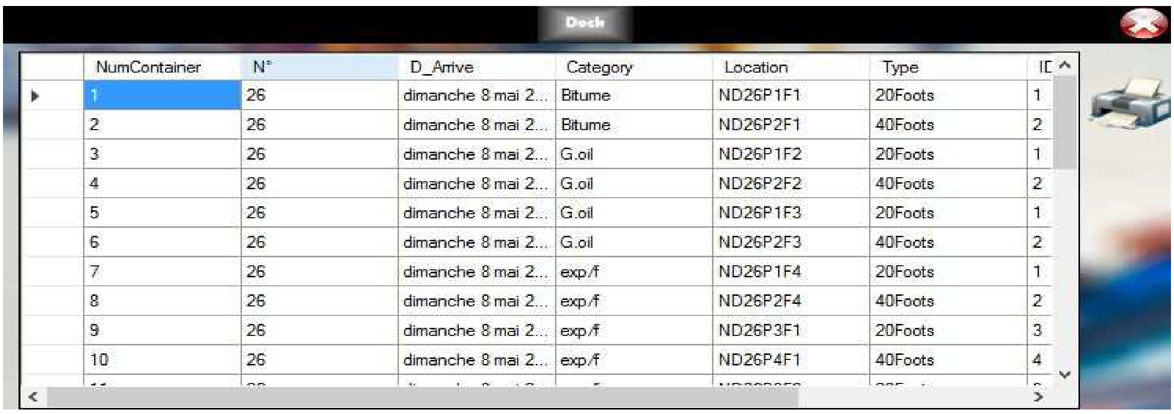

**Figure 4.8** Storage tab

When we click on Status of Storage Area button the page in **Figure 4.9** will display, in which we can see the bale of status.


| Port | Number_Of_Containers | Free_Slots |
|---|---|---|
| Yard_Of_Storage | 0 | 800 |
| Dock | 31 | 3305 |

**Figure 4.9** Status of storage area

It's the same thing when we click on Dock button (**Figure 4.10**), we can see the list of containers in Dock.




| NumContainer | N° | D_Arrive | Category | Location | Type | IC |
|---|---|---|---|---|---|---|
| 1 | 26 | dimanche 8 mai 2... | Bitume | ND26P1F1 | 20Foots | 1 |
| 2 | 26 | dimanche 8 mai 2... | Bitume | ND26P2F1 | 40Foots | 2 |
| 3 | 26 | dimanche 8 mai 2... | G.oil | ND26P1F2 | 20Foots | 1 |
| 4 | 26 | dimanche 8 mai 2... | G.oil | ND26P2F2 | 40Foots | 2 |
| 5 | 26 | dimanche 8 mai 2... | G.oil | ND26P1F3 | 20Foots | 1 |
| 6 | 26 | dimanche 8 mai 2... | G.oil | ND26P2F3 | 40Foots | 2 |
| 7 | 26 | dimanche 8 mai 2... | exp/f | ND26P1F4 | 20Foots | 1 |
| 8 | 26 | dimanche 8 mai 2... | exp/f | ND26P2F4 | 40Foots | 2 |
| 9 | 26 | dimanche 8 mai 2... | exp/f | ND26P3F1 | 20Foots | 3 |
| 10 | 26 | dimanche 8 mai 2... | exp/f | ND26P4F1 | 40Foots | 4 |

**Figure 4.10** Storage of Dock

We can also print this data as report for that we just click on printer button then we can see the report in other page (**Figure 4.11**).

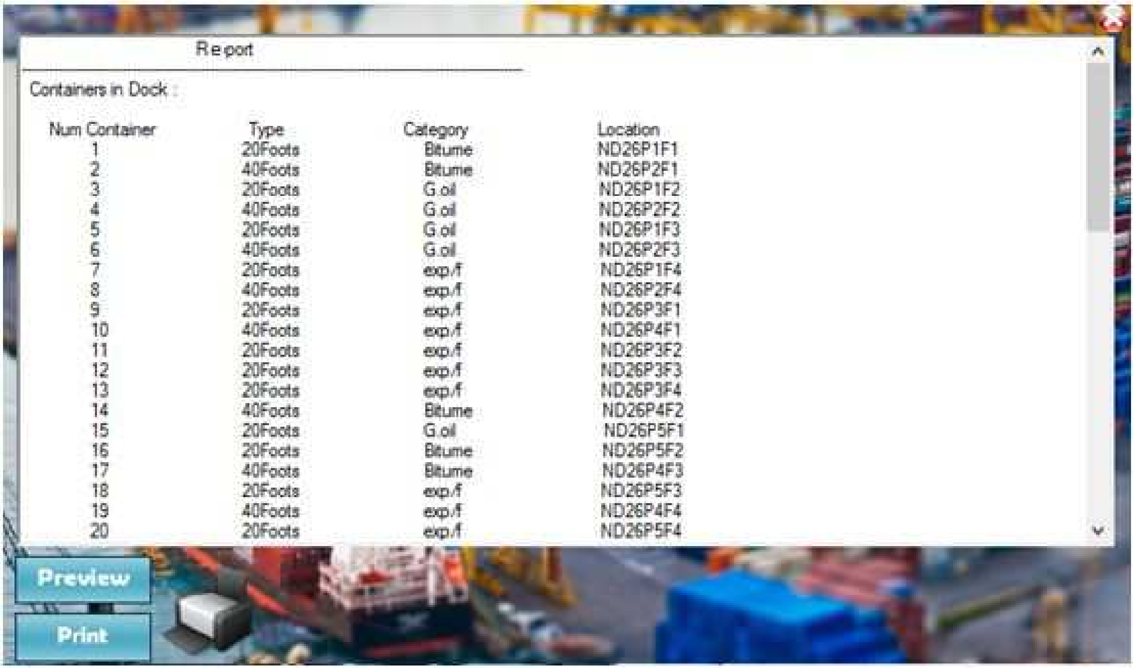



**Figure 4.11** Storage dock report

We can modify if we wanted, we can also see the report before we print it by click on Preview button as we see in **Figure 4.12**, and for print it we just click on Print button.

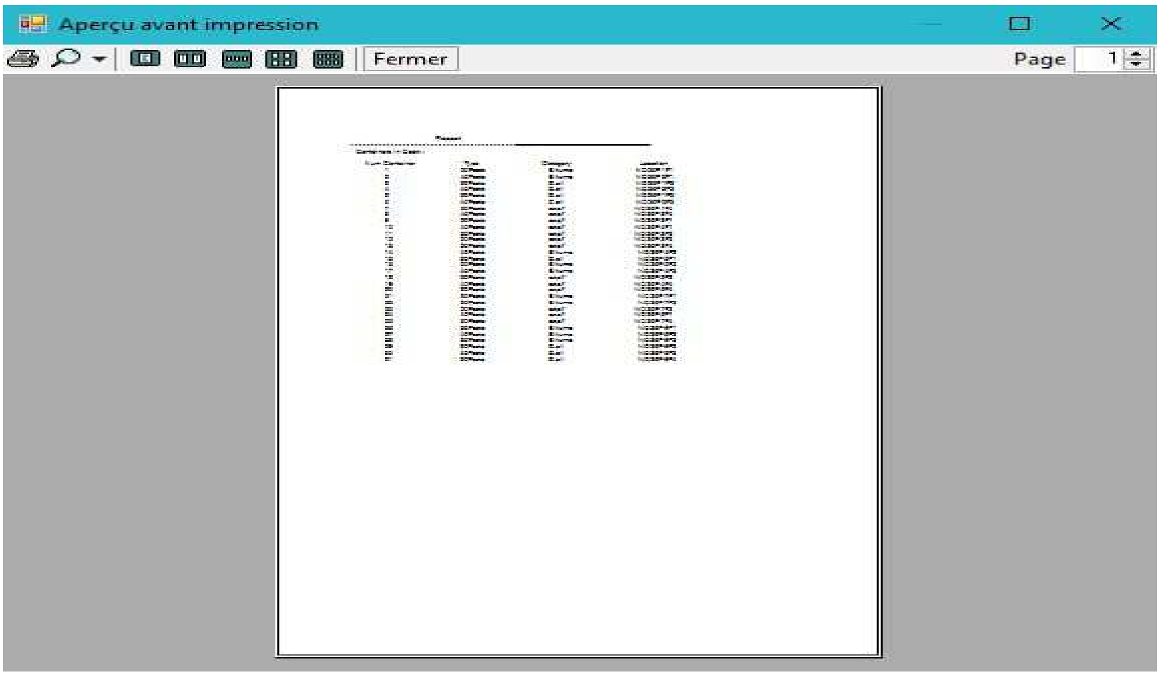



**Figure 4.12** Preview before print report

In **Figure 4.13** we see Exportation tab in which a list of containers in storage area is display for chose from it which containers we want to export it.


**Figure 4.13** Exportation tab

The last tab represented in **Figure 4.14** which is Importation tab in which a list of containers in dock is display for put containers from dock to storage area. We select these containers one by one i.e. when we select container from list we click on "+" button (select three containers for example) when we finished we click on Import button.


**Figure 4.14** Importation tab

## 5 An example illustration:

To evaluate our work we have an example:

We chose from list of containers in dock three containers to store them, their information is shown in **Table 4.4.**

| Num Container | N_Dock | Category | Location | Type | Stack | Floor |
|---|---|---|---|---|---|---|
| 23 | 26 | Exp/f | ND26P7F3 | 20Foots | 7 | 3 |
| 24 | 26 | Exp/f | ND26P6F1 | 40Foots | 6 | 1 |
| 25 | 26 | Exp/f | ND26P7F4 | 20Foots | 7 | 4 |

**Table 4.4** Represent information of containers which we have chosen from dock

After we chose these containers, they will be removed from dock and stored in storage area, so their information will be change concerned the location, it will be as we can see in **Table 4.5.**

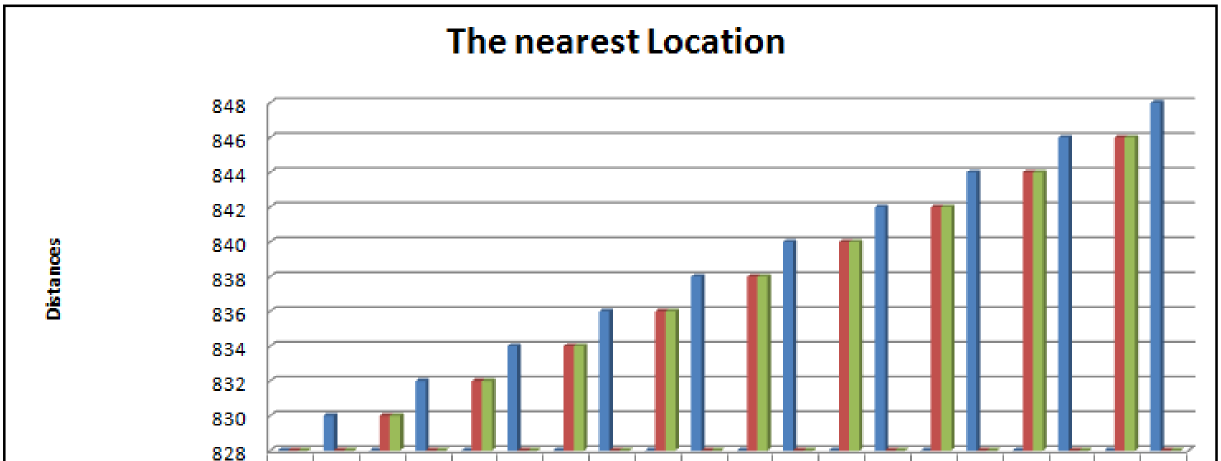
| Num Container | N_Block | Category | Location | Type | Stack | Floor |
|---|---|---|---|---|---|---|
| 24 | 1 | Exp/f | NB1P836F1 | 40Foots | 836 | 1 |
| 25 | 1 | Exp/f | NB1P835F1 | 20Foots | 835 | 1 |
| 23 | 1 | Exp/f | NB1P835F2 | 20Foots | 835 | 2 |

**Table 4.5** Represent information of containers in storage area

These containers will also showing in exportation tab as we can see in **Figure 4.13**, in which we can chose from list of containers in storage area the containers that we want to export if we chose one container for example it will be remove from storage area and saved in exports list in data base of the software.

## 5.1 Analysis of results:

We have taken just the 20 first stacks of storage area because number of stacks is so large(from 835 to 1034), we can see the statistics in **Figure 4.15** which represent the different distances between location of the three containers and the stacks of storage area, where the distance is calculated between each container and a stack which has the same size of this container therefore we see some distances equal to zero because these distances are calculated between container and stack which haven't the same size.




| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Container1(num 24) | 0 | 830 | 0 | 832 | 0 | 834 | 0 | 836 | 0 | 838 | 0 | 840 | 0 | 842 | 0 | 844 | 0 | 846 | 0 | 848 |
| Container2(num 25) | 828 | 0 | 830 | 0 | 832 | 0 | 834 | 0 | 836 | 0 | 838 | 0 | 840 | 0 | 842 | 0 | 844 | 0 | 846 | 0 |
| Container3(num 23 ) | 828 | 0 | 830 | 0 | 832 | 0 | 834 | 0 | 836 | 0 | 838 | 0 | 840 | 0 | 842 | 0 | 844 | 0 | 846 | 0 |

**Figure 4.15** The nearest location

In **Figure 4.15** we can see that for the two containers 2 and 3 the distance calculate for the same stacks because there are same types (1TEU) and it's the same size also for the stacks. But for the container 1 the distance was calculated for different stacks which have the same size with him (2TEU).

The choice of stacks here is based on two constraints; first about type and size of stack ,second about the nearest stack from container location, as we see in **Figure 4.15** the nearest stack from containers 2 and 3 is the stack 1 so they were stored in stack 835 which is the first stack in storage area in which the container 2 is stored in floor 1 and the container 3 in floor 2, in other part the container 1 stored in stack 2 (836) which is the nearest one in which it is stored in floor 1 as in **Table 4.5**.

# CONCLUSION

# AND PRESPECTIVES

# CONCLUSION AND PRESPECTIVES

This work presents a solution of the CSP by using the "Branch and Cut algorithm" for a good organization of containers in storage area and to find an optimal storage plan. It treats one of the container terminal problems which help to minimize the distance between placements of containers that belong to the same category in the storage area.

Our personal contribution is the application of "Branch and Cut algorithm" to solve the CSP by a software which based on this algorithm to give effective and rapid results and it should be simple and easy to use by the concerned user of course, to help him make the right decision about the container storage operations. To evaluate our software we took the port of Algiers as a case study.

As future prospects there are some perspectives: Apply other methods in parallel to solve more than one problem, study both cases static and dynamic storage at once, studied maximizing the number of stored containers, study more than one port.

# BIBLIOGRAPHY

# BIBLIOGRAPHY

## Books :

[1] Ashton M Arthur, Miller Martin J, Container storage systems, Patentscope Espacenet, French, 1999

[2] B. Borgman, E. V. Asperen, and R. Dekker, Online rules for container stacking, OR Spectrum, French, 2010.

[3] Dr. Jean-Paul Rodrigue, World Container Production, 2007, The Geography of Transport Systems, French, 2015.

[4] J. Cheng, Biomass To Renewable Energy Processes, CRC Press, USA, 2009.

[5] Kim K-H. Kim, H-B., The optimal determination of the space requirement and the number of transfer cranes for import containers, Computers ind. Engng, Korea, 1998.

[6] Kim K-H Park K-T , A note on a dynamic space-allocation method for outbound containers, European Journal of Operational Research, Korea, 2003.

[7] Kim K-H , Evaluation of the number of rehandles in container yards, Computers & Industrial Engineering, Korea, 1997.

[8] Levinson Marc, How the Shipping Container Made the World Smaller and the World Economy Bigger, Princeton, USA, 2013.

[9] M. B. Duinkerken, J. J. M. Evers, and 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.

[10] R. Dekker, P. Voogd, and E. V. Asperen, Advanced methods for container stacking, Container Terminals and Cargo Systems, USA, 2007.

[11] Russell G. Thompson, City Logistics: Mapping the Future, Eiichi Taniguchi, Australia, 2014.

[12] Wayne K. Talley, Port Economics, Routledge, USA, 2009.

[13] Zhang, C., Liu, J., Wan, Y-W., Murty, K-G., Linn, R-J., Storage space allocation in container terminals, Transportation Research Part B, USA, 2003.

## Articles :

[14] H. Toussaint, Introduction au Branch and cut Price et au solveur SCIP (Solving Constraint Integer Problems), LIMOS, Vol.33, 2013, pp.13-07.

[15] I. Ayachi, R. Kammarti, P. Borne, M. Ksouri, Harmony search to solve the container storage problem with different container types , International Journal of computer Application, Vol. 48(22),Tunisia French, 2012, pp. 26-32.

[16] John E., Mitchell, Branch-and-Cut Algorithms for Combinatorial Optimization Problems, Handbook of Applied Optimization, USA, 1999, pp 65–77,

[17] M. anfred Padberg and Giovanni Rinaldi, A Branch-and-Cut Algorithm for the Resolution of Large-Scale Symmetric Traveling Salesman Problems, Siam Journal, Vol.33, USA, 1991, pp. 60–100.

[18] M. Rebollo, Vicente Julian, Carlos Carrascosa and Vicente Botti, A Multi-Agent System for the Automation of a Port Container Terminal, Univercity of polytechnic of Valencia, Spain, 2000.

[19] Murty K.G., Liu J., Wan Y.W, Zhang C., Tsang M.C.L.,Linn R., A decision support system for operations in a container terminal, Journal Decision Support Systems, Vol. 39, USA, 2005.

[20] N. Fatma Ndiaye, A Branch-and-Cut Algorithm to Solve the Container Storage Problem, University of Le Havre, Franch, 2014.

[21] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, Optimization by simulated annealing, Science New Series ,vol. 220, 1983, pp. 671-680.

## Dissertations :

[22] C. LAAMA, Contribution à l'étude des paramètres physico-chimiques et bactériologiques des eaux du port d'Alger, magisterium, Ecole Nationale Supérieure des Sciences de la Mer et de l'Aménagement du Littoral DEUA en sciences de la mer, Algeria, 2009.

[23] I. Ayachi Hajjem, Techniques avancées d'optimisation pour la résolution du problème de stockage de conteneurs dans un port, Doctorate, Ecole centrale de Lille université de Tunis el Manar école nationale d'ingénieurs de Tunis, Lille French, 2012.

[24] K. Chebli, Optimisation des mouvements des conteneurs dans un terminal maritime , magisterium, Université de montréal Département de mathématiques et génie industriel Ecole polytechnique de montréal, Canada, 2011.

[25] M. Kefi Gazdar, Optimisation Heuristique Distribuée de Problème de Stockage de Conteneurs dans un Port, Doctorate, Computer Science, Ecole Centrale de Lille, French, 2009.

[26] Moustapha Abdi Bouh, Analyse des operations de manutention des conteneurs du PAD, Master, Ecole superieure de commerce de dakar, Senegal, 2012.

[27] N. Fatma NDIAYE, Algorithmes d'Optimisation pour la Résolution du Problème de Stockage de Conteneurs dans un Terminal Portuaire, Doctorate, Havre Univercity, Franch, 2015.

[28] O. Sharif Abul Bashar Mohammad, Application of Agent-Based Approaches to Enhance Container Terminal Operations, Doctorate, University of south Carolina, Columbia, 2013.

[29] P. Fouilhoux, Mise en oeuvre d'un algorithme de Branch-and-Cut avec le framework SCIP, Master, Pierre and Marie Curie University, 2015.

## Websites :

[30] CNAN Nord spa, http://www.cnan-nord.com/presentation.php , consulted: 20/01/2016.

[31] Container Handbook, https://www.containerhandbuch.de/chb_e/stra/index.html?/chb_e/stra/stra_01_01_00.html , consulted: 04/03/2016.

[32] Container Transportation, http://www.container-transportation.com/container-terminal.html , consulted: 17/02/2016.

[33] Dearborn Verhea Crane http://www.dearborncrane.com/crane_buyers_guide/gantry_cranes.htm , consulted: 18/03/2016.

[34] Entreprise Portuaire D'Alger EPAL, http://www.portalger.com.dz/ , consulted: 15/04/2016.

[35] Gorbel, https://www.gorbel.com/products/cranes/gantry-cranes , consulted: 15/03/2016.

[36] Google map, https://www.google.dz/maps?hl=fr&tab=wl , consulted: 10/04/2016.

[37] Industrial Engineering & Management Systems http://www.iemsjl.org/journal/article.php?code=1600 , consulted: 15/03/2016.

[38] Iris F.A. Vis, http://www.irisvis.nl/container/processes.html , consulted: 15/03/2016.

[39] Science Direct, http://www.sciencedirect.com/science/article/pii/S0360835208002003 , consulted: 05/02/2016.

[40] Wikipedia, http://en.wikipedia.org/wiki/VisualBasic , consulted: 16 /04/2016.

# ANNEXES

# ANNEX A

## Combinatorial optimization:

Combinatorial optimization is an approach to finding the best solution out of a very large set of possible solutions. When the set is so large that it's impractical to search through all of them, various techniques can be used to narrow down the set or speed up the search. Combinatorial optimization is a subset of mathematical optimization that is related to operations research, algorithm theory, and computational complexity theory. It has important applications in several fields, including artificial intelligence, machine learning, mathematics, auction theory, and software engineering [29].

Some research literature considers discrete optimization to consist of integer programming together with combinatorial optimization which in turn is composed of optimization problems dealing with graph structures although all of these topics have closely intertwined research literature. It often involves determining the way to efficiently allocate resources used to find solutions to mathematical problems.

Applications for combinatorial optimization include, but are not limited to:

- Developing the best airline network of spokes and destinations
- Deciding which taxis in a fleet to route to pick up fares
- Determining the optimal way to deliver packages
- Determining the right attributes of concept elements prior to concept testing

**Methods:**

There is a large amount of literature on polynomial-time algorithms for certain special classes of discrete optimization, a considerable amount of it unified by the theory of linear programming. Some examples of combinatorial optimization problems that fall into this framework are shortest paths and shortest path trees, flows and circulations, spanning trees, matching, and matroid problems.

For NP-complete discrete optimization problems, current research literature includes the following topics:

- Polynomial-time exactly solvable special cases of the problem at hand (e.g. see fixed-parameter tractable).
- Algorithms that perform well on "random" instances (e.g. for TSP).
- Approximation algorithms that run in polynomial time and find a solution that is "close" to optimal.
- Solving real-world instances that arise in practice and do not necessarily exhibit the worst-case behavior inherent in NP-complete problems (e.g. TSP instances with tens of thousands of nodes).

Combinatorial optimization problems can be viewed as searching for the best element of some set of discrete items; therefore, in principle, any sort of search algorithm or metaheuristic can be used to solve them. However, generic search algorithms are not guaranteed to find an optimal solution, nor are they guaranteed to run quickly (in polynomial time). Since some discrete optimization problems are NP-complete, such as the traveling salesman problem, this is expected unless P=NP.

## Classical problems in combinatorial optimization:

### The problem of bag-to-back:

Consider $n$ objects, denoted $i = 1,...,n$, each providing this benefit but with a weight $a_i$. We want to store items in a "bag" we want maximum weight $b$. The problem of bag-to-back (knapsack) is to select the objects to be among the $n$ objects in order to have maximum benefit and respect the constraint weight not to exceed. each object $i$, $i \in \{1,...,n\}$, must be selected at least $p_i$ and at most $q_i$ time. This problem occurs of course when we start hiking wanting to take as much as possible of useful items (food, drinks ...). But this problem is more frequently used to fill trucks, aircraft or cargo boats and even to manage memory of a microprocessor.

Let E = {1,...,n} a finite set of elements. Let $E_1,...,E_m$ subsets of E. Each set Ej we associate a weight $c_j$, j = 1,...,m. A family $F \subseteq \{E_1,...,E_m\}$ is known a covering E if

$\cup_{E_j} \in F_{E_j} = E$, for every $j \in \{1,..., m\}$, a paving E if $E_j \cap E_k = \emptyset$, for every $j \in \{1,...,m\}$, $j \neq k$ a partition E if F is both a cover and a paving. take for example elements 5 $E = \{1,2,3,4,5\}$ and 4 subsets $E_1 = \{1,2\}$, $E_2 = \{1,3,4,5\}$, $E_3 = \{3,4\}$ and $E_4 = \{3,4,5\}$. One can easily notice in **Figure A.1** let $\{E_1, E_2\}$ is a covering, which $\{E_1, E_3\}$ and is paving $\{E_1, E_4\}$ is a partition.

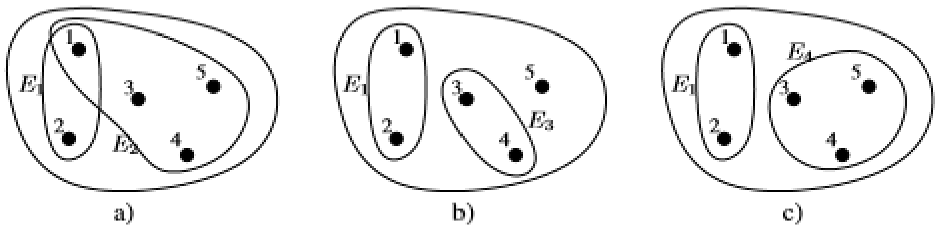



**Figure A.1** Illustration of a) Recovery, b) Paving and c) Partition

The covering problem (resp. Paving, partition) is to determine a recovery (resp. Paving, partition) whose combined weight sets that form the minimum weight is (resp. Maximum, minimum / maximum). One can interpret the constraints of collection problem (resp. Paving, partition) as the fact that an element of E must be taken at least once (resp. more than once, just once). These problems have multiple applications. For example, consider a region where it is desired implanting year fire stations to cover all towns. For each potential barracks determining the municipalities served and the cost of installation of the barracks. As we want to cover all towns, it is clearly an overlap problem.

**The problem of steady**

Let G = (V, E) an undirected graph *c* and a weight function that assigns to each vertex $v \in V$ a weight $c(v)$. A stable G is a subset S of vertices V such that there is no edge E between two vertices of S. The problem maximum weight stable includes determining a steady S of G such that $c(S) = P_v \in S\ c(v)$ is maximum. The **Figure A.2** gives such a graph where all vertices are valued weight 1. It may be noted which all the isolated peaks are 1 weight stable and a stable weight is a maximum of 2, which is obtained for example by stable $\{v_1,v_5\}$.

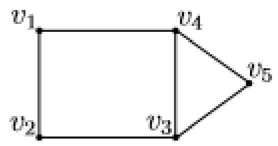



**Figure A.2** Illustration stable problem

**The traveling salesman problem**

Let G = (V, E) an undirected graph, where the vertices V are called cities and the edges between two cities bonds. Let c(*e*) a cost associated with the e bond, for any $e \in$ E. The TSP is to determine a cycle through each city once and only once. Such a cycle is called Hamiltonian; it is called as often round or tour. This classic problem has found many applications in areas yet as far which milk production or construction of integrated circuits.

**The coloring problem**

Let S ⊆ IN. A vertex coloring of a graph G = (V, E) is a function r : V → S such as r(*u*) ≠ r(v) for every pair of adjacent vertices *u*, *v*. A k-coloring is a coloring c : V → {1,..,k}. A graph is said-colorable if it has coloration. Test whether a graph is k-colorable is NP-complete problem if k ≥ 3, and if it is polynomial k = 2. While k = 2, just test if the graph is bipartite, it is to say it contains no odd cycle. This can be done by a simple graph traversal. Let G = (V, E) an undirected graph. The coloring problem is to determine the smallest k such that G is k-colorable [29].

# ANNEX B

## Storage strategies:

One advantage of using containers is that it is possible to superimpose on each other. However, this advantage has limitations insofar as it can cause alterations. Indeed, this kind of operation is performed mainly in the extraction containers that are stacks funds. Therefore, it is essential for each container terminal, to adopt adequate storage strategy. The different storage methods that exist in the literature can be divided into four categories according Saanen and al which are; Non-segregation and segregation, Consolidation and dispersion, Consolidation and dispersion, Direct and indirect storage, Consolidation and dispersion, and priority unloads and loads which is detailed below[27].

## Priority unloads and loads:

Storage methods that collect the unloading of containers seek to maximize the performance of all activities related to storage operations. The level storage method is the illustration. It stores the containers by layer, so that all ground slots are occupied, before superimposing the containers. This strategy was proposed by Duinkerken and al. in [9].

It is intuitive but does not use much of the information available. It contains four main steps, which follow these steps.

1 : Take any row that has at least one free slot.

2 : Search in this row, a free and proper location which is in contact with the ground.

3 : If found: y storing the container.

4 : If it is not found: search in the row, a free and suitable location that belongs to the lowest possible level.

With storage level by the risk of reworking is less important than the random method; moreover it can be non-existent when the entire surface of the terminal is not occupied.

However, its effectiveness is questioned when the number of containers to be stored is much higher than battery. Another weakness of this method is that it may require traveling long distances in the storage yard, which can increase the time required to perform storage or retrieval operations.

Unlike these methods, those receiving unloads of containers are designed to maximize the performance of withdrawal operations. This is the case of the method that stores the following containers descending order of their starting dates [10]. An improved version of this method, known as storage by leveling departure dates, is proposed [2] by Borgman and al. Their idea is to store the containers following descending order of their departure dates, while trying to minimize the used area. Thus, if several stacks are compatible with a container, then it is stored in the one that is the highest. In addition to that, storage is also minimizing the gaps between the starting dates of two containers which follow one another in the same stack. Leveling therefore is not done from the ground but from baseline dates. With this method, the search for a storage location for a container is done in three stages which follow one another as follows.

1 : First we look from batteries that are neither full nor empty, those that have, at their tops, containers that have departure dates than that of the container you want to store. If we find we calculate, for each of them, the difference between the starting date of the container that is at its peak and that of the container which it is sought
storing. Then the cell which leads to the smallest difference is selected.

2 : If such cells do not exist, then selects from spent batteries, that which is closest to the outlet through which the container will be delivered.

3 : If we did not find stack that belongs to the first two cases were then stores the container in the highest stack among those who are not full ; to minimize future revisions. according to Borgman and al. [2], this method is more effective than random storage and storage by level, because the risk of having alterations is much lower. Nevertheless, it favors the minimization of the storage area used to the detriment of speed storage or retrieval that can have impacts on customer satisfaction. Combinations between different storage strategies are proposed in the literature, they will be described in the next section.

# ANNEX C

## Example of application branch and cut algorithm:

We consider storage of 3 containers which have different types in three stacks which are initialize by empty. We supposed that its dates of departure are same for these three containers, and the max capacity of stack is equal to 4 and each stack has specific size. The distances used in this example are noted in **Table C.1.**

| P | $m_p^1$ | $m_p^2$ | $m_p^3$ |
|---|---|---|---|
| 1 | 829 | 828 | 828 |
| 2 | 830 | 829 | 829 |
| 3 | 831 | 830 | 830 |

**Table C.1** Table of distances

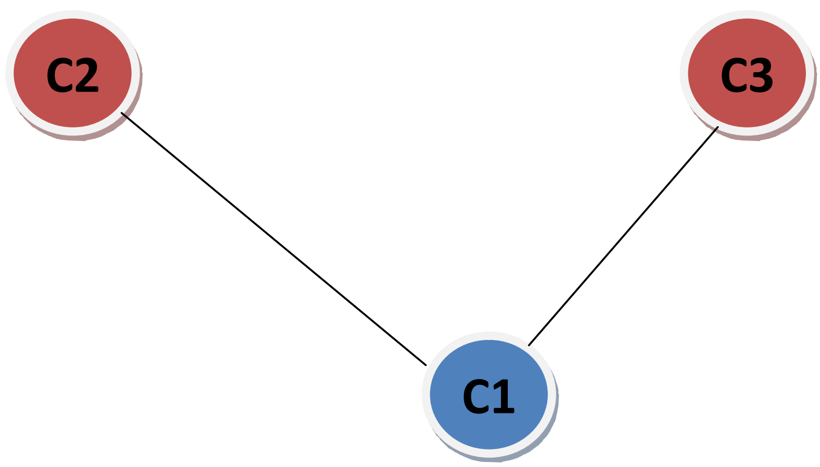



**Figure C.1** Incompatible graph

In **Figure C.1** we have the construction of incompatible graph according to our example we have C2 and C3 have the same type "20foots", C1 has type "40foots", and it's same thing for stacks P2 has size "40foots" and P1 and P3 have size "20foots".

We search UB of problem, we start by the container which has the maximum number of neighbors which is C1, it's assigns to P2 because it has the same size of C1 and it's the nearest one, After that we assigns the others of containers in which we must take in consideration the size of stack: C1 to P2, C2 to P3 and C3 to P3.
UB = $\{m_2^1, m_3^2, m_3^3\}$ = 2490

According to this data we have the following linier program:
Minimize

$829x_1^1$+$828x_1^2$+$828x_1^3$+

$830x_2^1$+$829x_2^2$+$829x_2^3$+

$831x_3^1$+$830x_3^2$+$830x_3^3$

Sub-constraints

(1)$x_1^1 + x_2^1 + x_3^1 \leq 1$

(2)$x_1^2 + x_2^2 + x_3^2 \leq 1$

(3)$x_1^3 + x_2^3 + x_3^3 \leq 1$

(4)$x_1^2 + x_1^1 \leq 1$

(5)$x_2^2 + x_2^1 \leq 1$

(6)$x_3^2 + x_3^1 \leq 1$

(7)$x_1^3 + x_1^1 \leq 1$

(8)$x_2^3 + x_2^1 \leq 1$

(9)$x_3^3 + x_3^1 \leq 1$

(10)$x_1^1 + x_1^2 + x_1^3 \leq 4$

(11)$x_2^1 + x_2^2 + x_2^3 \leq 4$

(12)$x_3^1 + x_3^2 + x_3^3 \leq 4$

(13)$x_p^q \in \{0,1\},\ \forall\, q = 1,2,3;\ p = 1,2,3$

The search tree T= $\{P^0\}$ then we gets $R^0$ after the relaxation of $P^0$.

Minimize

$829x_1^1$+$828x_1^2$+$828x_1^3$+

$830x_2^1$+$829x_2^2$+$829x_2^3$+

$831x_3^1$+$830x_3^2$+$830x_3^3$

Sub-constraints

(1)$x_1^1$+ $x_2^1$+ $x_3^1 = 1$

(2)$x_1^2$+ $x_2^2$+ $x_3^3 = 1$

(3)$x_1^3$+ $x_2^3$+ $x_3^3 = 1$

(10)$x_1^1$+ $x_1^2$+ $x_1^3 \leq 4$

(11)$x_2^1$+$x_2^2$+$x_2^3 \leq 4$

(12)$x_3^1$+$x_3^2$+$x_3^3 \leq 4$

(13)$x_p^q \geq 0, \ \forall\, q = 1,2,3;\ p = 1,2,3$

After that we solve $R^0$ we get results by simplex as follows:

($x_1^1$=0 ,$x_1^2$=0 ,$x_1^3$= 0 ,$x_2^1$= 1 ,$x_2^2$= 1 ,$x_2^3$= 1 ,$x_3^1$= 0 ,$x_3^2$= 0 ,$x_3^3$= 0 ) we get 2488. This solution isn't feasible because it doesn't satisfy the valid inequalities follows:

(14) $x_2^2$ + $x_2^3$ +2$x_2^1 \leq 2$

(15) $x_2^1 + x_2^2 \leq 1$

(16) $x_2^1 + x_2^3 \leq 1$

So we add these valid inequalities (14),(15),(16) to $R^0$ . Then we solve it again, ($x_1^1$=0 ,$x_1^2$= 0,1 ,$x_1^3$= 0 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 0.5 ,$x_3^3$= 0.2 ) we get 1493,8. This solution doesn't violate any valid inequality but it's not full.

We make a branch in variable of $x_3^2$ which has value of branching, so we get two sub-problems $P^{01} = P^0 \cup \{x_3^2=0\}$ and $P^{02} = P^0 \cup \{x_3^2=1\}$, we add this two sub-problems in T then we delete $P^0$ from T. The T list will be T= $\{P^{01}, P^{02}\}$.

While T not empty we choose node from T for relax it, we select $P^{01}$, we relaxed it then we get $R^{00}$ so we solve it which give to us ($x_1^1$=0 ,$x_1^2$= 0.3 ,$x_1^3$= 0 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 0 ,$x_3^3$= 0.5 ) equal to 1493,4 . This solution doesn't violate any valid inequality but it's not full. We make a branch in variable of $x_3^3$ so we get two sub-problems $P^{011} = P^{01} \cup \{x_3^3 =0\}$ and $P^{012} = P^{01} \cup \{x_3^3 =1\}$, we add this two sub-problems in T then we delete $P^{01}$ from T. The T list will be T= $\{ P^{02}, P^{011}, P^{012}\}$.

We chose $P^{02}$ to solved, we get ($x_1^1$= 0 ,$x_1^2$=0 ,$x_1^3$= 0 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0.4 ,$x_3^1$= 0 ,$x_3^2$= 1 ,$x_3^3$= 0.6) we get 2489,6 This solution isn't feasible because it doesn't satisfy the valid inequalities follows:

(14) $x_2^2 + x_2^3 + 2x_2^1 \leq 2$

(16) $x_2^1 + x_2^3 \leq 1$

So we add these valid inequalities (14),(16) for $R^1$. Then we solve it again, we get ($x_1^1$= 0 ,$x_1^2$= 0 ,$x_1^3$= 0,2 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 1 ,$x_3^3$= 0.7) which equal to 2406,6 This solution doesn't violate any valid inequality but it's not full. We make a branch in variable of $x_3^3$ which has value of branching, so we get two sub-problems $P^{021} = P^{02} \cup \{x_3^3=0\}$ and $P^{022} = P^{02} \cup \{x_3^3=1\}$, we add this two sub-problems in T then we delete $P^{02}$ from T. The T list will be T=$\{ P^{011}, P^{012}, P^{021}, P^{022}\}$.

We chose $P^{011}$ to solved ($x_1^1$= 0 ,$x_1^2$= 0.8 ,$x_1^3$= 1 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 0 ,$x_3^3$= 0 ) we get 2320,4 This solution doesn't violate any valid inequality but it's not full. We make a branch in variable of $x_1^2$ which has value of branching, so we get two sub-problems $P^{0111} = P^{011} \cup \{x_1^2 =0\}$ and $P^{0112} = P^{011} \cup \{x_1^2 =1\}$, we add this two sub-problems in T then we delete $P^{011}$ from T. The T list will be T=$\{P^{012}, P^{021}, P^{022}, P^{0111}, P^{0112}\}$.

We chose $P^{012}$ to solved ($x_1^1$= 0 ,$x_1^2$= 1 ,$x_1^3$= 0 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 0 ,$x_3^3$= 1) we get 2488 This solution is full and it doesn't violate any valid inequality so we up data UB=2488 Then we delete $P^{012}$, so T will be T={$P^{021}$, $P^{022}$,$P^{0111}$,$P^{0112}$}.

We chose $P^{021}$ to solved ($x_1^1$= 0 ,$x_1^2$= 0 ,$x_1^3$= 1 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 1 ,$x_3^3$= 0) we get 2488 This solution is full and it doesn't violate any valid inequality so we delete $P^{021}$, so T will be T={$P^{022}$,$P^{0111}$,$P^{0112}$}.

We chose $P^{022}$ to solved ($x_1^1$= 0 ,$x_1^2$= 0 ,$x_1^3$= 0 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 1 ,$x_3^3$= 1) we get 2490 This solution is full and it doesn't violate any valid inequality Then we delete $P^{022}$ because it doesn't improve the upper bound, T will be T={$P^{0111}$,$P^{0112}$}.

We solved $P^{0111}$ ($x_1^1$= 0 ,$x_1^2$= 0 ,$x_1^3$= 1 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 1 ,$x_3^3$= 0) we get 2488 This solution is full and it doesn't violate any valid inequality Then we delete $P^{0111}$ because it doesn't improve the upper bound, T will be T={$P^{0112}$}.

We solved $P^{0112}$ ($x_1^1$= 0 ,$x_1^2$= 1 ,$x_1^3$= 0.9 ,$x_2^1$= 1 ,$x_2^2$= 0 ,$x_2^3$= 0 ,$x_3^1$= 0 ,$x_3^2$= 0 ,$x_3^3$= 0) we get 2403.2 This solution doesn't violate any valid inequality but it's not full. We make a branch in variable of $x_1^3$ which has value of branching, so we get two sub-problems $P^{01121}$ = $P^{0112}$∪ {$x_1^3$ =0} and $P^{01122}$ = $P^{0112}$∪ {$x_1^3$ =1}, we add this two sub-problems in T then we delete $P^{0112}$ from T. The T list will be T={$P^{01121}$,$P^{01122}$}

We solved $P^{01121}$ results are ($x_1^1$= 0 ,$x_1^2$= 1 ,$x_1^3$= 0 ,$x_2^1$= 1 ,$x_2^2$= 0,$x_2^3$= 0 ,$x_3^1$=0 ,$x_3^2$= 0 ,$x_3^3$= 1) we get 2488 This solution is full and it doesn't violate any valid inequality Then we delete $P^{01121}$ because it doesn't improve the upper bound, T will be T={$P^{01122}$}.

We solved $P^{01122}$ ($x_1^1$= 0 ,$x_1^2$= 1 ,$x_1^3$= 1 ,$x_2^1$= 1 ,$x_2^2$= 0,$x_2^3$= 0 ,$x_3^1$=0 ,$x_3^2$= 0 ,$x_3^3$= 0) we get 2486 This solution is full and it doesn't violate any valid inequality so we up data UB=2486 Then we delete $P^{01112}$, so T will be T=∅ so the end of algorithm.

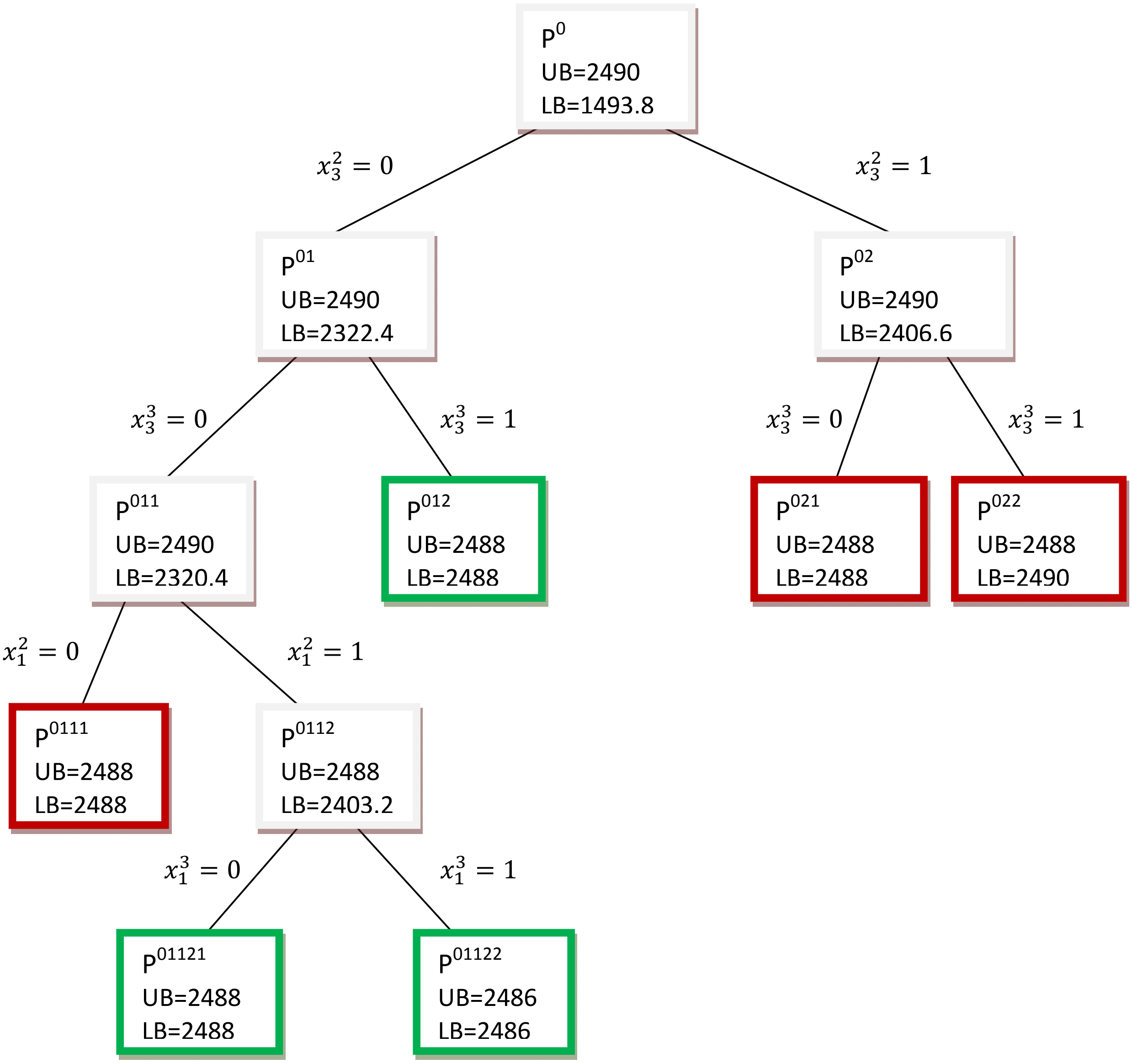



**Figure C.2** The search tree of example

The search tree of this example is in **Figure C.2**. the red nodes are those who are leagues and the green nodes are permit to update the upper bond in which one of them which has the minimal result is the final solution , in our example we have $P^{01122}$ $(m_2^1, m_1^2, m_1^3)$=2486 so we said that C1 assigns to P2 , C2 and C3 assigns to P1.

**ملخص :** الهدف من هذا العمل هو تحقيق برنامج للعثور على أفضل تنظيم لتخزين الحاويات، والتي تهدف أيضا إلى تحسين سعة التخزين. إنه يقدم قدرا كبيرا من الخدمة و يساهم في تسهيل عملية التخزين، و من بين الأساليب المختلفة لحل مشاكل التخزين في الميناء فقد إخترنا احد الخوارزمية المتبعة في حل مشاكل التحسين التوافقي و هي "فرع وقص" لحل هذه المشكلة. البيانات هي: مساحة رصيف الميناء، مساحة ساحة التخزين والحاويات وأنواعها. النتائج المتحصل عليها هي: عدد فتحات الحرة وموقع الحاويات في محطات الميناء وفي ساحة التخزين. .و نتيجة عملنا هذا تتجسد في برنامج " البحث عن خطة التخزين الأمثل لمحطة الحاويات مع أفضل تنظيم " الذي قمنا ببرمجته باستعمال لغة البرمجة فيزيال بازيك.

**كلمات المفاتيح :**

محطات الميناء، التحسين التوافقي ، فرع وقطع، الحاويات، رصيف الميناء ، ساحة التخزين ، فيزيال بازيك.

***Abstract :*** *The objective of this work is the realization of a software to find the best organization of containers storage, which also aims to optimize the storage capacity. It presents a great deal of work on port terminals, and different methods of solving storage problems. A combinatorial optimization algorithm "Branch and Cut" has been used to solve this problem. The data are: surface of quay studied the surface of the storage yard, containers and their types. The results obtained are: the number of free slots and the location of containers in dock and in yard. A contribution in this work is a software named "Search For Optimal storage plan Of Container Terminal With A Best Organization" produced by the Visual Basic language.*

***Keywords :*** *Port Terminals, Combinatorial Optimization, Branch And Cut, Containers, Quay, Yard, Visual Basic.*

***Résumé :*** *L'objectif de ce travail est la réalisation d'un logiciel qui permet de trouver la meilleure organisation de stockage des conteneurs, qui vise par ailleurs à optimiser la capacité de stockage. On présente une grande partie du travail sur les terminaux portuaires, et les différentes méthodes de résolution des problèmes de stockage. Un algorithme d'optimisation combinatoire « Coupe et Branchement » a été utilisé pour résoudre ce problème. Les donnés sont : la surface de quai étudié, la surface de la cour du stockage, les conteneurs et leurs types. Les résultats obtenus sont : le nombre de fentes libre et l'emplacement des conteneurs dans le quai et dans la cour du stockage. Une contribution dans ce travail est un logiciel nommé « Rechercher d'un plan de stockage optimale du terminal à conteneurs avec une meilleure Organisation» réalisé par le langage de Visual Basic.*

***Mots-clés :*** *Terminaux Portuaires, Optimisation Combinatoire, Coupe et Branchement, Conteneurs, Quai, Cour du Stockage, Visual Basic.*