

# Single quay crane and multiple yard trucks scheduling problem with integration of reach-stacker cranes at port of Tripoli-Lebanon

A. SKAF<sup>1,2</sup>, S. LAMROUS<sup>1</sup>, Z. HAMMOUDAN<sup>2</sup>, M. A. MANIER<sup>1</sup>

<sup>1</sup> Univ. Bourgogne Franche-Comté, FEMTO-ST Institute/CNRS, (UTBM), Belfort, 90010, France

<sup>2</sup> Univ. Libano-Française (ULF), Tripoli, Lebanon

(ali.skaf@utbm.fr, sid.lamrous@utbm.fr, zakaria.hammoudan@gmail.com, marie-ange.manier@utbm.fr)

**Abstract**—This paper addresses the scheduling problem in port of Tripoli-Lebanon for a single quay crane with multiple yard trucks, all containers that will be unloaded from the vessel are in the same bay. The objective is to reduce the completion time of all containers from the vessel to their store location, we used a mixed integer linear programming and a dynamic programming algorithm to solve the problem. Finally, we have compared and validated our results on real instances from the port.

**Keywords** – optimization, quay crane, yard truck, scheduling problem, MILP, dynamic programming, case study

## I. INTRODUCTION AND RELATED WORK

Quay cranes are used for unloading/loading containers from/to containers vessel(s), while yard trucks are used to transport containers from the quay cranes to the store location and vice versa. The reach stackers cranes are used to unload/load containers from the yard trucks/store location to the store location/yard trucks.

Usually in maritime terminals, several quay cranes and yard trucks work together, but when all containers to be unloaded from the vessel are located in the same bay in the vessel, only one quay crane is allowed to unload containers because more than one quay crane cannot work in the same bay to prevent the crossing between each other.

First the quay crane unloads a container from the containers vessel and puts it on the yard truck which transports the unloaded container to the reach-stacker crane station, then a reach-stacker crane unloads the container from the yard truck and puts it in the store location, finally the yard truck goes back in order to transport another container unloaded by the quay crane.

Fig.1. and Fig.2. describe all the operations of unloading containers from the vessel to the store location.

In our previous studies, [9] Skaf et al. (2018) proposed two exact methods to solve the quay crane scheduling problem at port of Tripoli-Lebanon. Later [10] Skaf et al. (2019) proposed a genetic algorithm to solve the same previous problem. We used a genetic algorithm because in larges instances the two exacts methods have difficulties to obtain results, and this genetic algorithm gave us a near-optimal solutions quickly.

Many researchers addressed the quay crane and yard trucks scheduling problem in the literature: [4] Shang Jing (2010) discussed how to decide the yard truck and how to allocate

containers to yard trucks. He proposed a mixed integer programming model and a two-phase heuristic algorithm to obtain a near-optimal solution. After that [1] Chen et al. (2012) studied the relation between handling containers and yard truck transportation in container terminal simultaneously.

[2] Dkhil el al. (2013) proposed three mathematical models and an exact resolution of quay crane and yard trucks scheduling in an automated container terminal. They aimed to minimize the makespan and the number of required trucks. Thereafter, they solved the previous planning problem and proposed a new bi-objective model determining the minimal storage time. They developed a hybrid genetic algorithm and a tabu search. [11] Tang et al. (2013) addressed the quay crane and yard truck scheduling problem at a container terminal. They proposed a mixed-integer linear programming model and a Particle Swarm Optimization (PSO) algorithm to minimize the makespan.

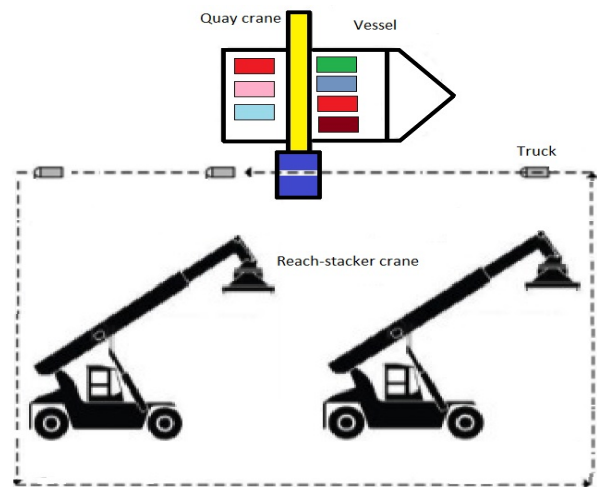


Fig. 1. Operation in the port

[3] Homayouni et al. (2013) presented a genetic algorithm (GA) to solve the integrated scheduling of quay cranes, automated guided vehicles and handling platforms. [8] Niu et al. (2014) proposed a particle swarm optimization (PSO) to obtain the optimal solution of the integrated yard truck scheduling and storage allocation problem. [5] Kaveshgar et al. (2014) developed a mixed integer programming for quay cranes and yard trucks scheduling and proposed a genetic

algorithm (GA) combined with a greedy search algorithm to solve the problem. [14] Zhen et al. (2016) studied the optimization problem on quay crane and yard truck scheduling in container terminals and proposed a mixed-integer programming model and genetic algorithm (GA).

[13] Zhang et al. (2016) proposed a strategy for the storage to improve the efficiency of yard operations and transportation with quay crane double cycling. They developed an approach based on cycle-time models to evaluate the performance. Furthermore [7] Liu et al. (2016) presented a model of the quay cranes and trucks scheduling problem in the container terminal. They developed a Hybrid intelligence swarm algorithm with a particle swarm optimization algorithm (PSO). [12] Vahdani et al. (2018) aimed to integrate the assignment of quay cranes and internal truck assignment among them. They proposed a bi-objective optimization model to solve the problem. Moreover [6] Liang et al. (2018) studied the quay crane scheduling problem (QCSP) with taking in consideration the configuration of quay cranes and handling sequence of quay cranes. They compared their objective to the results approach found in literature and took the optimal number of quay cranes into account to solve the model.

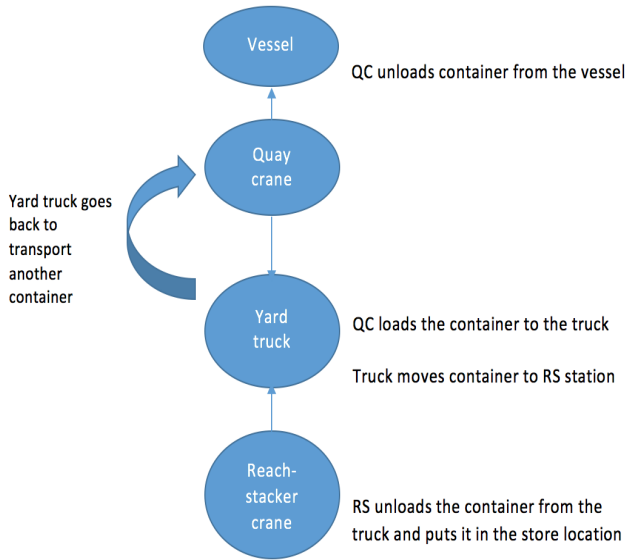


Fig. 2. Quay crane and yard truck scheduling problem

This paper proposes a formulation of the problem in the form of a mixed-integer programming model solved by CPLEX, and a dynamic programming algorithm coded with JAVA, to obtain the optimal solutions. After that we compare the obtained results from both exacts methods. We tested our methods for large and small instances and we found that dynamic programming is faster than the mixed-integer linear programming.

Later on in this paper, section II proposes a mixed-integer linear programming (MILP). Section III proposes an explanation of the dynamic programming algorithm. Section IV provides the experimental results obtained with the MILP and the dynamic programming. Finally, section V

gives some perspectives and finishes by the conclusion.

## II. MATHEMATICAL FORMULATION

### A. Assumptions

- Number of containers is between 0 and C+1 with 0 and C+1 are two dummy containers for the origin and final state respectively.
- We do not consider the number of reach-stacker cranes, and we suppose that there is always a reach-stacker crane available to transfer the container from the yard truck to the store location.

### B. Notations

- $C$  : number of containers, indexed  $i$  and  $j$ ;
- $T$  : number of yard trucks, indexed  $t$ ;
- $w_i$  : time needed to unload container  $i$  by the quay crane;
- $t_{ij}$  : setup time for the quay crane to unload container  $j$  after container  $i$ ;
- $\lambda_i$  : time needed for the yard truck to transport container  $i$  to the store location (it is same to the returning time for the truck to transport another container from the quay crane);
- $r$  : time needed to unload a container from the yard truck by the reach-stacker crane;
- $M$  : big integer.

### C. Decision variables

#### 1) Boolean variables:

- $A_{ij} \begin{cases} = 1 & \text{if quay crane unloads container } j \text{ directly} \\ & \text{after unloading container } i \\ = 0 & \text{otherwise, } (A_{ii} = 0, \forall i \in 1 \dots C) \end{cases}$
- $B_{it} \begin{cases} = 1 & \text{if yard truck } t \text{ transports container } i \\ = 0 & \text{otherwise} \end{cases}$
- $X_{ijt} \begin{cases} = 1 & \text{if yard truck } t \text{ transports container } j \\ & \text{directly after transporting container } i \\ = 0 & \text{otherwise, } (X_{iit} = 0, \forall i \in 1 \dots C, \forall t \in 1 \dots T) \end{cases}$

#### 2) Float variables:

- $s_i$  : the time when the quay crane starts unloading container  $i$ ;
- $s'_i$  : the time when the yard truck starts transporting container  $i$  ( $s'_0 = 0$ );
- $c_i$  : the time when container  $i$  is ready to be transported by the yard truck;
- $cp_i$  : the completion time of container  $i$ ;
- $C_{max}$  : the completion time of all containers.

### D. Modeling

The following is the mixed integer linear programming which models the quay crane and yard truck scheduling problem :

#### Objective

$$\text{minimize } C_{max} \quad (1)$$

The objective of this paper is to minimize the completion time of all containers allocated to the quay crane, from the vessel to the store location.

**Subject to**

$$\sum_{i=0}^C A_{ij} = 1 \quad \forall j \in \{1 \dots C+1\} \quad (2)$$

$$\sum_{j=1}^{C+1} A_{ij} = 1 \quad \forall i \in \{0 \dots C\} \quad (3)$$

$$\sum_{i=1}^T B_{it} = 1 \quad \forall i \in \{1 \dots C\} \quad (4)$$

$$\sum_{j=1}^{C+1} X_{0jt} = 1 \quad \forall t \in \{1 \dots T\} \quad (5)$$

$$\sum_{i=0}^C X_{i,C+1,t} = 1 \quad \forall t \in \{1 \dots T\} \quad (6)$$

$$\sum_{j=1}^{C+1} X_{ijt} = B_{it} \quad \forall i \in \{1 \dots C\}, \forall t \in \{1 \dots T\} \quad (7)$$

$$\sum_{i=0}^C X_{ijt} = B_{jt} \quad \forall j \in \{1 \dots C\}, \forall t \in \{1 \dots T\} \quad (8)$$

$$X_{ijt} + X_{jit} \leq 1 \quad \forall i, j \in \{1 \dots C\}, \forall t \in \{1 \dots T\} \quad (9)$$

$$X_{ijt} + X_{jit} \geq B_{it} + B_{jt} - 1 \quad \forall i, j \in \{1 \dots C\}, \forall t \in \{1 \dots T\} \quad (10)$$

$$s_j + (1 - A_{ij}) * M \geq t_{ij} + s'_i \quad \forall i \in \{0 \dots C\}, \forall j \in \{1 \dots C\} \quad (11)$$

$$c_i = s_i + w_i \quad \forall i \in \{1 \dots C\} \quad (12)$$

$$s'_i \geq c_i \quad \forall i \in \{1 \dots C\} \quad (13)$$

$$s'_j + (1 - X_{ijt}) * M \geq cp_i + \lambda_i \quad \forall i \in \{1 \dots C\}, \forall j \in \{1 \dots C\}, \forall t \in \{1 \dots T\}, \quad (14)$$

$$s'_j + (1 - X_{0jt}) * M \geq 0 \quad \forall j \in \{1 \dots C\}, \forall t \in \{1 \dots T\} \quad (15)$$

$$s'_i + \lambda_i + r \leq cp_i \quad \forall i \in \{1 \dots C\} \quad (16)$$

$$C_{max} \geq cp_i \quad \forall i \in \{1 \dots C\} \quad (17)$$

$$A_{ij} = [0, 1] \quad \forall i \in \{0 \dots C\}, \quad \forall j \in \{1 \dots C+1\} \quad (18)$$

$$B_{it} = [0, 1] \quad \forall i \in \{1 \dots C\}, \quad \forall t \in \{1 \dots T\} \quad (19)$$

$$X_{ijt} = [0, 1] \quad \forall i \in \{0 \dots C\}, \quad \forall j \in \{1 \dots C+1\}, \forall t \in \{1 \dots T\} \quad (20)$$

Constraints (2) and (3) force the assignment of all containers to make the quay crane unloading sequence.

Constraint (4) guarantees that each yard truck can transport one and only one container.

Constraints (5) and (6) force the first and last container of each yard truck.

Constraints (7) and (8) ensure that there is one succeeding container after and one preceding container before, to the allocated container unloaded by the yard truck t.

Constraints (9) and (10) relate the containers affected to the same yard truck.

Constraint (11) shows that if container j is immediately unloaded by the quay crane after container i, the quay crane needs a setup time before starting to unload container j when the yard truck transports container i.

Constraint (12) indicate that the completion time for unloading a container by the quay crane is equal to the starting time of unloading the container in addition to the processing time for the container by the quay crane.

Constraint (13) shows that a container can be transported only after being unloaded by the quay crane.

Constraints (14) and (15) make a relation between the starting time and the completion time to the adjacent containers for the same yard truck.

Constraint (16) shows that a container experiences the stacking and transportation times between the container's completion time and the starting of the transportation.

Constraint (17) calculate the completion time of the last container.

Constraints (18), (19) and (20) define the domain of the decision variables.

### III. DYNAMIC PROGRAMMING ALGORITHM

In this section, we propose a dynamic programming algorithm for the quay crane and yard trucks scheduling problem named QCYTSPDPA.

Dynamic programming is a mathematical and programming methods for combinatorial optimization problems, its main advantage is obtaining the optimal solutions instead of the near optimal solutions. It was developed by Bellman in 1950.

#### A. Flowchart

As shown in Fig.3., a flowchart describes the dynamic programming algorithm. First, we generate all parameters such as the number of containers that will be unloaded and the number of yard trucks... second step is getting all possible choices of containers order, per example if we have 4 containers to be unloaded from the vessel, and let us say that the best order with the optimal solution is 3142, this order means that the first container to be unloaded is number 3, then number 1, then number 4 and finally number 2. Third step is getting all assignments of containers pairs to the trucks, per example 132 means that truck number 2 transport container number 3 after transporting container number 1. After that, for each choice and assignment, we calculate the

beginning time for unloading and transporting each container respectively by the quay crane and the yard truck. Then we compute the completion time for all containers and we must make sure that there are no still assignments. Finally, the optimal solution is the minimal completion time of all assignments.

The most important thing in this algorithm is to find all possible choices for containers order and their assignment to the yard trucks.

### B. Algorithm description

---

#### Algorithm 1 Dynamic programming algorithm

---

```

1: Generate_variables();
2: Containers_order();
3: Containers_pairs();
4: for all assignments do
5:   Calculate_begTime_QC();
6:   Calculate_begTime_YT();
7:   for each container do
8:     CP_Container();
9:   end for
10:  Calculate_CP();
11:  if assignments_nb = 0 then
12:    Get_minimal_CP();
13:    End;
14:  else
15:    go_to_next_assignment;
16:  end if
17: end for

```

---

As shown above, Algorithm 1 shows the full description of how the QCYTSPDPA works. The algorithm follows the same steps of the flowchart seen in the previous section.

#### Lines explanation :

- Line 1 : generating all variables such as number of yard trucks, containers...
- Line 2 : generating all the containers order choices.
- Line 3 : generating all the containers pairs assignments affected to the yard trucks.
- Line 4 to line 17 : for all assignments, we calculate the beginning time for the quay crane to unload each container (line 5), then calculate the beginning time for the yard truck to transport each container (line 6), after that calculate the completion time of each container (lines 7, 8 and 9) and calculate the completion time of all containers for each assignment (line 10). Finally, if there are no more assignments, get the minimum completion time between all the assignments which is the optimal value (lines 11, 12 and 13), else continue to the next assignment (lines 14, 15 and 16).

## IV. EXPERIMENTAL RESULTS

Experimental instances were solved using CPLEX and dynamic programming (DP) and executed using a MacBook Pro 2,7 GHz Intel Core i5, 8 GB RAM.

In this section, we present results comparison between the

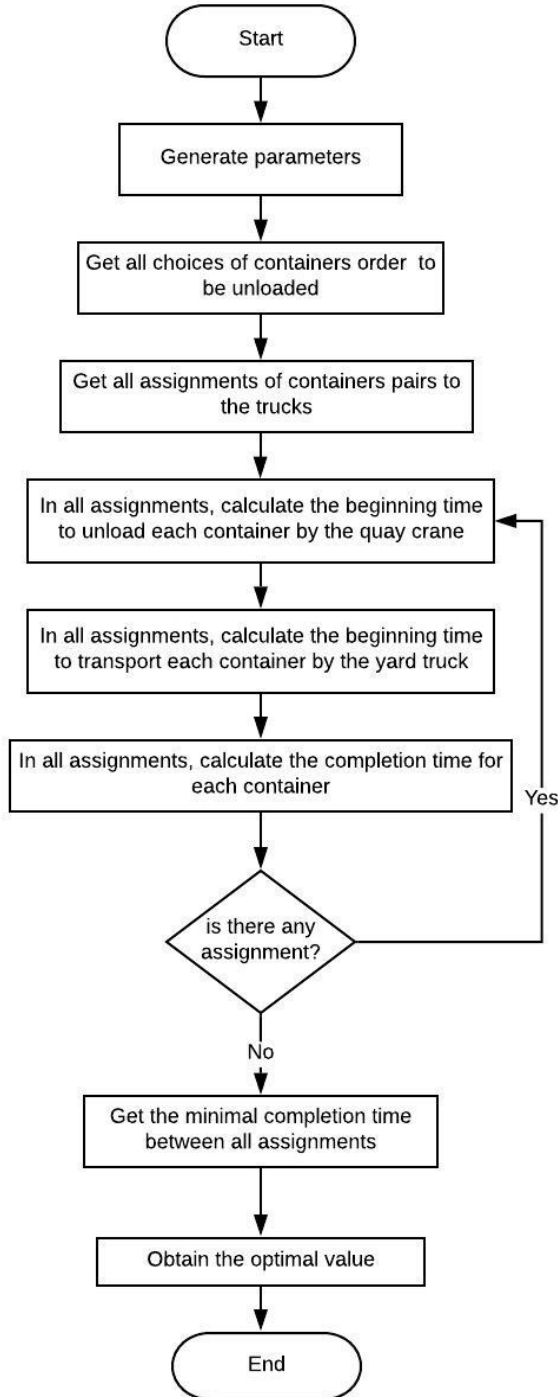


Fig. 3. Dynamic programming algorithm

dynamic programming algorithm and CPLEX, a brief explanation of how to obtain results and finally, a comparison with the effective results in the port of Tripoli-Lebanon results.

#### A. Results CPLEX vs Dynamic programming

Table I shows results for CPLEX and dynamic programming (DP) from 4 containers and 2 yard trucks to 18 containers and 3 yard trucks.

The needed time for the quay crane to unload a container is randomly generated between 60 and 80 time units. The setup time for the quay crane to unload container after another one is randomly generated between 8 and 10 time units. The needed time for the yard truck to transport a container to the store location is randomly generated between 30 and 40 time units. Finally, the needed time for the reach-stacker crane to unload a container from the yard truck is randomly generated between 8 and 10 time units.

TABLE I  
EXPERIMENTAL RESULTS

No.	(C x T)	Makespan		CPU Time		GAP (%)
		CPLEX	DP	CPLEX (s)	DP (s)	
1	4 x 1	397	397	< 1	< 1	0
2	4 x 2	383	383	< 1	< 1	0
3	5 x 1	487	487	< 1	< 1	0
4	5 x 2	468	468	< 1	< 1	0
5	6 x 1	577	577	1.01	< 1	≈ 0
6	6 x 2	555	555	1.23	< 1	≈ 0
7	7 x 1	667	667	5.74	1.45	74.74
8	7 x 2	641	641	6.55	1.52	76.80
9	7 x 3	603	603	22.57	1.53	93.22
10	8 x 1	757	757	9.81	4.14	57.80
11	8 x 2	726	726	42.77	5.16	87.93
12	8 x 3	697	697	252.36	5.67	97.75
13	10 x 2	898	898	101.73	29.23	71.27
14	10 x 3	N.A	891	N.A	513.04	-
15	11 x 2	N.A	1210	N.A	979.43	-
16	11 x 3	N.A	991	N.A	2031.18	-
17	12 x 2	N.A	2312	N.A	2518.59	-
18	12 x 3	N.A	1987	N.A	3977.16	-
19	14 x 2	N.A	2991	N.A	6941.04	-
20	14 x 3	N.A	N.A	N.A	N.A	-

- (C x T) means (Containers x Yard trucks).
- GAP =  $((\text{CPLEX CPU} - \text{DP CPU})/\text{CPLEX CPU}) * 100$ .
- N.A : No results after interrupting execution.

Both CPLEX and dynamic programming give the same makespan, but for instance 14, we operate moves of 10 containers with 3 yard trucks on CPLEX for 9 hours without giving any results while dynamic programming for this instance gives result.

So, CPLEX is limited to solve the small instances and dynamic programming comparing with CPLEX is faster and can be used to solves bigger size instances.

We execute our algorithm for instance 20 with 14 containers and 3 yard trucks. Dynamic programming was unable to give result after 5 hours of execution, so that why on our future

work we will try to propose a heuristic method to obtain near optimal solution more quickly.

#### B. Process explanation and completion time calculation

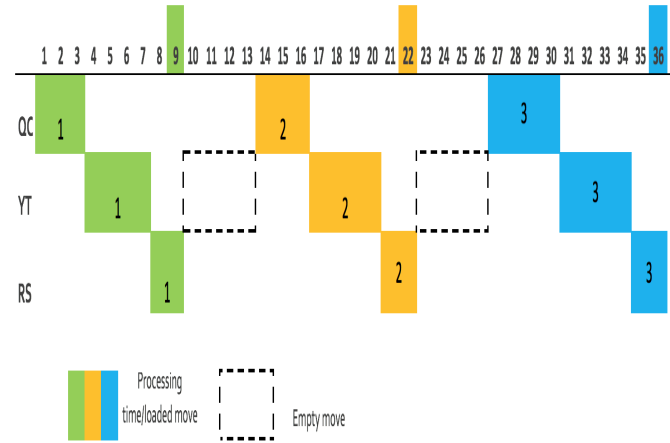


Fig. 4. Process for single yard truck

Let us suppose that we have one-yard truck and 3 containers to be unloaded, and the unloading times of these containers by the quay crane are respectively  $w_1 = w_2 = 3, w_3 = 4$ . The time needed for the yard truck to transport a container to the store location is  $\lambda = 4$  (same for all containers). The truck returning time with empty move is  $\lambda' = 4$ . Finally, the unloading time of a container by the reach-stacker crane is  $r = 2$  (same for all containers).

Fig.4. shows the full process of this operation.

Completion time of container 1 :  $w_1 + \lambda + r = 3 + 4 + 2 = 9$ .

Completion time of container 2 :  $w_2 + \lambda + r = 3 + 4 + 2 = 9$ .

Completion time of container 3 :  $w_3 + \lambda + r = 4 + 4 + 2 = 10$ .

Completion time of all containers :

$$cp_1 + \lambda' + cp_2 + \lambda' + cp_3 = 9 + 4 + 9 + 4 + 10 = 36.$$

#### C. Comparison between paper results with port of Tripoli-Lebanon results (real experiments)

Table II compare our results with previous results from the port of Tripoli-Lebanon. These results are given in the same conditions and values of the port (time needed to unload container by the quay crane - setup time for the quay crane to unload container after the next one - time needed for the yard truck to transport container to the store location - time needed to unload a container from the yard truck by the reach-stacker crane).

TABLE II  
QCYTSP BENCHMARKS

N.	Containers number	Yard trucks number	Port results (s)	Paper results (s)	GAP (%)
1	5	2	590	472	20
2	6	2	737	574	22.11
3	6	3	518	422	18.5
4	8	1	1360	1051	22.72
5	8	2	1154	913	20.9
6	8	3	901	723	19.75

As shown in Table II, in first instance with 5 containers and 2 yard trucks, the completion time for these containers in the port of Tripoli-Lebanon is about 590 seconds while for this instance we obtain about 472 seconds. So our model aims to improve the completion time of containers by an average 20%.

The Gap(%) is high maybe due to a technical problem or to the containers unloading order or to a waiting time.

$GAP(\%) = ((\text{port result} - \text{paper result})/\text{port result}) * 100$ .

## V. CONCLUSION

This paper studied a scheduling problem in port for the case of a single quay crane and multiple yard trucks. We proposed a MILP model and a dynamic programming algorithm to solve the problem. We compared the obtained results to the results of the port of Tripoli-Lebanon and showed that we obtained better performance. In future work, this model will be used to validate heuristic method.

## REFERENCES

- [1] L. Chen, A. Langevin, and Z. Lu, "Integrated scheduling of crane handling and truck transportation in a maritime container terminal," *European Journal of Operational Research* 225 (2013) 142152, 2012.
- [2] H. Dkhil, A. Yassine, and H. Chabchoub, "Optimization of container handling systems in automated maritime terminal," *Springer-Verlag Berlin Heidelberg*, 2013.
- [3] S. M. Homayouni, S. H. Tang, and O. Motlagh, "A genetic algorithm for optimization of integrated scheduling of cranes, vehicles, and storage platforms at automated container terminals," *Journal of Computational and Applied Mathematics* 270 (2014) 545556, 2013.
- [4] S. Jing, "A heuristic algorithm for the integrated yard truck scheduling in container terminal with twin 40-foot quay crane," *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering (CMCE)*, 2010.
- [5] N. Kaveshgar and N. Huynh, "Integrated quay crane and yard truck scheduling for unloading inbound containers," *Int. Production Economics* 159 (2015)168177, 2014.
- [6] C. Liang, L. Fan, D. Xu, Y. Ding, and M. Gen, "Research on coupling scheduling of quay crane dispatch and configuration in the container terminal," *Computers and Industrial Engineering* 125 (2018) 649657, 2018.
- [7] Y. Liu and T. Liu, "The hybrid intelligence swam algorithm for berth-quay cranes and trucks scheduling optimization problem," *15th Intl Conf. on Cognitive and Cognitive Computing*, 2016.
- [8] B. Niu, T. Xie, Q. Duan, and L. Tan, "Particle swarm optimization for integrated yard truck scheduling and storage allocation problem," *IEEE Congress on Evolutionary Computation (CEC)*, 2014.
- [9] A. SKAF, S. LAMROUS, Z. HAMMOUDAN, and M. A. MANIER, "Exact method for single vessel and multiple quay cranes to solve scheduling problem at port of tripoli-lebanon," *International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2018.
- [10] A. SKAF, S. LAMROUS, Z. HAMMOUDAN, and M. A. MANIER, "Genetic algorithm to optimize unloading of large containers vessel in port of tripoli-lebanon," *International Conference on Control, Decision and Information Technologies (CODIT)*, 2019.
- [11] L. Tang, J. Zhao, and J. Liu, "Modeling and solution of the joint quay crane and truck scheduling problem," *European Journal of Operational Research* 236 (2014) 978990, 2013.
- [12] B. Vahdani, F. Mansour, M. Soltani, and D. Veysmoradi, "Bi-objective optimization for integrating quay crane and internal truck assignment with challenges of trucks sharing," *Knowledge-Based Systems* 163 (2019) 675692, 2018.
- [13] Z. Y. Xiaoju Zhang, Qingcheng Zeng, "Modeling the mixed storage strategy for quay crane double cycling in container terminals," *Transportation Research Part E* 94 (2016) 171187, 2016.
- [14] L. Zhen, S. Yu, ShuaianWang, and Z. Sun, "Scheduling quay cranes and yard trucks for unloading operations in container ports," *Springer Science+Business Media New York*, 2016.