Exact method for single vessel and multiple quay cranes to solve scheduling problem at port of Tripoli-Lebanon

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

▶ To cite this version:

HAL Id: hal-02991560

https://hal.science/hal-02991560

Submitted on 6 Nov 2020

Exact method for single vessel and multiple quay cranes to solve scheduling problem at port of Tripoli-Lebanon

A. SKAF[1,2], S. LAMROUS[1], Z. HAMMOUDAN[2], M. A. MANIER[1]

[1] Univ. Bourgogne Franche-Comté, FEMTO-ST Institute/CNRS, (UTBM), Belfort, 90010, France
[2] 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 discusses the Quay Crane Scheduling Problem (QCSP) at port of Tripoli - Lebanon, determines the unloading/loading sequences of bays for quay cranes assigned to a single container vessel, provides a mixed integer programming model for the quay crane scheduling problem and proposes a dynamic programming algorithm to solve the QCSP. The objective of this paper is to minimize the completion time of unloading/loading containers and therefore to reduce the docking time of the vessel in the terminal. Finally the results of this paper are compared to the port results.

Keywords – Optimization, quay crane, scheduling problem, mixed integer linear programming, case study

I. INTRODUCTION

Port of Tripoli is in the north of Lebanon, it has an area of 3.000.000 m^2 divided to 2.200.000 m^2 of water zone, 320.000 m^2 for land zone and the rest for the dump zone. The dump zone in the port is for the container terminal and the free market. About 400 ships arrives to the port every year with an average of 32 ships per month.

Quay crane is a type of crane in the port for the loading and unloading of intermodal containers from vessels. A container is any receptacle or enclosure for holding a product used in storage, packaging, and shipping. As illustrated in Fig.1, every vessel is divided to many bays and each bay contains a number of containers and Fig.2 shows the quay crane assignment to container vessel.

The port has only two quay cranes and is waiting for two other ones to arrive. The largest vessel that reached the port has a maximum number of 12 or 13 bays. At this moment, the largest amount of containers on a vessel in the port of Tripoli is about 550 containers.

This study aims to obtain the minimal time of unloading/loading of all containers. It is a generalization of the standard QCSP where it is compulsory to unload/load the specified containers from the vessel.

In our problem, the number of bays has to be higher than the number of used cranes, otherwise we cannot have an optimization because we will have an idle crane(s).

Usually in the port of Tripoli, one quay crane can unload containers from a bay at a time and each quay crane cannot move to another bay before it completes the current bay; all quay cranes are on the same track, cannot cross over each other (non-interference constraints) and are assigned to a single vessel.

The working time in a bay is equal to the product of the containers number by the time needed for the quay crane to unload one container and put it in the storage location.

The unloading process is considered to be identical to the loading process because these two processes are symmetric. The rest of this paper is organized as follows: section II provides a brief for related work. Section III formally defines the mathematical formulation and introduces the constraints added to the initial linear program proposed, to improve its solving. Section IV provides the dynamic programming algorithm we applied. Section V is dedicated to experimental results and comparison with the port of Tripoli-Lebanon results. Section VI gives a conclusion and directions for future research.

II. RELATED WORK

A lot of papers have talked about this subject, but in this paper there are specific terms for the port of Tripoli-LEBANON such as quay cranes non-crossing, conditions for the assignments with bays and positionning.

[1] Daganzo (1989) is one of the first who studied the QCSP for multiple container vessels. He proposed that container vessels were to be divided into bays, and only one quay crane could work on a bay at a time. Quay cranes could be moved freely and quickly from one bay to another, and container vessels cannot depart until all their bays have to be unloaded. The objective is to minimize the cost of delay with an exact and approximate solution method. Furthermore, [2] Peterkofsky and Daganzo (1990) developed a branch and bound solution method for the static QCSP only if the quay cranes could cross over each other. Otherwise, [3] Kim and Park (2004) discussed the QCSP with non-interference constraints in which only a single container vessel was considered. The objective was to minimize the total completion time of all quay cranes.

[4] Lim et al. (2004) assumed that container vessel is a job and when this job is assigned to quay crane there is a profit value. They found an assignment matching which maximized the total profit. Dynamic programming algorithm and taboo search are used to provide a solution to the problem.

[5] Steeken and Stahlbock (2004) aimed to study the QCSP to enhance the competitiveness of port container terminals.

[6] Liang et al. (2008) addressed the berthing position problem and determining the number of quay cranes assigned to each ship; the objective was to minimize the handling time and the waiting time for every ship; a genetic algorithm was developed to find the best solution.

[7] Wang and Hu (2010) made a study to perform knowledge for quay crane and berth scheduling.

[8] Yi et al. (2012) gave a general model for the quay crane scheduling problem and introduced a heuristic model to solve it. Their objective is minimizing the total handling time of all tasks. Furthermore, [9] Diabat and Theodorou (2014) developed a formulation for the quay crane scheduling problem and quay crane assignment which accounts for crane position, and proposed a genetic algorithm to solve it. MoreOver, [10] Al-Dhaheri and Diabat (2015) determined the sequence of unloading operations of a vessel that a set number of quay cranes will perform so that the completion time of the operations is minimized; a mixed-integer programming (MIP) formulation was presented for the problem.

[11] Boysen et al. (2016) provided a classification scheme for crane scheduling problems without considering the non-interference constraints between the quay cranes. A scheme was applied to classify the existing literature, to determine the status-quo of complexity results, and to identify future research, while [12] Msakni et al. (2016) aimed to study the problem of determining the assignment of quay cranes to containers vessels and the scheduling operation by each quay crane and developed an exact method based on a branch-and-price algorithm.

[13] Oliveira et al. (2016) focused on the integration of quay side operations and the QCSP taking in consideration minimizing the time spent and the fuel consumption.

This study proposes two methods for solving the Quay Crane Scheduling Problem (QCSP). The first one is a formulation of a mixed-integer programming model, which can then be solved by different solvers such as CPLEX to find the optimal value. The second one is based on a Dynamic Programming Algorithm.

In order to evaluate the two methods performance in the solving of the mentioned QCSP, they were tested using a small and large numbers of instances. At the end of this study, the Dynamic Programming Algorithm was found to be better in performance compared to the MIP; It was much faster and could be used to solve small and large instances, while the MIP was limited to the small instances.

The contribution revealed to presenting the Quay Cranes Work Order in the bays without crossing, taking in consideration that the Quay Cranes should finish its job in a bay before moving to another one and the way how its work.
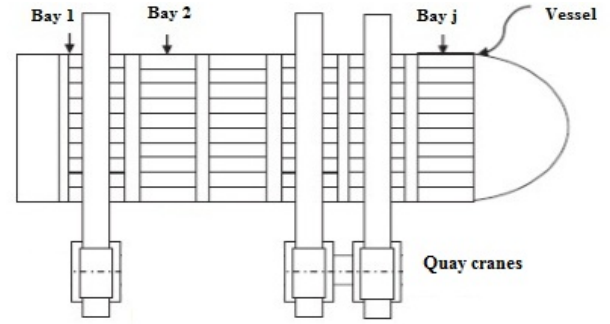

Fig. 1. Containers vessel

Fig. 2. Explanation of quay cranes scheduling

III. MATHEMATICAL FORMULATION

A. Assumptions

- Quay cranes unload containers from a single vessel.
- Quay cranes are on the same track and can not be crossed.
- Each quay crane can be assigned to at most one bay at a time.
- Each bay is handled by at most one quay crane at a time.
- A quay crane can not work on another bay before it completes its work in the current bay.
- The travel time of a quay crane between two bays is ignored because it is small (average 10 seconds).

B. Notations

N	Number of Quay Cranes, indexed i, $1 \leq i \leq N$
B	Number of Bays, indexed j, $1 \leq j \leq B$
C_j	Number of Containers in bay j
Tc	Time needed for the QC to unload a container and put it in the storage location
M	Big integer

C. 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

D. Formulation

The mixed integer linear formulation :

Objective

Minimize C_{max} (1)

Subject to

$$t_j \geq (Tc * C_j) \quad \forall j \in B \tag{2}$$

$$\sum_{i=1}^{N} x_{(j,i)} = 1 \quad \forall j \in B \tag{3}$$

$$t_j - t_{j'} + (Tc * C_{j'}) + z_{(j,j')} * M > 0 \quad \forall j, j' \in B \tag{4}$$

$$t_j - t_{j'} + (Tc * C_{j'}) - (1 - z_{(j,j')}) * M \leq 0 \quad \forall j, j' \in B \tag{5}$$

$$\sum_{i=1}^{N} i * x_{(j,i)} - \sum_{i'=1}^{N} i' * x_{(j',i')} + 1 \leq (z_{(j,j')} + z_{(j',j)}) * M$$
$$\forall j, j' \in B, j < j' \tag{6}$$

$$C_{max} = \max_j t_j \tag{7}$$

$$x_{(j,i)} = [0,1] \quad \forall j \in B, \forall i \in N \tag{8}$$

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

Equation (1) is the objective function which aims to minimize the makespan (latest completion time among all bays). Constraint (2) ensures that the completion time should be upper or equal to the working time in each bay (Working time = number of containers * time needed to unload a container and store it). Constraint (3) ensures that every bay must be handled only by one quay crane. Constraint (4) indicates when bay j finishes before bay j' starts ($z_{(j,j')} = 1$ when $t_j - t_{j'} + (Tc * C_{j'}) \leq 0$).
Constraint (5) indicates when bay j finishes after bay j' starts ($z_{(j,j')} = 0$ when $t_j - t_{j'} + (Tc * C_{j'}) > 0$).
Constraint (6) avoid the interference between the quay cranes. If bays j and j' are performed simultaneously, this means that $(z_{(j,j')} + z_{(j',j)}) = 0$.
If quay crane i and i' work in bay j and j' respectively, then $i+1 \leq i'$. Constraint (7) defines the C_{max} value.
The total containers number is $|C| = \sum_{j=1}^{B} C_j$.

IV. DYNAMIC PROGRAMMING ALGORITHM

A flowchart is a type of diagram that represents an algorithm, work flow or process. In this section, a generalized dynamic programming scheme is proposed, named Quay Crane Scheduling Problem Dynamic Programming Algorithm (QCSPDPA) , to build the optimal solutions starting from a single unloaded/loaded container and ending with all the unloaded/loaded containers (Fig.3).
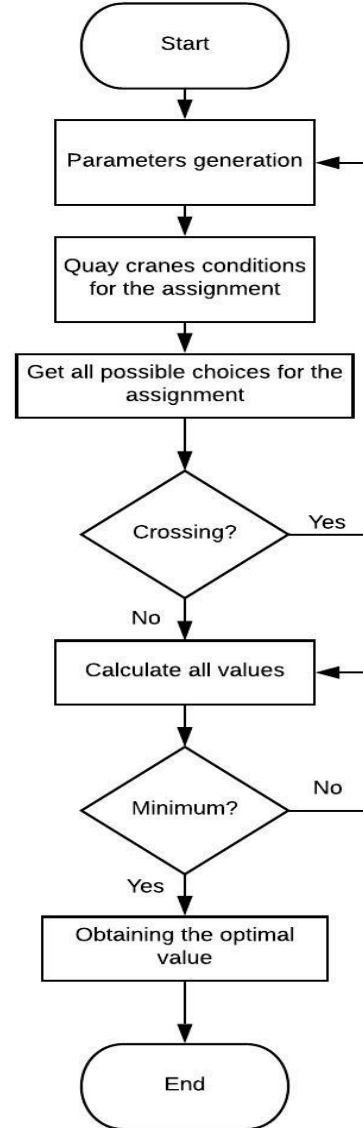

Fig. 3. The flowchart of the dynamic programming algorithm

Dynamic Programming Algorithm Description

The QCSPDPA is developed with JAVA J2EE.
First step of the algorithm is defining all parameters (number of cranes, number of bays....), then creating a function for the quay cranes work order conditions for the non-crossing (assignments) to get all possible choices

for all the assignments. Second step is making tests if each assignment satisfies the crossing conditions, if no we test another assignment, if yes we continue to the final step. Finally the optimal solution is the minimum of all assignments.

This procedure allows to find all possible solutions for the completion time which grows very quickly when the cranes number or the bays number increases.

The basic and important thing of this algorithm is to find all possibilities of quay cranes work order without any crossing between each other.

The QCSPDPA is based on finding the optimal solution for the cranes-bays assignment with non-crossing constraints and for the conditions of bays work, which is the case of the port of Tripoli-Lebanon.

V. EXPERIMENTAL RESULTS

Some experimental instances were solved using CPLEX 12.7.1 (commercial software for exactly solving integer programming) and Dynamic Programming (DP) (It refers to simplifying a complex problem by dividing it into simpler sub-problems recursively).

Programs are executed in a HP laptop core i5 CPU 2.50 GHZ with 12GB RAM.

A. Instances with small and large sizes and comparison between CPLEX and Dynamic Programming

As shown in Table I, small and large sizes instances were created randomly, and the number of containers in each bay is randomly generated between 10 and 50 containers; the time needed to unload a container and put it in the store location is about 1.17 minutes.

TABLE I
EXPERIMENTAL RESULTS

No.	(B x C)	Makespan (mins)	CPU Time CPLEX CPU (s)	DP CPU (s)	GAP (%)
1	4 x 2	43.29	0.15	0.015	90
2	4 x 3	29.25	0.14	0.013	90.71
3	6 x 2	95.94	0.15	0.031	79.33
4	6 x 3	69.03	0.17	0.029	82.94
5	8 x 2	111.15	0.21	0.051	75.71
6	8 x 3	83.07	0.12	0.048	60
7	10 x 2	134.55	02.01	0.075	96.26
8	10 x 3	92.43	0.28	0.069	75.36
9	12 x 2	171.99	221.52	0.59	99.73
10	12 x 3	115.83	5.24	0.36	93.12
11	13 x 2	184.86	2079.29	1.03	99.95
12	13 x 3	124.02	32.09	0.97	96.97
13	14 x 2	201.79	91288.63	2.36	99.99
14	14 x 3	128.7	75.87	2.45	96.77
15	15 x 2	226.13	144578.36	2.78	99.99
16	15 x 3	139.89	510.19	3.16	99.38
17	16 x 2	215.28	> 144578.36	3.57	-
18	16 x 3	152.1	> 144578.36	4.83	-
19	17 x 2	221.13	> 144578.36	5.41	-
20	17 x 3	156.78	> 144578.36	10.95	-
21	18 x 2	249.21	> 144578.36	12.97	-
22	18 x 3	168.487	> 144578.36	36.57	-

- (B x C) means (Bays x Cranes).
- GAP = ((CPLEX CPU - DP CPU)/CPLEX CPU)*100.
- DP CPU time < CPLEX CPU time.
- CPU time increases as the instance increases.

The makespan value is the same for CPLEX and Dynamic programming algorithm, but from instances 17 to 22, CPLEX took a long time of execution without any result while the DP gave results. DP compared to the CPLEX is much faster and could be used to solve instances quickly, while CPLEX is limited to the small instances.

B. Results compared to port of TRIPOLI

As shown in Table II, a simple comparison is made between this paper results and the results collected from the port of Tripoli. In first operation, an example for two quay cranes and six bays containing 25 containers, the time needed to unload a container and store it is about 1.17 minutes. The optimal result is 19.5 minutes and cranes cannot cross over each other.

In the port, the unloading time of 25 containers from the vessel is about 40 minutes, then our algorithm enables us to save time by 51.25%.

The Gap(%) is high due to the waiting time in the port, or to a technical problem, or to the cranes-bays assignments.

The model accommodates the true condition of the Port, in another meaning, the paper result and the port result are given in the same conditions (Cranes number, bays number and containers number in each bay).

GAP(%) = ((Port result - case study result)/Port result)*100.

TABLE II
QUAY CRANES SCHEDULING PROBLEM BENCHMARKS

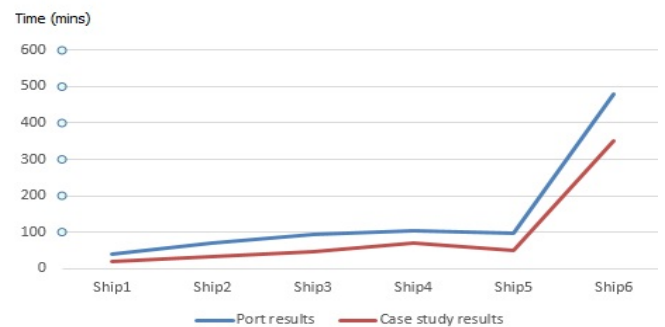| N.	Cranes number	Bays number	Containers number	Port results (mins)	Case study results (mins)	GAP (%)
1	2	6	25	40	19.5	51.25
2	2	3	40	70	32.76	53.2
3	2	4	62	95	44.78	52.86
4	2	3	90	103	70.2	31.84
5	2	4	84	96	49.14	48.81
6	2	5	600	480	351	26.88

Fig. 4.   Line chart for TableII

As shown in Fig.4, a line chart is made to explain the results presented in Table II.

The blue curve is for the port of Tripoli-Lebanon results while the red one is for the case study results.

The port results curve is greater than the case study results curve and all results rise when the number of containers increases (for example, the completion time of ship 6 is high due to the big number of containers).

The main target is to find an approach between the case study results and the port results.

C. Quay crane scheduling process example

TABLE III

SCHEDULING EXAMPLE WITH TWO QUAY CRANES AND 4 BAYS

Bays	B1	B2	B3	B4
Number of Containers	25	13	14	10
Unloading time per 1 cont.	1.17			
Quay cranes assignment	QC_1	QC_2	QC_2	QC_2
Completion time	29.25	15.21	16.38	11.7
Makespan	29.25	43.29		

We detail here a process example for 4 bays and two quay cranes with 62 containers (25 + 13 + 14 + 10), as shown in Table III. The resulting assignment of quay cranes is 1-2-2-2, which means that the first quay crane works in the first bay and the second quay crane works in the remaining bays. The completion time of first quay crane is 29.25 minutes and for the second one is 15.21+16.38+11.7 = 43.29 minutes. Makespan is the largest value between the completion time of all quay cranes, then the latest completion time between all bays is 43.29 minutes.

D. Quay cranes work order with non-crossing

TABLE IV

QUAY CRANES WORK ORDER EXAMPLES

No.	(Bx C)	Quay cranes work order
1	4 x 2	QC_1-QC_2-QC_2-QC_2
2	6 x 3	QC_1-QC_1-QC_1-QC_2-QC_2-QC_3
3	8 x 2	QC_1-QC_2-QC_1-QC_2-QC_1-QC_2-QC_2-QC_2
4	8 x 3	QC_1-QC_1-QC_1-QC_1-QC_2-QC_3-QC_2-QC_3

As shown in Table IV, it is clear that the cranes work without any interference between each other.

In the first instance two quay cranes are assigned to work in four bays; the best way of quay cranes assignment is 1-2-2-2, which means that the first quay crane works in the first bay and the second quay crane works in the bays 2,3 and 4.

In the second instance, 1-1-1-2-2-3 means that the first quay crane works in the first three bays, the second quay crane works in the 4th and 5th bay and finally the last quay crane works in the last bay.

In the third instance, 1-2-1-2-1-2-2-2 means that the first

quay crane works in the 1st, 3rd and 5th bay, and the second quay crane works in the 2nd, 4th, 6th, 7th and 8th bay.

As shown in all instances, we have no collision between quay cranes during assignments to the bays.

VI. CONCLUSION

This paper studied the case of a single vessel and multiple quay cranes; a mixed integer programming model is provided and a dynamic programming algorithm is proposed to solve the quay crane scheduling problem in the port. The model instances are validated dealing with the QCSP and compared to the results in the port of Tripoli.

In further research, this model will be used as a source to validate heuristic methods for the QCSP and a base for new model to solve the scheduling of quay cranes and yard trucks for unloading and loading containers.

VII. ACKNOWLEDGMENTS

REFERENCES

[1] C.F Daganzo, The quay crane scheduling problem, Transportation research ,1989, vol. 23 , pp 159-175.
[2] R.I Peterkofsky and C.F Daganzo, A branch and bound solution method for the quay crane scheduling problem, Transportation research, 1990, vol. 24, pp. 159–172.
[3] K.H Kim and Y.M Park, A crane scheduling method for port container terminals, European Journal of Operation Research, 2004, vol. 156, pp. 752–768.
[4] A. Lim, B. Rodrigues, F. Xiao and Y Zhu, Quay crane scheduling with spatial constraints, Naval Research Logistics, 2004, vol. 51, pp. 386–406.
[5] D Steeken and R Stahlbock, Container terminal operation and operations research - classification and literature review, OR Spectrum, 2004, vol. 26, pp. 3–49.
[6] Chengi Liang, Youfang Huang and Yang Yang, A quay crane dynamic scheduling problem by hybrid evolutionary algorithm for berth allocation planning, Computers and industrial engineering, 2008, vol. 56, pp. 1021–1028.
[7] Su Wang and Wenbin Hu, An investigation into berth and quay crane scheduling for container terminals based on knowledge, 2010 International Conference on Future Information Technology and Management Engineering, 2010.
[8] Ding Yi, Lin GuoLong and Liang ChengJi, Model and Heuristic Algorithm for Quay Crane Scheduling at Container Terminal, 2012 9th International Conference on Fuzzy Systems and Knowledge Discovery, 2012.
[9] Ali Diabat and Effrosyni Theodorou, An integrated quay crane assignment and scheduling problem, Computers and industrial engineering, 2014, vol. 73, pp. 115–123.
[10] Noura Al-Dhaheri and Ali Diabat, The quay crane scheduling problem, Journal of Manufacturing Systems, 2015, vol. 36, pp. 87–94.
[11] A. Nils Boysen, B. Dirk Briskorn and C. Frank Meisel, A generalized classification scheme for crane scheduling with interference, European Journal of Operational Research, 2016, vol. 000, pp. 1–15.
[12] Mohamad Kais Msakni, Ali Diabat, Ghaith Rabadi and Mariam Kotachi, An integrated quay crane assignment and scheduling problem using branch-and-price,2016 International Conference on Computational Science and Computational Intelligence, 2016.
[13] Jean Phelippe Ramos De Oliveira, Juan David Barbosa and Maria Lamprou, Multi-Objective Optimization of the quay crane assignment and scheduling problem: Time and movement optimization, 7th International Conference on Information, Intelligence, Systems and Applications (IISA), 2016.