



HAL
open science

Integrated quay crane and yard truck scheduling problem at port of Tripoli-Lebanon

Ali Skaf, Sid Ahmed Lamrous, Zakaria Hammoudan, Marie-Ange Manier

► **To cite this version:**

Ali Skaf, Sid Ahmed Lamrous, Zakaria Hammoudan, Marie-Ange Manier. Integrated quay crane and yard truck scheduling problem at port of Tripoli-Lebanon. *Computers & Industrial Engineering*, 2021, 159, pp.107448 (11). 10.1016/j.cie.2021.107448 . hal-03551924

HAL Id: hal-03551924

<https://hal.science/hal-03551924v1>

Submitted on 2 Aug 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Integrated quay crane and yard truck scheduling problem at port of Tripoli-Lebanon

Ali Skaf^a, Sid Lamrous^b, Zakaria Hammoudan^c, Marie-Ange Manier^b

^a*Univ. Franche-Comté, FEMTO-ST Institute, CNRS, IUT de Belfort-Montbéliard, Belfort, 90010, France*

^b*Univ. Bourgogne Franche-Comté, FEMTO-ST Institute, CNRS, UTBM, Belfort, 90010, France*

^c*Jinan University (JU), Tripoli, Lebanon*

Abstract

The scheduling problem is one of the most important operation in maritime ports. In this study we solved the scheduling problem for a single quay crane and multiple yard trucks in port of Tripoli-Lebanon. In a previous study we proposed two exact methods for this problem but we faced a problem for large instances in CPU time. For this reason, in this paper we developed a heuristic method (genetic algorithm) to obtain near optimal solution with an acceptable CPU time. The main objective of this paper is to minimize the completion time of all containers from the container vessel to its storage location and to have good results compared with real results in the port of Tripoli-Lebanon.

Keywords: Optimization, Quay crane, Yard truck, Scheduling problem, MILP, Exact Enumerative algorithm, Genetic algorithm, Case study

1. Introduction and related work

The transport of goods is a necessary activity in the modern world economy. Except for raw materials, these goods (hydrocarbons, coal, cement, cereals,

building materials, etc.) are shipped in containers traveling by ship, train, truck or even by plane. These containers are huge standardized boxes whose maximum mass should not exceed 40 tons and can measure up to 40 feet long. They allow goods to be transported without unpacking them from a supplier to their customer and are thus very useful. In addition, the largest transport ships, the 'post-panamax' can load up to 13,000 containers.

Handling resources are essential to load and unload containers on the ports: they are quay cranes and container tippers. There are of course several types of quay cranes and rockers depending on the needs on the ports. Thus, for container tippers, the ports can propose a loading system vertically or horizontally. In the vertical level, they can be inclined at an angle between -10° and 90° . The rockers can handle containers between 20 and 40 feet long and are designed by expert manufacturers.

Quay cranes (QC) and containers tippers are driven by trained professionals. They take care of emptying the containers on the port or loading them on a cargo. A team of about 10 people is needed, not to mention the inspectors who oversee the operations to make sure everything goes well. These inspections include the determination of quantity and weight, verification of packaging and labeling, sealing of goods, temperature control, traceability of containers, operational control of loading/unloading as well as writing reports and transaction tracking documents.

A yard truck (YT) is a type of truck, driven by one person and used to transport container from the quay cranes station to the storage location and vice versa. It can transport only one container at a time, while a reach-stacker crane (RS) is a type of crane also driven by one trained professional and used

to unload container from YT to the storage location, it can unload only one container at a time. To unload containers from YT and store them in the port, yard cranes are often used instead of RS.

There are several types of scheduling problem, we find the quay crane scheduling problem (QCSP), the yard truck scheduling problem (YTSP), the integrated quay crane and yard truck scheduling problem (QCYTSP) and the integrated quay crane and automated vehicle scheduling problem(QCAVGSP). In this paper, we address the QCYTSP at port of Tripoli-Lebanon, because in our previous studies we considered the QCSP only (without integration of yard trucks) and due to the simplicity of the YTSP comparing to the QCYTSP. Finally, the QCAVGSP is used only in the port that have automated vehicle in the terminals.

Figure 1 describes the full maritime activity of unloading or loading containers at the mentioned port.

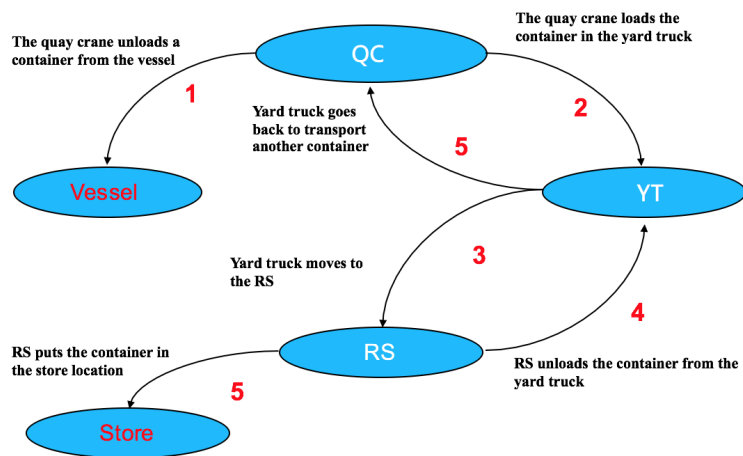


Figure 1: Full considered process

Firstly the vessel arrives at the containers terminal with a number of con-

tainers to be unloaded by the QC and transported by YT to the storage location. Each container will be unloaded by a QC (arrow 1 in Figure 1) which loads it to a YT (arrow 2 in Figure 1). Then the YT transports it to the RS (arrow 3 in Figure 1), and finally the RS unloads the container (arrow 4 in Figure 1) and puts it in the storage location (arrow 5 in Figure 1). Finally, all customers can take their containers from the storage location. In case we have only one YT, after unloading a container by the QC from the vessel and putting it on the YT to be transported, the QC will unload the second container and wait for the return of this YT.

Figure 2 illustrates the full process for unloading containers by the quay crane, from the vessel to the store location.

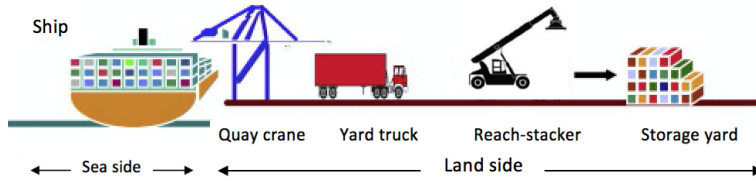


Figure 2: Quay crane and yard truck scheduling problem

In our previous papers, Skaf et al. [1] suggested two methods to solve the quay crane scheduling problem at port of Tripoli-Lebanon, to minimize the total completion time for all containers from the vessel to the storage location. Later Skaf et al. [2] proposed for the same problem a genetic algorithm to obtain near-optimal solutions in an acceptable CPU time.

After that, Skaf et al. [3] proposed two exact methods for single quay crane and multiple yard trucks scheduling problem at the same port, to obtain the best completion time as an optimal solution.

The quay crane scheduling problem was addressed by some researchers in

the literature:

Daganzo [4] studied the scheduling problem for quay crane and multiple vessels. He divided each vessel into bays and each bay contains a set of containers. Its objective is to minimize the cost of delay, with two solving methods, exact and approximate. Moreover, Diabat and Theodorou [9] proposed a new formulation for the quay crane scheduling problem and its assignment. They developed a genetic algorithm for solving this problem. After that, Al-Dhaheeri and Diabat [13] aimed to minimize the completion time of all containers to be unloaded by a set of quay cranes with determination of the unloading operations sequence. They proposed a formulation based on mixed-integer programming for the problem.

Liu et al. [16] aimed at minimizing the turn-around time for a vessel with solving the scheduling problem for only two quay cranes. They developed an integrated approximation algorithm to solve the problem. Then, Haoyuan and Qi [20] presented a new model to solve the quay crane scheduling problem and then optimizing the delay time of all vessels, and Xiazhong et al. [21] aimed at minimizing the completion time of containers and traveling times for the yard trucks. They presented a mixed-integer linear programming (MILP) and a Tabu search for solving the problem.

Liang et al. [22] aimed at studying the quay crane scheduling problem (QCSP) with the handling sequence and the configuration of quay cranes. They compared their results to existing ones in the literature.

Finally, Boysen et al. [15] introduced a scheme for the quay crane scheduling problem without taking in consideration the quay cranes non-crossing constraints. The scheme was applied for the existing literature taxonomy and

for future research identification.

The quay crane and yard truck scheduling problem was addressed by some researchers in the literature:

Jing [5] proposed a solution for the allocation of containers to the yard trucks. This solution consists of a mixed integer programming and a two-phase algorithm. After that Chen et al. [6] proposed a relation between the unloading of containers and the transport of yard trucks simultaneously. Then, Xue et al. [8] aimed to examine the quay crane scheduling, yard truck scheduling and the store location assignment for unloading containers. They proposed a mixed-integer programming, an ant colony optimization and a greedy search algorithm and they conduct numerical experiments to test these algorithms performances. Moreover, Tang et al. [12] addressed the scheduling problem at container terminal for both quay crane and yard truck. They proposed a mixed-integer linear programming (MILP) model and developed a particle swarm optimization algorithm (PSO) to obtain the minimal completion time. Then, Kaveshgar and Huynh [14] proposed a mixed-integer programming and a genetic algorithm (GA) with a greedy search to solve the quay crane and yard truck scheduling problem.

In the same year, Liu and Liu [17] produced a new model for the quay cranes and yard trucks scheduling problem in a maritime terminal. They proposed a particle swarm optimization algorithm (PSO) and a hybrid algorithm, Zhang et al. [18] produced a new strategy to store the containers and improve the yard operations with quay crane (double cycling) efficiency. They proposed an approach method for cycle-time for the performance evaluating, and Zhen et al. [19] studied the optimization problem for quay crane and yard truck

scheduling and they proposed a mixed-integer programming (MIP) and a genetic algorithm (GA).

Finally, Vahdani et al. [23] aimed to integrate the assignment of quay cranes in terminals and yard truck sharing assignment among them, and proposed a bi-objective optimization model for solving the problem.

The yard truck scheduling problem was also addressed by Niu et al. [11] who aimed to find a solution for the yard truck scheduling with the containers allocation problem. They proposed a particle swarm optimization (PSO) to solve the problem.

The quay crane and automated vehicle scheduling problem was also addressed by Dkhil et al. [7] who aimed to minimize the completion time of all containers and the required number of yard trucks. They used mixed-integer linear programming, tabu search and hybrid genetic algorithm to solve the problem and Homayouni et al. [10] who developed a genetic algorithm (GA) to solve the quay crane scheduling with integration of automated guided vehicles (AGV).

Compared to what the researchers studied above, in this paper we introduce some notions such as the repositioning time needed for the quay cranes to unload a container after a previous one. Moreover, we consider the positioning of the containers and a single quay crane. Also, we do not consider automated vehicle for transporting the containers, however we consider the yard trucks that will be driven by a specialist driver, and all yard trucks cannot exceed each other. These are our main constraints encountered in the port of Tripoli-Lebanon, and which do not jointly exist in the previous studies mentioned above. In order to present our contribution among the

literature, Table 1 summarizes the papers listed above.

Table 1: Similarities and differences with near problems

Ref.	Problem	Constraints					Resolution methods
		Repos.		Position	YT	Zcrane	
		QC	YT				
[4]	QCSP			✓			MILP
[9]	QCSP	✓		✓			MILP,GA
[13]	QCSP	✓		✓			MIP
[16]	QCSP	✓		✓			Approx. algorithm
[20]	QCSP	✓		✓			PSO,SA
[21]	QCSP	✓		✓		YC	MILP,PSO
[22]	QCSP	✓		✓		YC	Two MILP
[5]	QCYTSP		✓		✓	YC	MILP,CWA
[6]	QCYTSP		✓		✓	YC	MILP
[8]	QCYTSP	✓			✓		MIP,ACO
[12]	QCYTSP	✓		✓	✓		MILP,PSO
[14]	QCYTSP	✓		✓	✓	YC	MILP,GA
[17]	QCYTSP			✓	✓		MILP,PAO-AFSA
[18]	QCYTSP		✓		✓	YC	MILP
[19]	QCYTSP	✓	✓	✓	✓		MILP,PSO
[23]	QCYTSP			✓	✓	YC	MILP, MNSGA-II
[11]	YTSP		✓		✓	YC	MILP,PSO
[7]	QCAVGSP	✓		✓		YC	MILP,GA
[10]	QCAVGSP	✓	✓			YC	GA
This study	QCYTSP	✓	✓	✓	✓	RS	MILP, EEA, GA

* **Repos. (QC repositioning time):** Time needed for the quay crane to move the next bay for unloading a container after putting a previous one on the yard truck.

* **Repos. (YT repositioning time):** When a yard truck transports a container from/to the vessel to/from the storage location, there exists an empty moving or a waited time to transport another container.

* **Containers position:** Each vessel can be divided into many bays and each bay contains a number of containers, or all containers can be numerated to know the loading/unloading sequence.

* **Zcrane:** There are several resources in any maritime port for grouping containers in the storage location, e.g reach stacker crane, yard cranes... Also, it can be used to load/unload containers to/from the yard truck, who can pass under it.

In this study we propose three solving methods for the scheduling problem in the port of Tripoli-Lebanon which are a mixed-integer programming solved with CPLEX, an exact enumerative algorithm developed with java, a genetic algorithm developed with java. We also proposed some equations to find the lower and upper bounds, also developed with java. All results of this study are compared between them for small and large instances, after we compared our solutions with the real results in the port of Tripoli-Lebanon.

The rest of this study is designed as follows: section 2, a mathematical formulation in the form of MILP. Then, section 3 presents the solving methods for our problem, while section 4 reports the experimental results. Section 5 presents the results discussion and finally, section 6 concludes this study and gives some future researches.

2. Mathematical Formulation

2.1. Assumptions

- Single vessel, single quay crane and multiple yard trucks are considered.
- C is the total number of containers, but we created two fictive containers, which represent an origin location (number 0) and the final location (number $C+1$).
- Each quay crane can unload at most one container at a time.
- Each yard truck can transport only one container at a time.
- Each container can be unloaded by at most one quay crane.
- Each container can be transported by at most one yard truck.
- All containers are similar and have the same width and height.
- We do not take into consideration reach-stacker cranes number, we suppose that there is always an available reach-stacker crane to unload containers from the yard trucks.

2.2. Notations

- C Number of containers, indexed i and j
- T Number of yard trucks, indexed t
- w_i Time needed to unload a container i with the quay crane
- t_{ij} Repositioning time for the quay crane to unload container j after container i

- λ_i Time needed for the yard truck to transport container i to the storage location (we suppose that it is same for the return time of 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

2.3. Decision variables

- $A_{ij} \begin{cases} = 1 & \text{if the quay crane unloads container } j \text{ directly after} \\ & \text{unloading container } i \\ = 0 & \text{otherwise, } (A_{ii} = 0, \forall i \in \{1..C\}) \end{cases}$
- $B_{it} \begin{cases} = 1 & \text{if a yard truck } t \text{ transports container } i \\ = 0 & \text{otherwise} \end{cases}$
- $X_{ijt} \begin{cases} = 1 & \text{if a yard truck } t \text{ transports container } j \text{ directly after} \\ & \text{transporting container } i \\ = 0 & \text{otherwise, } (X_{iit} = 0, \forall i \in \{1..C\}, \forall t \in \{1..T\}) \end{cases}$
- s_i The time when the quay crane starts to unload container i
- s'_i The time when a yard truck starts to transport container i ($s'_0 = 0$). It also corresponds to the time when the quay crane finishes to load container i on a yard truck.

- z_i The time when container i is ready to be transported by the yard truck
- c_i The completion time of container i from the vessel to the entry of storage location
- C_{max} The completion time of all containers

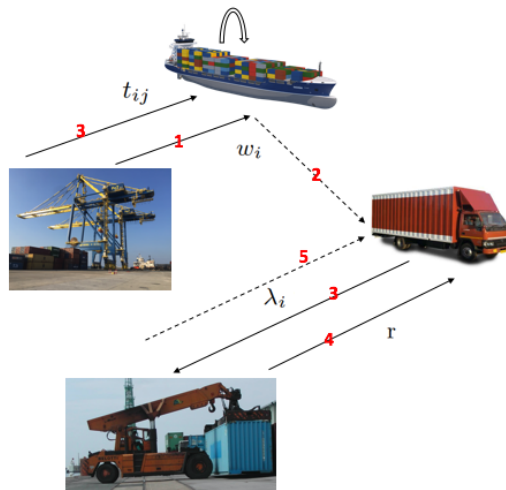


Figure 3: Illustration for the notations

2.4. Mathematical model

The following succinctly presents the mixed integer linear programming for the quay crane and yard truck scheduling, that we have proposed in Skaf et al. [3].

Objective

$$\text{minimize } C_{max} \tag{1}$$

The objective of this problem is to minimize the makespan for dispatching the set of containers allocated to the QC.

Subject to

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

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

Constraints (2) and (3) are used to assign all containers to the quay crane for making the unloading sequence.

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

Constraint (4) confirms that each container should be allocated/transported to/by one and only one yard truck.

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

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

Constraints (5) and (6) are used to find the first container and the last container who will be transported by each yard truck.

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

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

Constraints (7) and (8) ensure that each unloaded container by the yard truck, there is one preceding container and another succeeding one.

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

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

Constraints (9) and (10) find all containers to be transported by the same yard truck.

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

Constraint (11) ensures that there is a repositioning time t_{ij} for the quay crane before starting to unload the container j after the unloading of container i .

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

Constraint (12) calculates the value of a container's unloading makespan by the quay crane.

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

Constraint (13) ensures that each container should be transported by the yard truck only after being unloaded from the vessel by the quay crane.

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

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

Constraints (14) and (15) relate the completion and the starting times for the adjacent containers assigned to the same yard truck.

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

Constraint (16) checks that each container should face the stacking and transport times between the makespan of a container and the transportation.

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

Constraint (17) calculates the last container's completion time.

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

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

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

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

3. Solving methods

After presenting our mixed-integer linear programming model, we first propose some equations to find the lower and upper bounds of the optimal C_{max} . Then, we propose two methods to solve the integrated quay crane and yard truck scheduling problem, which are the exact enumerative algorithm and the genetic algorithm. The exact enumerative algorithm is an exact method to obtain the optimal solutions, while the genetic algorithm is a heuristic method to obtain near optimal solutions.

3.1. Lower and Upper Bounds

In this section, we determine lower and upper bounds of C_{max} , for the quay crane and yard truck scheduling problem. In optimization, the lower bound is an element which is less than or equal to the optimal solution obtained by an exact method, while the upper bound is an element which is greater than

or equal to the optimal solution (Figure 4).

Two lower bounds are expressed by the equations (21) and (22). The first one is based on the containers while the second one is based on the tools. An upper bound is expressed by equation (24).

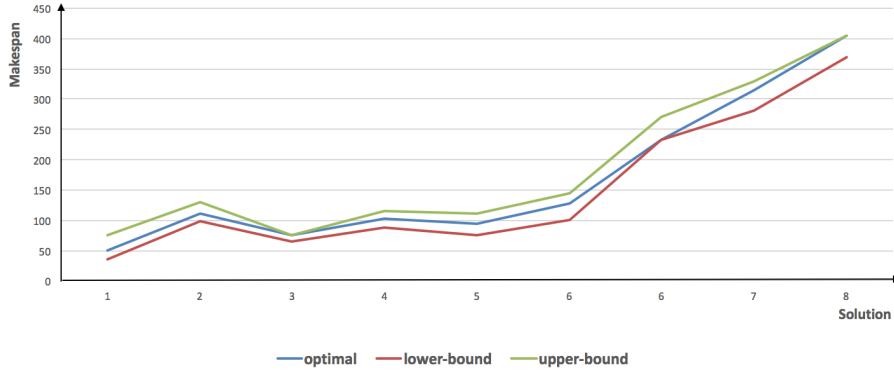


Figure 4: Optimal solutions vs upper bounds vs lower bounds

3.1.1. Lower bounds

The following two equations give us the lower bound feasible solution for the quay crane and yard truck scheduling problem:

$$lb_1 = \max_{i=1..C} (w_i + \min_{j=0..C} t_{ji} + \lambda_i + r) \quad (21)$$

Equation (21) demonstrates the lower bound based on containers. The working time of the quay crane, the transportation of the yard truck, the stacking time in the storage location and the repositioning time are all experienced by every container i . Note that the repositioning time is the minimal required time to set up container i .

$$\begin{aligned}
lb_2 = & \min_{i=1..C} (w_i + \min_{j=0..C} t_{ji}) + \frac{\sum_{i=1}^C (2\lambda_i + r)}{T} \\
& + \frac{1}{T} \sum_{t=1}^{T-1} [M_{i=1..C}^{[t]}(w_i + \min_{j=0..C} t_{ji}) - \min_{i=1..C} (w_i + \min_{j=0..C} t_{ji})] - \min_{i=1..C} \lambda_i
\end{aligned} \tag{22}$$

where $M_{i=1..|C|}^{[t]}(w_i + \min_{j=0..|C|} t_{ji})$ is the $(t + 1)^{th}$ smallest element of a list R whose i^{th} element take the value $w_i + \min_{j=0..C} t_{ji}$.

Let's take an example with $C = 3$, $w_1 = 4$, $w_2 = 3$, $w_3 = 5$, $t_{01} = t_{02} = t_{03} = 2$, $t_{12} = t_{21} = 2$, $t_{13} = t_{31} = 4$ and $t_{23} = t_{32} = 5$. In this case :

$R = ((w_1 + \min(t_{01}, t_{11}, t_{21}, t_{31})); (w_2 + \min(t_{02}, t_{12}, t_{22}, t_{32})); (w_3 + \min(t_{03}, t_{13}, t_{23}, t_{33})))$. This gives $R = (6, 5, 9)$, and therefore $M_R^{[0]} = \min(6, 5, 9) = 5$, $M_R^{[1]} = \min(6, 9) = 6$ and $M_R^{[2]} = 9$.

The first part of the equation (22) represents the minimum time required by the quay crane for a first container to be ready to be transported by a yard truck. This time includes work and repositioning time. The second part of this equation indicates that all containers are transported without preemption by yard trucks. Finally the third part stems from the observation that the second yard truck only works when the second container arrives and so on for all the yard trucks up to container number T . The minimum is then the sum of all the required times for the second, third and other containers that will be ready for transport and assigned to the yard trucks. The first two terms correspond to the lower bound of when the last container is finished and the yard truck returns to the quay crane. In the last term, the return time of the yard truck is subtracted to find the minimum transport time of the container in order to calculate the lower bound of the *makespan*.

Finally, the equation (23) is the lower bound deduced from the two values lb_1 and lb_2

$$lb = \max(lb_1, lb_2) \quad (23)$$

3.1.2. Upper bounds

The following equation give us the upper bound feasible solution for the scheduling problem:

$$ub = \sum_{i=1}^C (w_i + r) + \frac{(2C - 1) \times \max_{i=1..C} \lambda_i}{T} + [C \times \max_{j=0..C}^{i=1..C} t_{ij}] \quad (24)$$

Explanation of equation (24) with a numerical example:

We suppose that we have 1 container and 1 yard truck, then the completion time is : $C_{max} = t_{01} + w_1 + \lambda_1 + r$, with one repositioning time. Let's apply the equation (24): $ub = w_1 + r + ((2 - 1) \times \lambda_1)/1 + t_{01}$, then $C_{max} = ub$, so we obtain $C_{max} \leq ub$.

We suppose now that we have 2 containers and 1 yard truck. The end time of handling is : $C_{max} = t_{01} + w_1 + \max(t_{12} + w_2; \lambda_1 + r + \lambda'_1) + \lambda_2 + r$, with two repositioning times.

Let's apply the equation (24): $ub = w_1 + r + w_2 + r + ((4 - 1) \times \lambda)/1 + 2t$, with $t = \max_{j=0..C}^{i=1..C} t_{ji}$ and $\lambda = \max_{i=1..C} \lambda_i$. So $ub = w_1 + w_2 + 2r + 3\lambda + 2t$.

If the quay crane is the critical resource for transporting the second container,

then $C_{max} = t_{01} + w_1 + t_{12} + w_2 + \lambda_2 + r$. Therefore, as $\forall i, j \in \{1 \dots C\}$, $\lambda \geq \lambda_i$ and $t \geq t_{ji}$, so $C_{max} \leq 2t + w_1 + w_2 + \lambda + r \leq ub$. ub is therefore an upper bound here.

Otherwise, the yard truck is the resource arriving at the latest meeting point, and $C_{max} = t_{01} + w_1 + \lambda_1 + r + \lambda'_1 + \lambda_2 + r$. Therefore $C_{max} \leq t + w_1 + 2r + 3\lambda$. As $w_2 + t \geq 0$, then $C_{max} \leq ub$. ub is therefore an upper bound of the *makespan* in all cases.

3.2. Exact Enumerative Algorithm (EEA)

We propose in this section the exact enumerative algorithm to solve our problem, and it is called EEA.

3.2.1. Algorithm description

Algorithm 1 shows the full description of how the exact enumerative algorithm works from generating parameters ending by obtaining the optimal solution.

Explanation of Algorithm 1:

- Line 1 : first we generate the number of containers and yard trucks, the time needed to unload container by the quay crane, the repositioning time for the quay crane to unload container after a previous one, the time needed for the yard truck to transport container to the storage location, and finally the time needed to unload a container from the yard truck by the reach-stacker crane.
- Line 2 : we generate all possible choices for the containers order. For example if we have 4 containers and if the first order is 1324, this

means that the container number 1 is the first one to be unloaded, then the container number 3, then the container number 2 and finally the container number 4.

Algorithm 1 Exact enumerative algorithm

```
1: generate_all_variables();
2: get_all_containers_order();
3: get_all_containers_pairs();
4: for each assignment do
5:   QC_beginning_time();
6:   YT_transport_time();
7:   for each container do
8:     calculate_completion_time();
9:   end for
10:  calculate_total_completion_time();
11:  if assignment = 0 then
12:    get_minimal_completion_time();
13:  end;
14: else
15:   next_assignment();
16: end if
17: end for
```

- Line 3 : we generate all possible choices for the containers pairs assignments to be transported by the yard trucks. For example if the first order is 321, this means that yard truck number 1 transports container

number 2 after transporting container number 3.

- Line 4 to line 17 : for each assignment, we calculate for the quay crane, the beginning time to unload each container (line 5). Then we calculate for the yard truck, the beginning time to transport each container (line 6). Then we calculate each container's completion time (lines 7, 8 and 9) and for each assignment, we calculate the total completion time of all containers (line 10). Finally, if the number of assignments is over, we set the lowest completion time between all the assignments as the optimal solution (in lines 11, 12 and 13), otherwise we continue to the next assignment (lines 14, 15 and 16).

3.2.2. Numerical examples

In this sub-section, we present two numerical examples for full process of the scheduling. The first one is for 3 containers and 1 yard truck (Figure 5), while the second one is for 3 containers and 2 yard trucks (Figure 6).

The quay crane unloading times of the three containers (w_i) are 3,3 and 4, and the yard truck transporting time for all containers to the store location (λ_i) is 4 (same to the yard truck empty returning (λ')). Finally the reach-stacker crane unloading time of all containers (r) is 2.

We suppose that container 1 is the first one which will be unloaded by the quay crane before container 2 and finally container 3.

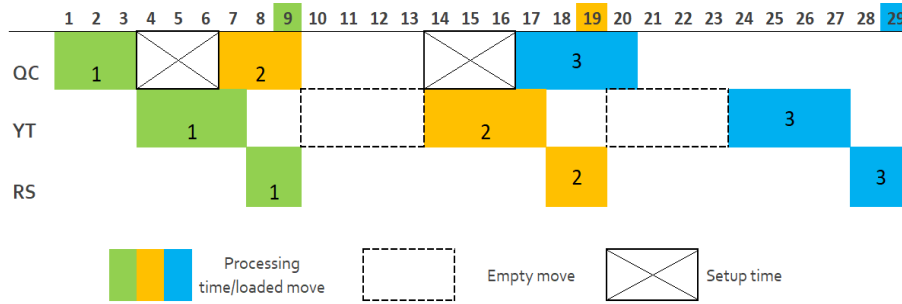


Figure 5: Process for single yard truck

For the example with one yard truck, the evaluation of the scheduling illustrated by the Figure5 gives the following result :

* Container 1 : handling time $ht_1 = w_1 + \lambda_1 + r = 9$

* Container 2 : handling time $ht_2 = w_2 + \lambda_2 + r = 9$

* Container 3 : handling time $ht_3 = w_3 + \lambda_3 + r = 10$

* Total handling time of all containers :

$$C_{max} = w_1 + \max(t_{12} + w_2; \lambda_1 + r + \lambda'_1) + \max(t_{23} + w_3; \lambda_2 + r + \lambda'_2) + \lambda_3 + r = 29.$$

In the figure, we can see that the quay crane repositioning time does not impact the overall evaluation, as it is less than the transport times of the yard truck. In addition, in practice on the quays, more safety constraints are sometimes applied linked to safety and prohibiting a container from remaining suspended above the quay without a yard truck to receive it. In this case, a quay crane will only start a new unloading if a yard truck is already present. Of course this implies additional delays. This so-called security constraint is not taken into account in our models.

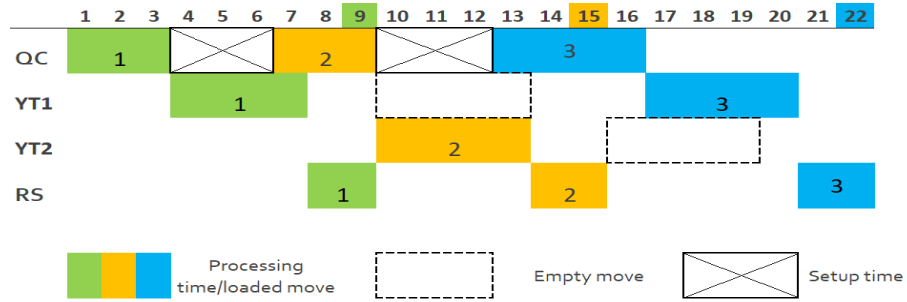


Figure 6: Process for two yard trucks

In the second example illustrated by Figure 6, yard truck YT1 transports containers 1 and 3, while yard truck YT2 takes care of container 2. The evaluation of the presented solution gives as result:

* Handling of container 1 :

- time $ht_1 = w_1 + \lambda_1 + r = 9$
- end time $cp_1 = s_1 + ht_1 = 0 + 9 = 9$

* Handling of container 2 :

- time $ht_2 = w_2 + \lambda_2 + r = 9$
- end time $cp_2 = s_2 + ht_2 = 6 + 9 = 15$

* Handling of container 3 :

- time $ht_3 = w_3 + \lambda_3 + r = 10$
- end time $cp_3 = s_3 + ht_3$, with $s_3 = s_2 + w_2 + t_{23} = 6 + 3 + 3 = 12$, so $cp_3 = 12 + 10 = 22$

* Total handling time (*makespan*) $C_{max} = Max(cp_1, cp_2, cp_3) = 22$

3.3. Genetic algorithm

The genetic algorithm is a heuristic method which was proposed in 1975 by Holland and developed in 1989 by Goldberg. This paradigm, combined with the terminology of genetics, allows us to exploit genetic algorithms: we find again the notions of population, individual, chromosome and gene. The population is the set of possible solutions, the individual represents a solution, the chromosome is a component of the solution and the gene is a characteristic. There are three evolution operators in genetic algorithm, to allow the evolution of a population:

- Selection: choice of the best adapted individuals.
- Crossing: mix the characteristics of solutions by the reproduction of the particularities of the selected individuals.
- Mutation: random change of individual characteristics.

For solving the scheduling problem and the operation problems in maritime port, some numerous researchers choose the genetic algorithm as the best meta-heuristic method.

The stage of creating a random population is the starting point of our algorithm, then the evaluation is the analysis of individuals to determine if a solution is available or not. After that we test if there is still another chromosome to consider: if no we obtain the solution, and if yes we apply the genetic algorithm operators selection, crossover and mutation (Algorithm 2).

3.3.1. Algorithm

Algorithm 2 Genetic algorithm

Population;

```
foreach chromosome do  
    Evaluation();  
    if last generation then  
        | Solution;  
    end  
    else  
        | selection();  
        | crossover();  
        | mutation();  
    end  
end
```

3.3.2. Representation of a chromosome

As shown in Table 2, a chromosome is represented by a range describing a quay crane unloading sequence. The chromosome is composed of multiple genes and each gene represents the index of a container. The chromosome is decoded by the yard truck transmission base which assigns transfer task to the first idle yard truck.

Table 2: Chromosome representation

5	1	4	2	3
---	---	---	---	---

3.3.3. Initialize the first generation

To form the first generation, some feasible solutions are randomly generated. Let S be the population size of the first generation. Therefore, S chromosomes are generated and each chromosome is a string to describe the quay crane unloading sequence.

3.3.4. Fitness evaluation

The fitness function is critical, so it is important to define it carefully to take into account all the parameters of the problem.

The fitness function is represented in equation 25, which is the inverse of the total completion time of all containers.

$$Fitness = 1/C_{max} \quad (25)$$

3.3.5. Roulette wheel selection

The circular wheel in roulette wheel selection is divided into many parts. We choose a start point and the wheel is rotated and the fixed point is set as the parent. We repeat the same process for the second parent. The steps of roulette wheel selection are as follows, and illustrated by Figure 7:

1. Calculate the sum of all fitnesses and call it S_1 .
2. Generate a number randomly between 0 and S_1 and call it R .
3. Add all fitnesses from the top of the population to a sum called S_2 , with a condition $S_2 < S_1$.
4. Choose the individual for which S_2 exceeds R .

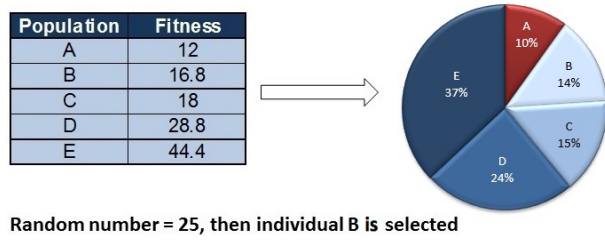


Figure 7: Roulette wheel selection

3.3.6. Order crossover

The steps of order crossover are as follows, and illustrated by Figure 8:

1. Select a random substring from one selected parent.
2. Create an offspring by copying the substring in their positions.
3. From the second parent, delete all existing genes in the substring and place from left to right all the genes in the empty positions of the offspring, taking in consideration the order of the sequence for the creation of the offspring.

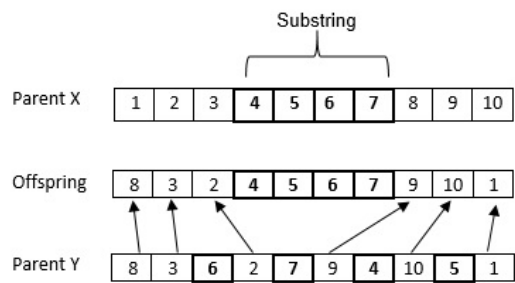


Figure 8: Crossover

3.3.7. Swap mutation

In mutation, we test all the individuals bit by bit (gene by gene), and we select randomly two positions from the chromosome and swap their values

(see Figure 9).

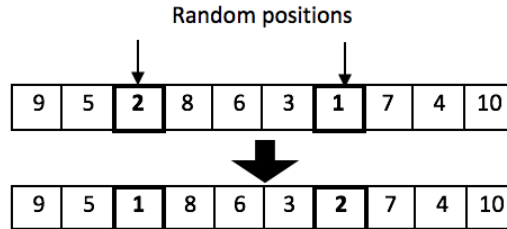


Figure 9: Mutation

4. Experimental results

Our mixed-integer linear program is solved by CPLEX 12.8 while the exact enumerative algorithm (EEA), lower bounds (LB), upper bounds (UB) and the genetic algorithm (GA) are all developed with java. All programs are executed on a MacBook professional with 8GB RAM and intel core i5. All needed parameters are randomly generated such as:

w_i is randomly generated between 250 and 300 time units (t.u), t_{ij} is randomly generated between 20 and 30 time units, λ_i is randomly generated between 150 and 200 time units and r is randomly generated between 30 and 50 time units.

At the end of this section, we present a comparison of the results obtained by our methods with the real results in the port of Tripoli-Lebanon.

4.1. MILP versus EEA

As shown in Table 3, the mixed-integer linear programming model solved with CPLEX and the exact enumerative algorithm, both give us the same completion time because they are exact methods. When we execute instance

16, for 10 containers and 3 yard trucks, CPLEX is unable to give results after 9 hours while EEA gave us results, so we can say that EEA is more efficient than CPLEX. But for instance 19, we execute EEA without any result after 5 hours. That's why we proposed a genetic algorithm in this paper to be compared with the EEA to obtain near-optimal solutions with an acceptable CPU time. Then we will adopt the EEA as the best exact method to be compared with the genetic algorithm.

$$\text{GAP}(\%) = ((\text{greater value} - \text{lower value}) / \text{greater value}) * 100.$$

Table 3: MILP vs EEA

Instance	C	T	Makespan (t.u)	CPU Time		
				CPLEX (s)	EEA (s)	GAP (%)
1	5	1	3069	< 1	< 1	-
2	5	2	2294	< 1	< 1	-
3	5	3	1815	< 1	< 1	-
4	6	1	3924	1.05	< 1	≈ 0
5	6	2	2902	1.32	< 1	≈ 0
6	6	3	1967	5.47	1.34	75.5
7	7	1	6992	6.44	1.53	76.24
8	7	2	3469	22.75	1.62	92.88
9	7	3	2432	39.18	4.41	88.74
10	8	1	8634	42.57	5.61	86.82
11	8	2	4305	242.63	6.57	97.29
12	8	3	3262	102.37	29.32	71.36
13	9	1	11199	2678.94	531.4	80.16
14	9	2	6512	3984.12	997.34	74.97
15	9	3	5060	5816.85	2013.81	65.38
16	10	3	9335	N.A	2581.95	-
17	10	4	9123	N.A	3797.16	-
18	10	5	6647	N.A	6914.45	-
19	15	3	N.A	N.A	N.A	-

4.2. Lower bound versus exact enumerative algorithm versus upper bound

As shown in Table 4, we compared EEA's results with the lower bounds and upper bounds results. We can check that the optimal completion time obtained by EEA is always between the lower and the upper bounds values. In all instances, the optimal solutions are higher than the lower bounds values for an average of 5.39% and lower than the upper bounds values for an average of 15%.

Table 4: LB vs EEA vs UB

Instance	C	T	Makespan			GAP	
			LB (t.u)	EEA (t.u)	UB (t.u)	(LB/EEA) (%)	(EEA/UB) (%)
1	5	1	2868	3069	3166	6.56	9.4
2	5	2	2122	2294	2398	7.49	7.35
3	5	3	1718	1815	2204	5.33	22.06
4	6	1	3634	3924	4248	7.39	14.46
5	6	2	2666	2902	3240	8.14	17.72
6	6	3	1809	1967	2424	8.04	25.37
7	7	1	6798	6992	8065	2.77	15.71
8	7	2	3257	3469	3573	6.11	8.84
9	7	3	2291	2432	2555	5.79	10.35
10	8	1	8365	8634	9022	3.11	7.28
11	8	2	4176	4305	4459	2.99	6.35
12	8	3	3121	3262	4192	4.32	25.54
13	9	1	10305	11199	13509	7.98	23.72
14	9	2	6158	6512	6647	5.44	7.35
15	9	3	4892	5060	6514	3.32	24.9
16	10	3	9082	9335	10141	2.71	10.44
17	10	4	8419	9123	9464	7.72	11.04
18	10	5	6212	6647	8296	6.54	25.12
19	15	3	12223	N.A	15157	-	-

4.3. Exact enumerative algorithm versus genetic algorithm

As shown in Table 5, from instance 19 for 15 containers and 3 or 4 yard trucks, EEA was unable to produce any solution after more than 5 hours, while GA produces a near optimal solution with a very acceptable CPU processing time. The GAP between the optimal solutions provided by EEA and the near optimal solutions provided by GA are all between 0% and 1.88%. Then we can consider the proposed genetic algorithm as a suitable algorithm to solve our problem.

Table 5: EEA vs GA

Instance	C	T	Makespan		CPU time	
			EEA (t.u)	GA (t.u)	EEA (s)	GA (s)
1	5	1	3069	3136	< 1	< 1
2	5	2	2294	2359	< 1	< 1
3	5	3	1815	1815	< 1	< 1
4	6	1	3924	4023	< 1	< 1
5	6	2	2902	2963	< 1	< 1
6	6	3	1967	2011	1.34	< 1
7	7	1	6992	7166	1.53	< 1
8	7	2	3469	3544	1.62	1.03
9	7	3	2432	2463	4.41	3.12
10	8	1	8634	8839	5.61	2.56
11	8	2	4305	4402	6.57	3.97
12	8	3	3262	3328	29.32	7.19
13	9	1	11199	11199	531.4	13.43
14	9	2	6512	6642	997.34	21.47
15	9	3	5060	5164	2013.81	27.38
16	10	3	9335	9602	2581.95	34.22
17	10	4	9123	9236	3797.16	43.69
18	10	5	6647	6779	6914.45	59.77
19	15	3	N.A	12926	N.A	102.35
20	15	4	N.A	11471	N.A	119.83

4.4. (EEA,GA) versus real experiments

In Table 6, we consider 6 instances (P_1 to P_6) corresponding to real situations in the port of Tripoli-Lebanon. As shown in Table 6, for example in the second instance P_2 for 6 containers and 2 yard trucks, the port's makespan is about 737 seconds while EEA's makespan is 574 seconds, so our model enables us 77.88% savings. For the same instance, also the GA's makespan is 613 and it is better than the port's makespan and it enables us 83.18% savings. We use the same parameters as those ones in the port to make a real results comparison.

Table 6: (EEA,GA) vs real experiments

Instance	C	T	Makespan			GAP	
			EEA (s)	GA (s)	Port (s)	(EEA/port) (%)	(GA/port) (%)
P_1	5	2	472	503	590	20	14.75
P_2	6	2	574	613	737	22.12	16.82
P_3	6	3	422	422	518	18.53	18.53
P_4	8	1	1051	1131	1360	22.72	16.84
P_5	8	2	913	989	1154	20.88	14.3
P_6	8	3	723	781	901	19.76	13.32

5. Results discussion

Our assumptions are consistent with the actual conditions that are present in the investigated industry and our model is actually able to reproduce the real conditions as it is shown with the tests conducted and the results obtained for the real case instances from the port of Tripoli-Lebanon.

In this study we compared our results with the port of Tripoli-Lebanon re-

sults, but in our future work we will achieve additional tests with more instances from the literature.

6. Conclusion

In our study, we proposed some methods to solve the studied problem, a mathematical formulation modeled with MILP, an exact enumerative algorithm (EEA) and a genetic algorithm (GA). We also proposed some lower and upper bounds of the searched makespan. We have shown that our methods are efficient and can be applied to improve port's results. In future work, we will extend this model to multiple quay cranes and multiple yard trucks scheduling problem in the mentioned port.

References

- [1] A. Skaf, S. Lamrous, Z. Hammoudan, 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).
- [2] A. Skaf, S. Lamrous, Z. Hammoudan, 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).
- [3] A. Skaf, S. Lamrous, Z. Hammoudan, M.-A. Manier, Single quay crane and multiple yard trucks scheduling problem with integration of reach-stacker cranes at port of tripoli-lebanon, International Conference on Systems, Man, and Cybernetics (SMC) (2019).

- [4] C. Daganzo, The quay crane scheduling problem, *Transportation research* 23 (1989) 159–175.
- [5] 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).
- [6] L. Chen, A. Langevin, Z. Lu, Integrated scheduling of crane handling and truck transportation in a maritime container terminal, *European Journal of Operational Research* 225 (2013) 142–152.
- [7] H. Dkhil, A. Yassine, H. Chabchoub, Optimization of container handling systems in automated maritime terminal, Springer-Verlag Berlin Heidelberg (2013).
- [8] Z. Xue, C. Zhang, L. Miao, W.-H. Lin, An ant colony algorithm for yard truck scheduling and yard location assignment problems with precedence constraints, *Journal of Systems Science and Systems Engineering* (2013) 021–037.
- [9] A. Diabat, E. Theodorou, An integrated quay crane assignment and scheduling problem, *Computers and industrial engineering* 73 (2014) 115–123.
- [10] S. M. Homayouni, S. H. Tang, 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) 545–556.

- [11] B. Niu, T. Xie, Q. Duan, L. Tan, Particle swarm optimization for integrated yard truck scheduling and storage allocation problem, *IEEE Congress on Evolutionary Computation (CEC)* (2014).
- [12] L. Tang, J. Zhao, J. Liu, Modeling and solution of the joint quay crane and truck scheduling problem, *European Journal of Operational Research* 236 (2014) 978–990.
- [13] N. Al-Dhaheri, A. Diabat, The quay crane scheduling problem, *Journal of Manufacturing Systems* 36 (2015) 87–94.
- [14] N. Kaveshgar, N. Huynh, Integrated quay crane and yard truck scheduling for unloading inbound containers, *Int. Production Economics* 159 (2015) 168–177.
- [15] A. N. Boysen, B. D. Briskorn, C. F. Meisel, A generalized classification scheme for crane scheduling with interference, *European Journal of Operational Research* 000 (2016) 1–15.
- [16] M. Liu, F. Zheng, Y. Xu, C. Chu, Approximation algorithm for uniform quay crane scheduling at container ports, *Discrete Mathematics, Algorithms and Applications* (2016).
- [17] Y. Liu, T. Liu, The hybrid intelligence swarm algorithm for berth-quay cranes and trucks scheduling optimization problem, *15th Int’l Conf. on Cognitive and Cognitive Computing* (2016).
- [18] X. Zhang, Q. Zeng, Z. Yang, Modeling the mixed storage strategy for quay crane double cycling in container terminals, *Transportation Research Part E* 94 (2016) 171–187.

- [19] L. Zhen, S. Yu, ShuaianWang, Z. Sun, Scheduling quay cranes and yard trucks for unloading operations in container ports, Springer Science+Business Media New York (2016).
- [20] L. Haoyuan, S. Qi, Simulation-based optimization on quay crane scheduling of container terminals, 29th Chinese Control and Decision Conference (CCDC) (2017).
- [21] C. Xiazhong, Z. Ye, H. Hongtao, Optimization research of joint quay crane scheduling and block selection in container terminals, International Conference on Service Systems and Service Management (2017).
- [22] C. Liang, L. Fan, D. Xu, Y. Ding, M. Gen, Research on coupling scheduling of quay crane dispatch and configuration in the container terminal, Computers and Industrial Engineering 125 (2018) 649–657.
- [23] B. Vahdani, F. Mansour, M. Soltani, D. Veysmoradi, Bi-objective optimization for integrating quay crane and internal truck assignment with challenges of trucks sharing, Knowledge-Based Systems 163 (2019) 675–692.