# Genetic Algorithm to Optimize Unloading of Large Containers Vessel in the Port

**Sanaa Aidi**
Student researcher, Mechanical department
School of Electricity and mechanics ENSEM /Hassan II University
Casablanca, MOROCCO
sanaa.aidi@gmail.com

**Imane Torbi**
Student researcher, Mechanical department
School of Electricity and mechanics ENSEM /Hassan II University
Casablanca, MOROCCO
torbi.imane@gmail.com

**Mazouzi mohamed**
Professor, Mechanical department
School of Electricity and mechanics ENSEM /Hassan II University
Casablanca, MOROCCO
m.mazouzi@ensem.ac.ma

## Abstract

The scheduling of quay cranes is a global problem that all the ports of the world seek to solve, with the main objective of minimizing the loading and clearing time of container ships and thus reducing the berthing time in the marine terminals. It consists of assigning each gantry crane to a set of bays for a given ship while sequencing the unloading of these bays. In a previous study, they discussed different methods whose main drawback is the difficulty of obtaining results for large instances. Therefore, in this article, we propose a genetic algorithm that allows us to overcome this and quickly obtain near-optimal solutions. We have tested and validated our method on real instances from the port.

## Keywords

Optimization, quay crane, scheduling problem, genetic algorithm, container.

## 1. Introduction

The Port of Casablanca is the second port in Morocco, it receives about 1000 ships every year, and the port receives container vessels loaded with more than 70000 containers. Quay cranes serve to unload or load containers from the vessel to the store or the trucks. Like the other ports, one of the challenges in the port is to address the Quay Crane Scheduling Problem (QCSP), subject to non-interference constraints for the quay cranes, assignments conditions with bays ensure thatch bay can be handled by at least one quay crane. Fig.1 is an illustration of the quay cranes and the containers while Fig.2 is an illustration of the containers vessels and quay cranes assignment

In a previous study, we proposed exact methods to solve the QCSP with non-interference constraints in the port. But solving large instances spends long execution times and reaches the out-of-memory exception. So the objective of this paper is to reduce the execution time by using a metaheuristic approach.

Fig. 1. Quay cranes and containers

Fig. 2. Containers vessel

## 2. Literature Review

Several researchers addressed the QCSP in the literature [1] and resented a novel, mixed integer programming (MIP) model for the quay crane scheduling and assignment problem, after that[2] determined the sequence of unloading operationalization, to minimize the weighted operation time of jobs and travel time.

[3] proposed a modified genetic algorithm to deal with the problem and to test the optimization reliability of the proposed algorithm. A set of well-known benchmarking problems was solved, meanwhile [4] aimed to minimize the total handle time of all tasks, using a polynomial-time algorithm to solve the problem.[5]founded the handling sequence of tasks at the ship; their objective is to minimize the time spent by the vessel at berth and minimize the total cost. Mixed-integer linear programming and ant colony algorithm was used in the previous papers to solve the problem. [6] proposed models and algorithms for the general double-cycling problem with internal reshuffles, where reshuffle containers are allowed to move directly from one stack to another. A branch-and-price framework is used to solve the problem.[7]studied a scheduling problem with two uniform quay cranes; their objective was to minimize the turn-around time of a vessel, meanwhile [2] Awar et al. (2016) proposed a solution to minimize the entire processing time for all vessels, the mixed-integer linear algorithm is used to solve the problem.[8]proposed a simulation model of the container terminal, to solve its quay crane scheduling problem, to optimize the shortest total delay time for all vessels.[9] elaborated a model that combines three distinct problems, namely berth allocation, quay crane assignment, and quay crane scheduling that arises in container ports, a genetic algorithm was developed to solve the problem, and finally. [10] aimed to find the optimal storage location of container groups while minimizing both the maximum completion time and the traveling time of trucks. They used mixed-integer linear programming and tabu search to solve the problem. This brief review suggests that methods such as genetic algorithms are adapted to solve the Quay Crane Scheduling Problem (QCSP).[11]we presented the different stages of the container unloading process at the port of Casablanca as well as the technical data for the resolution via the genetic algorithm.

## 3. Methods

Due to the difficulty to solve large instances, we propose in this work a genetic algorithm (GA). A genetic algorithm (GA) is a heuristic search algorithm that was first proposed by [Holland, 1975] and developed and applied by [Goldberg, 1989]. It can be easily coded and often gives good solutions. Roulette wheel selection, order crossover, and swap mutation are used due to their results efficacy.

The steps of the genetic algorithm are provided as follows: first, we generate an initial population of solutions, then for all generations, we should make a test about the non-interference constraints of quay cranes in other words we want to know if quay cranes work without crossing. If yes, we calculate the fitness value, otherwise, we set the fitness to value zero. After that, if the current generation is the last one, then the program is ended, otherwise, we should execute the steps of the genetic algorithm such as roulette wheel selection, order crossover, and swap mutation. We have chosen the genetic algorithm because of its simplicity, easy to be implementation n and logical reason to use its operators like selection, crossover, and mutation.

## 4. Data Collection

### Fitness:

The previous procedure prevents interference between quay cranes, but all quay cranes must be checked if they satisfy the constraints (4) and (5) can be used, and then the quay crane scheduling can be checked if it satisfies (6), if yes then the fitness is as shown in Eq.(10) else it will be set to zero.

$$\text{Fitness} = 1/C\ \text{max} \qquad (10)$$

### Selection:

Selection is a process in which individuals from a population are chosen according to the values of their cost or" fitness" function to form a new population.

Individuals evolve through successive iterations of selection, called generations. Each individual is selected proportionally to their «fitness" function, so an individual with a higher fitness function will be more likely to be selected than another with a lower fitness value. This function can be considered as a measure of profit or quality that one wishes to maximize. A simple operator of selection is the technique of the weighted roulette where each individual of a population occupies a surface of the roulette proportional to the value of its function «fitness". For reproduction, candidates are selected with a probability proportional to their fitness. For each selection of an individual, a simple rotation of the wheel gives the selected candidate. Roulette Wheel Selection works as follows: Calculate the sum of all fitness, then generate a number between 0 and the sum, after that add the fitness values to a partial sum X starting from the top of the population, and finally, the chosen chromosome is the first chromosome for which X overpass the random number

### Crossover:

Crossover works as follows: Select a random substring from one parent randomly, then create an offspring by copying the selected substring in their corresponding positions. After that, delete from the second parent all existing in the substring and place the genes in the empty positions of the offspring from left to right. Finally, we should take into consideration the order of the sequence of the offspring's creation. The procedure is illustrated in Fig.3

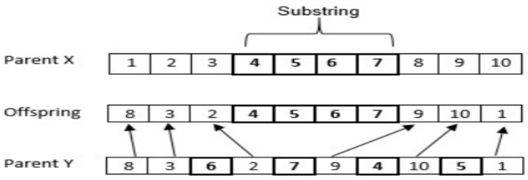

Figure 3: Selection

In the mutation, all individuals are tested bit by bit. It selects two positions on the chromosome randomly, then swaps the values on these positions. The procedure is illustrated in Fig.4

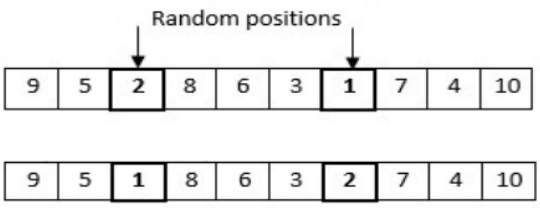

Figure 4: Mutation

## 5. Results and Discussion

The quay crane problem we have studied has the following characteristics containers unloading is in a single vessel, all quay cranes move on the same track and need to work without any interference between them. A crane is assigned to a bay and can work in another bay before it has finished unloading containers in the current bay. Finally, each bay is handled by at most one quay crane at a time. In this study, the quay cranes' travel time between bays is ignored as it is very short compared to the unloading times

**Notations:**
j: index for bay (j=1 to B)
Cj: number of containers in bay j
Q: number of quay cranes
i: index for quay crane (i=1 to Q)
B: number of bays

Tc: time to unload a container by a quay crane and put it in the storage. It is supposed to be the same for all containers in all bays.

M : big integer

**Decision variables**

$X_{(j,\, i)} = 1$ if bay j is handled by quay crane i

$= 0$ otherwise

$Z_{(j,\, j)} = 1$ if the unloading of bay j finishes before starting to unload bay j

$= 0$ otherwise

$T_j$ completion time of bay j

$C_{max}$ makespan

**Modeling**

The following is the mixed-integer linear formulation we have presented in previous work[13] Here we use some of those equations in the fitness calculation of the genetic algorithm which we will talk about later

Objective

$$\text{Minimize } C\ max \qquad (1)$$

Subject to

$$\sum x_{(j,\, i)} \geq 1\ \forall\, j \in B \qquad (2)$$

$$t_j \geq (T\ c *C_j)\ \forall\, j \in B \qquad (3)$$

$$t_j - t_j + (T_c *C_j) + z_{(j,\, j)} * M > 0\ \forall\, j, j\ 0 \in B \qquad (4)$$

$$t_j - t_j + (T\ c *C_j) - (1 - z_{(j,\, j)}) * M \leq 0\ \forall\, j, j\ 0 \in B \qquad (5)$$

$$\sum\ i * x_{(j,i)} + 1 \leq \sum\ i' * x_{(j,i)} + (z_{(j,\, j)} + z_{(j\, ,\, j)}) * M$$
$$\forall\, j, j\ 0 \in B, j < j\ 0 \qquad (6)$$

$$C\ max = max\ \ t_j \qquad (7)$$

$$x_{(j,i)} = [0,1]\ \forall\, j \in B, \forall i \in Q \qquad (8)$$

$$z_{(j,\, j)} = [0,1]\ \forall\, j, j\ 0 \in B, j < j\ 0 \qquad (9)$$

Constraint (1) is the objective function that serves to minimize the latest completion time among all bays.
Constraint (2) ensures that each bay can be handled by at least one quay crane constraint (3) ensures that the completion time is bigger than the working time in each bay (Working time = number of containers * time needed to unload a container and store it).
Constraint (4) shows when j finishes before bay j bay' starts and constraint (5) shows the opposite.

Constraint (6) avoids interference between the quay cranes.
Constraint (7) defines the Cmax value
In a genetic algorithm, a solution is called a chromosome. In our case, it is composed of a series of bays

Table 1. Chromosome representation

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| **5** | 9 | 1 | 2 | 6 | 10 | 2 | 4 | 3 | 1 |

For example, the second element (second gene) in the chromosome represents bay number 9.

## Initial positions for quay cranes:

We propose to fix the initial position of each quay crane i to the position of bay 1+ (i-1)*D, (i ∈ Q) with D should be a random number integer between 1 and B/Q (if B/Q is a rational number, we take its natural part number). It allows a homogeneous initial distribution of the cranes along the vessel.

Let us note that with this formula, the first quay crane position should always be at the first bay

Illustration: Let us suppose that we have 10 bays and 3 quay cranes, then D should be a random number between 1 and 10/3 (hence between 1 and 3). If D=3, then the main position of quay crane 1 is on bay 1, quay crane 2 is on bay 4 and quay crane 3 is on bay 7.

The main completion time of the three quay cranes is initialized to 0 because no unloading operation has been performed until now

### Verification of unloading constraints:

Table2. verification of constraints

| Constraint | Verification |
|---|---|
| Non-Crossing | The first step is a test of interference which identifies the two cranes which are candidates to unload the bay in the current gene, i.e. we keep the cranes whose last position is immediately at the left or the right of this current bay. For our example, quay crane 2 and quay crane 3 can handle bay 6 without interference with quay crane 1 |
| Makespan | Among these two cranes, we choose the one which is available sooner. In the case of equality, that means if the two cranes have the same completion time, then we go to step 3 to discriminate them according to a distance criterion. In our example, the completion times of |

| | |
|---|---|
| | quay crane 2 and quay crane 3 are equal. |
| Distance | If no crane has been chosen in step 2 with the time criterion, we assign the current bay to the nearest crane. In the case of equality (if the two cranes are at an equal distance from the bay), we choose the crane at the left (with the smallest index).<br>The distance between bay 6 and bay 4 (which quay crane 2 works inside) is 1 bay and the distance between bay 6 and bay 7 (which quay crane 3 works inside), is 0 bay. Then crane 3 is to from bay 6 than crane 2. |
| Assignment | Assign bay 6 to quay crane 3. Update the position of quay crane 3 and the associated completion time. Then update the current gene (following one) in the chromosome (gene 2 then bay 9 in our example). Repeat the four steps until all bays are assigned to one quay crane |

## 5.1 Numerical Results

Comparison between dynamic programming and genetic algorithm
In the literature, we solved some instances using CPLEX and Dynamic programming, small and large sizes instances were generated randomly, and container numbers are also generated in a random way between 10 and 50 containers. In this article, we add the genetic algorithm results time, as shown in Table 3,
According to the primary tests, the population size, the probability of mutation, the probability of crossover, and the limit of generations are 300, 0.2, 0.25, and 1000 respectively, in these computational experiments.

Table 3: Results

| No | (B*C) | DP(mins) | Makespan GA(mins) | GAP(%) |
|---|---|---|---|---|
| 1 | 12*2 | 171.99 | 173.16 | 0.68 |
| 2 | 12*3 | 115.83 | 117.12 | 1.1 |
| 3 | 13*2 | 184.86 | 184.86 | 0 |
| 4 | 13*3 | 124.02 | 126.36 | 1.85 |
| 5 | 14*2 | 201.79 | 204.75 | 1.45 |
| 6 | 14*3 | 128.7 | 129.92 | 0.95 |
| 7 | 15*2 | 226.13 | 231.66 | 2.39 |

B x C: bays number x cranes number
NA: Interrupt execution time after 5 hours. (No results)
GAP = ((DP makespan - GA makespan)/DP makespan) *100
The GAP between the optimal solutions and the near-optimal solution is between 0% and 2.8%, so we can say that the proposed genetic algorithm is successful and practical for the mentioned problem of the port.

## 6. Conclusion

On large instances, this paper shows the premises of the QCSP solving by metaheuristics. The results show that the proposal is promising to have quickly good solutions. Here, the first attempt explores a search space with a genetic algorithm. To further improve the quality of solutions in terms of execution time and convergence towards the optimal solution, other types of metaheuristics deserve to be tested. A large-scale comparison of benchmarks in the literature is obvious to the following of this work.

## References

[1] R. Tavakkoli-Moghaddam, A. Makui, S. Salahi, M. Bazzazi, and F. Taheri, "An efficient algorithm for solving a new mathematical model for a quay crane scheduling problem in container ports," *Comput. Ind. Eng.*, vol. 56, no. 1, pp. 241–248, 2009, DOI: 10.1016/j.cie.2008.0DOI11.

[2] W. Su and H. Wenbin, "Multi quay crane scheduling problem based on ACO in container terminals," *Proc. - Int. Conf. Manag. Serv. Sci. MASS 2009*, pp. 2–5, 2009, DOI: 10.1109/ICMSS.2009.5301977.

[3] S. H. Chung and K. L. Choy, "A modified genetic algorithm for quay crane scheduling operations," *Expert Syst. Appl.*, vol. 39, no. 4, pp. 4213–4221, 2012, DOI: 10.1016/j.eswa.2011.09.113.

[4] Y. Ding, L. Guolong, and L. Chengji, "Model and heuristic algorithm for Quay Crane Scheduling at the container terminal," *Proc. - 2012 9th Int. Conf. Fuzzy Syst. Knowl. Discov. FSKD 2012*, no. Find, pp. 1054–1061, 2012, DOI: 10.1109/FSKD.2012.6233825.

[5] L. Azza, M. El Merouani, and A. Medouri, "Ant colony system for solving Quay Crane Scheduling Problema in a container terminal," *Proc. 2nd IEEE Int. Conf. Logistics. Oper. Manag. GOL 2014*, pp. 176–180, 2014, DOI: 10.1109/GOL.2014.6887437.

[6] F. Zheng, Y. Pang, M. Liu, and Y. Xu, "Dynamic programming algorithms for the general quay crane double-cycling problem with internal-reshuffles," *J. Comb. Optim.*, vol. 39, no. 3, pp. 708–724, 202DOIdoi: 10.1007/s10878-019-00508-9.

[7] M. Liu, F. Zheng, Y. Xu, and C. Chu, "Approximation algorithm for uniform quay crane scheduling at

container ports," *Discret. Math. Algorithms Appl.*, vol. 8, no. 2, pp. 1–9, 201DOIdoi: 10.1142/S179383091650018X.

[8] L. Haoyuan and S. Qi, "Simulation-based optimization on quay crane scheduling of container terminals," *Proc. 29th Chinese Control Decis. Conf. CCDC 2017*, pp. 1135–1139, 201DOIdoi: 10.1109/CCDC.2017.7978689.

[9] A. Salhi, G. Alsoufi, and X. Yang, "An evolutionary approach to a combined mixed integer programming model of seaside operations as arise in container ports," *Ann. Oper. Res.*, vol. 272, no. 1–2, pp. 69–98, 201DOIdoi: 10.1007/s10479-017-2539-7.

[10] X. Chen, Y. Zhang, and H. Hu, "Optimization research of joint quay crane scheduling and block selection in container terminals," *14th Int. Conf. Serv. Syst. Serv. Manag. ICSSSM 2017 - Proc.*, 201DOIdoi: 10.1109/ICSSSM.2017.7996175.

[11] S. Aidi, Torbierbi, M. Mazouzi, and I. Belassiria, "Planning of one-way quay cranes using the genetic algorithm," *SHS Web Conf.*, vol. 119, p. 07005, 2021, DOI: 10.1051/sh conf/202111907005.

[12] A. Skaf, S. Lamrous, Z. Hammoudan, and M. A. Manier, "Genetic algorithm to optimize unloading of large containers vessel in the port of Tripoli-Lebanon," *2019 6th Int. Conf. Control. Decis. Inf. Technol. CoDIT 2019*, no. 978, pp. 569–574, 2019, DOI: 10.1109/CoDIT.2019.8820367.

[13] A. Skaf, S. Lamrous, Z. Hammoudan, and M. A. Manier, "Exact method for single vessel and multiple quay cranes to solve scheduling problem at the port of Tripoli-Lebanon," *IEEE Int. Conf. Ind. Eng. Eng. Manag.*, vol. 2019-Decem, pp. 457–461, 2019, DOI: 10.1109/IEEM.2018.8607546.