2013 2nd International Conference on Systems and Computer Science (ICSCS)
Villeneuve d'Ascq, France, August 26-27, 2013

A Heuristic for Re-Marshalling Unbound and Outbound Containers

I. Ayachi(1),(2) , R. Kammarti (1),(2) , M. Ksouri (2), P. Borne (1)

(1) LAGIS
Ecole Centrale de Lille, Scientific city - BP 48 - 59651 Villeneuve d'Ascq Cedex - France

(2) LACS
Ecole Nationale des Ingénieurs de Tunis – BP 37, Le Belvédère, 1002 Tunis - Tunisia

ayachiimen@gmail.com , kammarti.ryan@planet.tn , Mekki.Ksouri@insat.rnu.tn, pierre.borne@ec-lille.fr

Abstract **- A container terminal is a complex set of logistics systems with dynamic and random interactions. Information related to the container flow is often uncertain. This modification is due to inaccuracies in the arrival, departure dates of ships and arrival times of customers' trucks. Indeed, the container storage problem becomes a dynamic problem which was the subject of this work.**

In this paper, a heuristic method for re-marhaling outbound and inbound containers is presented. With this method, we can overcome the uncertainties of departure, arrival time of container and improve the container storage planning. This problem is studied with different container types (regular, open side, open top, tank, empty, refrigerated).The objective is to find an optimal container storage plan which respects the departure dates of container and minimize their re-handling operations at their departure time. Some experimental results are presented and discussed to verify the performance of the proposed approach.

Keywords-component: re-marshalling, container storage, transport , heuristic, logistics

I. INTRODUCTION

The workload in a container terminal for a specific period of time can be measured by the number of containers handled during this period. It is the sum of two quantities: the containers unloaded from ships, temporarily stored in the storage areas and unbound containers, arrived in trucks and stored in external storage space.

According to Steenken and his colleagues [1], in European terminals, 30-40% of export containers arrive at the terminal and lack of accurate data of the respective ship, destination port, the container weight - data that are needed to make an effective decision on containers storage.

Even after their arrival, the ship and the unloaded port can be modified by the shipping line. The situation is even worse for import containers, unloaded from ships since in less than 10-15% of cases, the land transportation mode is known when unloading ship.

These uncertainties are usually due to:

- Incomplete information on the initial state (weight, destination ...)
- The dwell time of containers is unknown in advance. Thus, the arrival and departure dates of ships depend on weather uncertainties; the arrival of customers' trucks at the terminal gates is subject to uncertain traffic conditions on the roads.
- The arrival of containers at the port is sometimes unexpected (a ship falling and requires unloading)
- The uncertain results of port operations

The operating environment in a container terminal changes over time. Indeed, the static container storage problem becomes a dynamic problem. Policy storage containers must be able to cope with these environmental changes and that is the subject of our work.

In this paper a heuristic approach is proposed to give a plan to relocate the unbound and outbound containers stored in the port in order to minimize the re-handling operations at their departure time. This algorithm speed up the customers' trucks loading and the ship loading even if there were uncertain and unforeseen arrivals or departures containers after the execution of the initial storage plan. Another challenge in this work is that it considers different containers types (regular, open side, open top, tank, empty and refrigerated). Although, it represents the real word case, this problem was not well studied since many storage constraints appear linked to the diversity type.

This paper is divided into six sections. Following this introduction, we define the dynamic container storage problem and we present works who studied. The mathematical formulation will be described in the third section. Then, we detail the proposed heuristic to solve the dynamic CSP. In the fifth section, some experiments and results are presented and discussed to verify the performance of the proposed heuristic. Finally, section 6 covers our conclusion.

II. LITERATURE REVIEW

The workload in a container terminal for a specific period of time can be measured by the number of containers handled during this period.
It is the sum of two quantities: the containers unloaded from ships, temporarily stored in the storage areas and unbound containers, arrived in trucks and stored in external storage space.

According to Steenken and his colleagues [1], in European terminals, 30-40% of export containers arrive at the terminal and lack of accurate data of the respective ship, destination port, the container weight - data that are needed to make an effective decision on containers storage.

Even after their arrival, the ship and the unloaded port can be modified by the shipping line. The situation is even worse for import containers, unloaded from ships since in less than 10-15% of cases, the land transportation mode is known when unloading ship.

These uncertainties are usually due to:

- Incomplete information on the initial state (weight, destination ...)
- The dwell time of containers is unknown in advance. Thus, the arrival and departure dates of ships depend on weather uncertainties; the arrival of customers' trucks at the terminal gates is subject to uncertain traffic conditions on the roads.
- The arrival of containers at the port is sometimes unexpected (a ship falling and requires unloading)
- The uncertain results of port operations

The operating environment in a container terminal changes over time. Indeed, the static container storage problem becomes a dynamic problem. Policy storage containers must be able to cope with these environmental changes and that is the subject of our work.

The dynamic container storage problem can be defined as a variant of dynamic Bin Packing Problem.

It was introduced by Coffman ,Garey and Johnson [2].

This problem assumes that the arrival and departure dates of items are random and uncertain.

Modern planners have developed techniques to find efficient storage plans when many uncertainty sources arise, but these plans require a lot of time [3]

Other researchers have used "on line" search algorithms to solve the dynamic CSP. This method treats any new application or modification when it appears.

In the work [4], the researchers propose an online search algorithm that dynamically adjusts the storage containers policy represented by a vector of criteria values (container type, Stack height, the load distribution of the stacking cranes).

This work focuses on the problem of determining the positions of storage containers in automated container terminal (ACT). The proposed algorithm is used to create and evaluate several storage strategies within a relatively short time. Note that previous work of Kim and his colleagues [5] has already provided some preliminary results on the dynamic adjustment of stacking policies, but this work [4] considers more factors to determine the position stacking and provides a more comprehensive analysis.

Another alternative is proposed to solve the dynamic CSP is the use of Re-marshalling algorithms. These methods modify the organization of containers stored in the yard during downtime cranes and this to satisfy certain criteria like improving the productivity of the loading operation, reducing the container rehandling and minimizing the working time of cranes.

To achieve better performance, terminal operators can arrange their containers stacks in an order that fits the next loading sequence. The rearrangement operation is a good solution to solve problems caused by an improperly containers stacked. However, it takes cost and must be performed during idle time which is usually short and subsequently we cannot rearrange all containers.

Lee and Chao propose a heuristic algorithm to rearrange the export containers stored in blocks at port to accelerate the loading of ships. Given the initial bay layout of container, the proposed method determines a plan to pre-marshal the containers in such an order that it fits the loading sequence of ships. The proposed consists of a neighborhood search process, an integer programming model, and three minor sub-routines. The test results demonstrate the performance of this approach. [6]

To maintain balance and stability of a ship, the heaviest containers are loaded before the lighter. To achieve effective load, terminal operators generally classify export containers entering by weight. They group the containers having the same weight in the same stack. However, the estimated weight of the containers is not available at their arrival time in port. What makes a stack often includes containers belonging to different groups of weight. This combination causes additional rehandling movement during the ship loading process.

In the paper [7] ans [8], researchers proposed a simulated annealing algorithm to derive a more effective strategy stacking for incoming containers and having uncertain weight. The simulation results showed the performance of this approach. In addition, improvements may be made to this method in the presence of a best estimate of the weight and applying the learning process.

Park and his colleagues determined algorithms to to derive a rearrangement plan in an automated container terminal. The new approach divides the problem into two sub-problems: one for determining where to move the containers and the other to determine the priority of movement. [9]

Jiang and colleagues present a simulation model to dynamically imitate the container storage operation and their rehandling process [10]

The animation of the model allows the user to evaluate and compare in real time the different rules or algorithms reshuffle.

After this literature review, we can note that the majority of studies have solved the dynamic CSP for a single type of container and researchers have treated the problem for import or export containers. However, this does not stand the problem under its real-life statement as there are multiple container types that should be considered, (refrigerated, open side, empty, dry, open top and tank). Also, import and export containers are stored in the same place therefore; it is more effective to treat the problem for all containers

In this paper, we propose a heuristic solution for re-marshalling unbound and outbound containers having various types.

This work is an extension of the study of static CSP presented in our papers [11] and [12]. We have proposed in these work solutions to static CSP using genetic algorithms and Harmony search. The CSP is solved for a single and multiple container types.

III. PROBLEM FORMULATION

The dynamic container storage problem is an extension of the static CSP. In fact, you must satisfy the uncertain and unexpected arrivals and departures of containers after the preparation and implementation of a storage plan.

The planner must satisfy the new and the former events while optimizing the selected criteria and respecting the constraints.

In general, we can say that a dynamic objective function depends on time. The global optimum of the function that we seek to optimize may change over time. Indeed, we must add the concept of time in the problem definition.

A container terminal works every day (365 days per year). We decompose the day into several periods and each of them; we solve the PSC in a static manner.

A period is created until the arrival of a set of containers at the port (import or export) requiring their temporary storage. Dynamic events (unexpected arrived of containers, departure date modified container ...) happening in a period will be treated in the next time interval. To resolve this uncertainty and improve the organization of containers, we will apply a re-marshalling method to the storage plan.

This heuristic is applied at the beginning of each period. It consists on modifying the organization of the containers in order to satisfy new information, concerning their departure dates and the unloading sequence.

A. *Objective function*

The objective of this dynamic CSP is to determine movement sequences of containers that improve their unloading (departure) process. The new organization of container minimize the number of rehandling operations that will occur when we rearrange containers and when they leave the storage area.

The parameters used in this formulation are defined as follows:

- i : container index, $i = 1, \ldots, N_c$
- b : index of the storage block ; $b = 1, \ldots, N_{Block}$
- $N_{Block} = N_{stock\_reg} + N_{stock\_refrig}$: the number of available blocks
- d_i : Departure date of container i
- N_D : number of container type
- $N_c(D)$: number of container having the type D, where D = 1, …, 6. It denotes the type of container.

$$D = \begin{cases} 1 & \text{if it is a dry container} \\ 2 & \text{if it an empty container} \\ 3 & \text{if it is an open top container} \\ 4 & \text{if it is a platform container} \\ 5 & \text{if it is a tank container} \\ 6 & \text{if it is a refrigerat ed container} \end{cases}$$

- $C_{i,D}(x, y, z, b)$: decision variable

$$C_{i,D}(x,y,z,b) = \begin{cases} 1 & \text{if thereis a containeri of typeD in theposition} \\ & x, y, z \text{ at theblock b} \\ 0 & \text{otherwise} \end{cases}$$

- N_{cm} : number of containers stored in the blocks (initial state) with departure dates changed.

The following equation expresses the objective function:

$$\text{Min } \left(\sum_{cm=1}^{N_{cm}} m_{cm} + \sum_{D=1}^{N_D} \sum_{i=}^{Nc(D)} \sum_{b=1}^{N_{Bloc}} m_{i,b}\, C_{i,D}(x, y, z, b) \right) \quad (1)$$

with:

- m_{cm} : cumulative number of rehandling mouvement at the n[th] period to rearrange the storage area.
- $m_{i,b}$: the minimum number of container rehandlings to extract the container I in the block b.

A re-handling is a container movement made in order to permit access to another, or to improve the overall stowage arrangement, and is considered a product of poor planning.

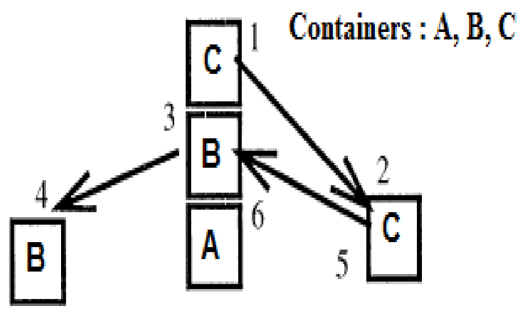

Figure 1. Rehandling operations

B. *Contraints*

The difference between types of containers often results from the difference between goods and items inside them such as foods, chemical substances, liquids. Therefore, many storage constraints appeared, related to the container type's requirements. For instance, refrigerated containers must be allocated to the blocks equipped by the power point, empty containers must not be settled under a full container, on an open top container, we cannot place a container at the top; tank container must be placed on each other, etc[12]

The following equations described these storage constraints

$$C_{i,r}(x,y,z,b) - C_{j,D}(x,y,z+1,b) = 1, \quad (2)$$
$$\forall\, i \in [1..N_c(r)],\ \forall\, r = 3,4,\ \forall\, j \in [1,..N_c(D)]$$

$$C_{i,4}(x,y,z,b) - \sum_{m=x+1}^{n3} C_{j,r}(m,y,z,b) = 1, \quad (3)$$
$$\forall\, i \in [1..N_c(4)], \forall\, j \in [1..N_c(D)], \forall\, r \in [1,..N_D]$$

$$C_{i,5}(x,y,z,b) - C_{j,r}(x,y,z+1,b) = 1, \quad (4)$$
$$\forall\, i \in [1..N_c(5)], \forall j \in [1..N_c(r)], \forall r \in \{1,2,3,4,6\}$$

$$C_{i,6}(x,y,z,b) = \begin{cases} 1, \text{ Si the block is reefer} \\ 0, \text{ Otherwise} \end{cases} \quad (5)$$

The constraint 2 ensures that an open top container or an open side container can not have another container above.
The constraint 3 indicates that there aren't any containers at the open side of container type 4 (open side container.
The constraint 4 indicates that tank containers must be placed on each other.
The constraint 5 suggests that a reefer container must be allocated to the blocks equipped by the power point.

IV. THE RE-MARSHALLING HEURISTIC

The heuristic that we adopt to remedy to uncertainties related especially to departure dates of containers consists on decomposing the day into a specified number of working times.

During each period (working time), a reorganization of the initial storage plan is established if there are changes in departure dates of containers already in the storage area.

Any changes occurred in the period N will be processed at the beginning of the next period (N +1).

The reorganization of the initial state of the storage blocks can be described as follows:

For each container whose departure date has changed (denoted cm), we try, if we can assign it to a new location that will be beneficial in the unloading sequence.

Thus, a movement of container is permitted only if it reduces mis-overlays (mis-overlay is the situation where two containers are stacked in the wrong order), which is very useful to help performing re-marshalling operation in practice.

To do this, we permute the container "cm" using all possible combinations (with all containers having the same type) and each time we calculates the following three parameters :

- R_{init} : represents the number of rehandling movements which will be occurred in the 2 stack containing the two containers before their permutation
- N_{Permut} : the number of operations needed to reshuffle two containers
- R_{final} = represents the number of rehandling movements which will be occurred in the 2 stack containing the two containers after their permutation

Then the gain, denoted G, generated by this operation is calculated as follow:

$$G = (N_{permut} + R_{final}) - R_{init} \quad (6)$$

Thus, we keep all permutations having a negative gain since it is beneficial swap to unloading process (it minimizes the number of parasites movements) and we choose the best permutation. (Having N_{permut}+ R_{final} minimum).

If we did not find a negative gain, cm container keeps its current position.
The following algorithm describes the re-marshalling process.

Re-marshalling algorithm
Begin

```
For  i =1 to Ncm
// cm is the container that his departure date is modified
// Tc : Type of container cm
   Gain = 0
  For each container cm having the type Tc do
        For each container c with type Tc do
          Calculate Rinit (c, cm)
          Calculate Npermut( c, cm)
          Calculate Rfinal (c,cm)
          G = (Npermut + Rfinal) - Rinit
               If (G<Gain) then
                     Gain = G
                      Bestpermut = Conteneur courant  (c)
               End if
       End For
   End For
End For
```

End

Figure 2. Re-marshalling algorithm

Below, we present a diagram illustrating the rearrangement process.. In this example, the departure date of container n° 9 with type 1 has changed from 4 to 10.

By applying the re-marshalling heuristic, a switches between the container n° 9 and the container n° 4 with the same type having a departure date = 5 is beneficial to unloading plan. R_{init} = 10, N_{Permut}, =5 , R_{final} = 3.

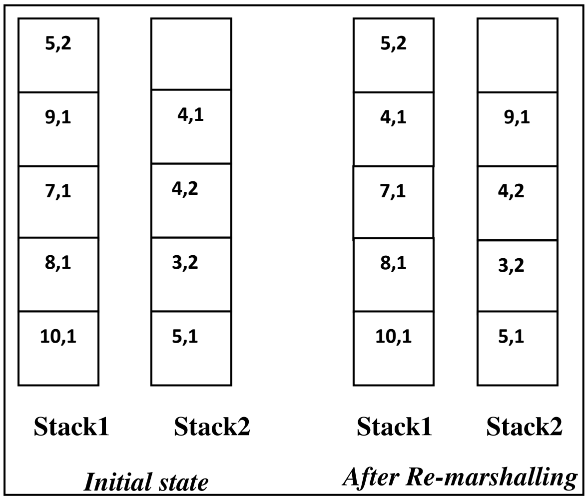

Figure 3. Heuristic for containers Re-marshalling

V. EXPERIMENTAL RESULTS

In this section, we provide computational examples to verify the performance of the heuristic proposed in this paper. The re-marshalling solution is applied to solve 4 instances of Dynamic CSP.

For each test instance, we present the containers that are already in the storage area and their departure dates changed (advanced, delayed). We specify their previous and new departures. Instances of the problem are presented in the table I

The storage area is composed of 3 blocks for refrigerated containers and 7 general blocks with $n_1 = n_2 = n_3 = 5$.

TABLE I. INSTANCES OF DYNAMIC CSP

Instance 1			Instance 2			Instance 3			Instance 4		
N°cont	date_P	date_N	N°cont	date_P	date_N	N°cont	date_P	date_N	N°cont	date_P	date_N
219	19	40	219	19	40	219	19	40	219	19	40
623	23	90	623	23	90	623	23	90	623	23	90
648	48	80	648	48	80	648	48	80	648	48	80
226	26	70	226	26	70	226	26	70	226	26	70
641	41	56	641	41	56	641	41	56	641	41	56
			719	19	42	719	19	42	719	19	42
			724	24	45	724	24	45	724	24	45
			620	20	42	620	20	42	620	20	42
			732	32	50	732	32	50	732	32	50
			62	1	80	62	1	80	62	1	80
						731	31	51	731	31	51
						60	60	85	60	60	85
						29	29	71	29	29	71
						18	18	82	18	18	82
						3	3	50	3	3	50
									36	36	45
									152	52	70
									126	26	75
									33	33	53
									148	48	75

Initially, there are in the storage area:

- 70 Dry containers.
- 30 Open top containers.
- 30 Platform container
- 70 refrigerated containers.
- 50 Tanks
- 70 empty containers.

As it can be seen in fig 4, the use of the re-marshalling method generated largely better results. In fact, it reduces the rehandling operations that consume time and minimize the port productivity.

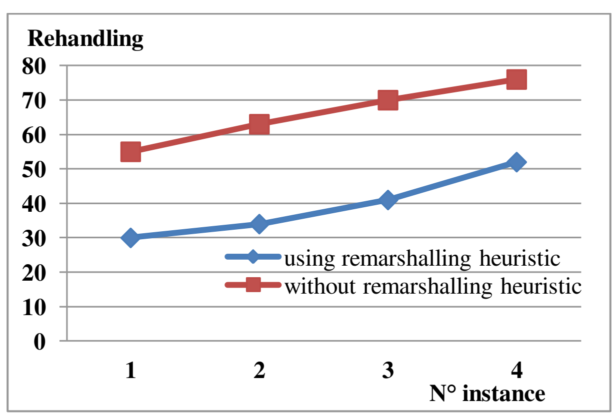

Figure 4. Performance of the re-marshalling heuristic

In addition to the solutions quality generated by the proposed approach, our re-marshalling heuristic needs a reasonable execution time which can be illustrated in the fig 5.

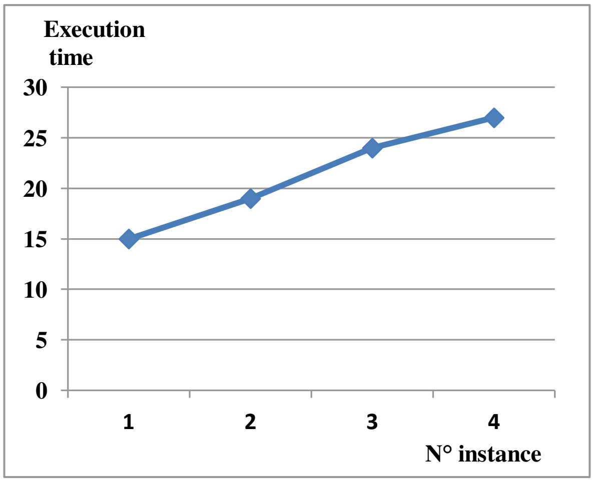

Figure 5. Execution time of the re-marshalling heuristic

VI. CONCLUSION

In this study, a heuristic for re-marshalling import and export container is applied to solve the dynamic container storage problem.. It considers diversity type of containers (regular, open side, open top, tank, empty and refrigerated). As a result many storage constraints appear and the problem becomes more complex.

Despite this difficult, the proposed method generated good results in a reasonable execution time. Experimental study confirms these and shows the effectiveness of this method.

The objective of our approach was to find an optimal rearrangement plan of containers which minimize their re-handling operations at their departure times.

REFERENCES

[1]: Steenken D., Voß S., and Stahlbock R. 2004, Container terminal operation and operations research – a classification and literature review. OR Spectrum (2004) 26: 3–49

[2]: E. G. Coffman, Jr., M. R. Garey, and D. S. Johnson. Dynamic bin packing. SIAM Journal on Computing, 12(2):227–258, 1983.

[3]: Blythe Jim, Decision-Theoretic Planning, AI Magazine, Summer 1999

[4]: Park T., Choe R., Kim, Y.H., Ryu, K.R., 2011, Dynamic adjustment of container stacking policy in an automated container terminal, Int. J. Production Economics, pp 385–392

[5]: Kim, Y.H., Park T., Ryu, K.R., 2007, Dynamic weight adjustment for developing a stacking policy for automated container terminals. Paper presented at The International Conference on Intelligent Manufacturing and Logistics Systems (IML 2007), Kitakyushu, Japan, 26–28 February 2007.

[6] : Yusin Lee , Shih-Liang Chao, A neighborhood search heuristic for pre-marshalling export containers, Production, Manufacturing and Logistics, European Journal of Operational Research, Volume 196, Issue 2, 16 July 2009, Pages 468-475.

[7]: Kang, J., Ryu, K.R., Kim, K.H., 2006a. Determination of storage locations for incoming containers of uncertain weight. In: Ali, M., Dapoigny, R. (Eds.). Advances in applied artificial intelligence. Proceedings of the 19th Interna- tional Conference on Industrial, Engineering and other Applications of Applied Intelligent Systems, IEA/AIE 2006, vol. 4031. Annecy, France, June 27–30, Springer. pp. 1159–1168

[8] : Kang, J., Ryu, K.R., Kim, K.H., 2006 b, Deriving stacking strategies for export containers with uncertain weight information. Journal of Intelligent Manufacturing , volume 17, pp 399–410

[9]: Park, K., Park, T., Ryu, K.R., 2009, Planning for remarshaling in an automated container terminal using cooperative coevolutionary algorithm, Proceeding of the 2009 ACM Symposium on Applied Computing, SAC'09. ACM, New York, NY, pp. 1098–1105

[10]: Wei Jiang, Yun Dong, Lixin Tang , Simulation study on reshuffling problem in logistics operations of a container terminal yard, Service Operations, Logistics, and Informatics (SOLI), 2011 IEEE International Conference.

[11]: I. Ayachi, R. Kammarti, M. Ksouri et P. Borne, 2011, Application of Harmony search to Container Storage Location, 2011 IEEE International Conference on Systems, Man, and Cybernetics (IEEE SMC 2011).

[12]: I. Ayachi, R. Kammarti et P. Borne, 2012, " Genetic algorithm to solve Container Storage Problem for a single and various types ", chapitre n°19 in the book « Computational intelligence applications in industrial engineering » , pages 411-433, édité par Prof. Cengiz Kahraman, publié par Atlantis & Springer .