IJE TRANSACTIONS A: Basics Vol. 32, No. 10, (October 2019) 1464-1479

International Journal of Engineering

Journal Homepage: www.ije.ir

A Mathematical Model and Grouping Imperialist Competitive Algorithm for Integrated Quay Crane and Yard Truck Scheduling Problem with Non-crossing Constraint

S. Behjat, N. Nahavandi*

Industrial & Systems Engineering Faculty, Tarbiat Modares University, Tehran, Iran

PAPER INFO

Paper history:
Received 23 June 2019
Received in revised form 13 August 2019
Accepted 12 Septemaber 2019

Keywords:
Grouping
Imperialist Competitive Algorithm
Mathematical Model
Non-crossing
Quay Crane Scheduling
Yard Truck Scheduling

ABSTRACT

In this research, an integrated approach is presented to simultaneously solve quay crane scheduling and yard truck scheduling problems. A mathematical model was proposed considering the main real-world assumptions such as quay crane non-crossing, precedence constraints and variable berthing times for vessels with the aim of minimizing vessels completion time. Based on the numerical results, this proposed mathematical model has suitable efficiency for solving small instances. Two versions of imperialist competitive algorithm (ICA) are presented to heuristically solve the problem. The grouping version of algorithm (G-ICA) is used to solve the large-sized instances based on considering the allocation of trucks as a grouping problem. Effectiveness of the proposed metaheuristics on small-sized problems is compared with the optimal results of the mathematical model. In order to compare the efficiency of the proposed algorithms for large-sized instances, several instances were generated and solved, and the performance of algorithms has been compared with each other. Moreover, a simulated annealing (SA) algorithm is developed to solve the problem and evaluate the performance of the proposed ICA algorithms. Based on the experimental results, the G-ICA has a better performance compared to the ICA and SA. Also an instance of a container terminal in Iran has been investigated which shows that the proposed model and solution methods are applicable in real-world problems.

doi: *10.5829/ije.2019.32.10a.16*

1. INTRODUCTION

Container terminals, especially in the recent years, play a significant role in worldwide transportation system. In 2017, based on UNCTAD review of maritime transport world container port throughput has increased up to 752 million 20-foot equivalent units (TEUs). In last three years, an average annual growth rate of 3.1% was reported [1]. Due to the rapid growth rate of transported container volume in the last decade, it is critical to improve the operational efficiency of container terminals .

In general, a container terminal consists of two main interfaces. These interfaces are quayside with berths for loading/unloading of vessels and landside to load or unload containers on trucks and store containers. As shown in Figure 1, there are three types of equipment involved in the loading and unloading process in a container terminal, including quay cranes (QCs), yard trucks (YTs) and yard cranes (YCs).

For inbound material flow, when a vessel arrives, quay cranes unload containers from the vessel, and transfer containers to landside for the storage through

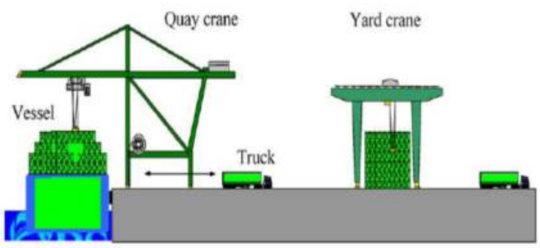


Figure 1. Schematic layout of a container terminal [2]

*Corresponding Author Email: *n_nahavandi@modares.ac.ir* (N. Nahavandi)

Please cite this article as: S. Behjat, N. Nahavandi, A Mathematical Model and Grouping Imperialist Competitive Algorithm for Integrated Quay Crane and Yard Truck Scheduling Problem with Non-crossing Constraint, International Journal of Engineering (IJE), IJE TRANSACTIONS A: Basics Vol. 32, No. 10, (October 2019) 1464-1479

yard trucks. However, for outbound containers, this process occurs in the opposite direction. While quay cranes are used to load outbound containers and unload inbound containers, yard trucks provide ground transportation of containers between the quayside and the landside.

Due to fierce competition among container ports to attract customers, the services should have an acceptable level of quality [3]. Rapid loading and unloading of containers and adhering to the promised scheduling program of ships are considered among the major factors in assessing the performance of container ports [4].

The major operations in container ports includes berth allocation problem (BAP), quay crane assignment problem (QCAP), quay crane scheduling problem (QCSP), ground transport equipment scheduling, yard crane scheduling and storage space allocation [5]. Berths are important resources in container ports that should be allocated to the vessels. The berth allocation problem determines the time window and location which is assigned to a vessel. Quay cranes are also a critical resource in the quayside that should be assigned to vessels in a productive manner. Effective solutions for quay crane scheduling may lead to reduction in loading and unloading procedure duration. This means that more vessels can be served and better utilization of the resources is expected. Yard trucks as another resources in container terminals are used to transfer containers between quay cranes and yard cranes. Yard trucks assignment and scheduling with the objective of coordination of YTs and QCs received many attention in literature.

The QCSP and the YT scheduling problems are highly interrelated. So it is needed to make an effective coordination among these equipment to prevent a situation when quay cranes or trucks need to wait for each other. Therefore, this study focuses on joint quay cranes and trucks scheduling problem in a container terminal. Both loading and unloading operations have been considered.

2. LITERATURE REVIEW

The studies related to QCSP and YT scheduling can be categorized into three major groups. First one is about researches considering the quay crane scheduling problem. Research on the QCSP started thirty years ago with the work conducted by Daganzo [6]. He proposed a model considering multiple vessels and multiple cranes for a set of vessels. Kim and Park [7] investigated the QCSP; in addition to presenting a method based on branch and bound (B&B) algorithm, they also developed a heuristic method called greedy randomized adaptive search procedure (GRASP). The above was later on modified by Moccia et al. [8], Bierwirth and Meisel [9]. Moccia et al. [8] developed a solution method based on branch and cut (B&C) algorithm and obtained significantly better solutions for the benchmark set provided by Kim and Park [7]. Bierwirth and Meisel [9] developed a B&B algorithm which allows only unidirectional movement of the QCs in order to search within a reduced solution space [9]. This algorithm provides better results in both objective function value and computation time when compared to the algorithms developed by Kim and Park [7] and Moccia et al. [8]. The model of Kim and Park [7] was extended by Tavakkoli-Moghaddam et al. [10] considering a set of vessels in parallel with the objective to minimize the total weighted completion time of QCs and the total weighted tardiness of ships and proposed a genetic algorithm (GA). Recent studies on the QCSP have focused on modeling problems considering the typical attributes of the QCSP in real world. These attributes include safety margin between quay cranes (e.g. Nguyen et al., [11]; Kaveshgar et al., [12]), non-crossing of quay cranes (e.g. Liu et al., [13]; Emde, [14]) and precedence relationships between containers (e.g. Kim and Park, [7]; Sammarra et al, [15]). Liang et al. [16] studied the QCSP with the aim of determining the sequence of processing containers on QCs and determining the optimal number of QCs that should be allocated to each vessel. They developed a loop iteration method with the aim of finding the optimum number of quay cranes that should be allocated to each vessel and the sequence of processing containers on QCs. A new mathematical model for QCSP is addressed by Sun et al. [17] to handle the non-crossing constraint in an easier manner. Moreover an exact problem-solving method based on Benders decomposition has been developed for solving the problem. Kasm and Diabat developed an exact and computationally efficient method [18] to solve QCSP considering non-crossing and safety margin assumptions for only one vessel. A two-step technique incorporating a partitioning heuristic and Branch and Price algorithm has been presented proposed for solving this problem. The numerical experiments show that the presented technique is suitable for solving the problem within a reasonable period of time. See Bierwirth and Meisel [19, 20] for a comprehensive survey of applications and optimization models for the operations management in container terminals.

The second category of researches in the literature are the researches considering the YT scheduling problem. Bish [21] considered the yard truck routing problem in a container terminal with the aim of minimization of the makespan of a given set of vessels. He has formulated the problem as a transshipment problem and proposed a heuristic algorithm [21]. Nishimura et al. [22] developed a new routing scheme for trailers at a container terminal. They also proposed a heuristic method called dynamic routing which reduces overall cost and trailer fleet size [22]. Ng et al. [23] investigated the trucks scheduling problem considering sequence-dependent processing times and different times of readiness for jobs. Further, a mathematical model and a GA have been developed for solving the problem [23]. Also there are numerous studies on the automated guided vehicles (AGVs) in automated container terminals [24, 25]. Implementation of AGVs instead of yard trucks can lead to higher efficiency and stability, cost reduction and safer environment in container terminals [26].

In the last years, a few studies have considered the interrelations between quay cranes operation and the trucks scheduling. Chen et al. [27] formulated the integrated operation of QCs, YTs, and YCs as a hybrid flowshop scheduling problem considering real world assumptions like precedence and blocking constraints. The problem has been simplified by assigning a given range of bays for each quay crane. They presented a metaheuristic algorithm for solving this problem. Cao et al. [28] studied the joint QC and YT scheduling problem for containers which are unloading and developed a mathematical model and a GA to solve the problem. Chen et al. [29] formulated the integrated QCs, YTs and YCs scheduling problem as a constraint programming model. They assumed a given assignment of the containers to QCs and proposed a three-stage algorithm to solve the problem. Tang et al. [5] proposed an integer programming model for the joint QC and truck scheduling problem. They developed several valid inequalities as well as two lower bounds for the model. A PSO algorithm has also been proposed for solving the problem and the solutions have been compared with the optimal results of mixed integer programming model and lower bounds for small sized and large sized problems, respectively.

Moreover, a mathematical model was developed by Kaveshgar and Huynh [30] for the joint QCs and YTs scheduling for unloading inbound containers. They considered real world limitations such as safety margin, precedence relationships between containers and quay crane non-crossing. In this study, the number of quay cranes which are allocated to each vessel has been assumed as a constant and specific number. He et al. [31] addressed the integrated QCs, YTs and YCs scheduling problem considering the transportation energy consumption as the objective. They presented a mathematical model and a solution method based on simulation-optimization. The results indicated that there is a trade-off between completion time of the vessels and energy consumption. Karam et al. [32] developed a mathematical model considering assignment of quay cranes and yard trucks simultaneously. This study investigates the allocation of QCs to vessels and assignment of yard trucks to each quay crane. The results of this research suggest that the presented model has led to improved average berthing stay per vessel. Vahdani et al. [33] proposed a bi-objective model to coordinate the assignment of vessels to the container terminals, assignment of QCs to vessels and assignment of YTs to quay cranes. They proposed two meta-heuristic algorithms based on GA and particle swarm optimization (PSO) algorithm. The computational results show that joint assignment of the above mentioned equipment results in reduction of the remaining workload in the container terminals.

Table 1 summarizes recent studies dealt with integrated quay crane and yard truck scheduling problem. In this research, the number of QCs assigned to each vessel has not been considered as a fix number. This assumption leads to a better utilization of quay cranes as an important resource in the container terminals. Moreover considering other constraints such as different berthing time for vessels causes more complexity in the problem.

TABLE 1. the summary of recent studies on joint QC and YT scheduling problem

Literature	QC assignment	QC scheduling	YT assignment	YT scheduling	Containers flow		Multiple vessels with different berthing time	Non-crossing
					inbound	outbound		
Chen et al. [27]	given range of bays for each QC	×	×	×	×	×		
Cao et al. [28]	Assumed given	only one QC considered	Fixed set of YTs for a QC	×	×			
Chen et al. [29]	Assumed given	×	×	×	×	×		×
Tang et al. [5]	×	×	×	×	×	×		×
Kaveshgar and Huynh [30]	Assumed given	×	×	×	×			×
He et al. [31]	Assumed given	×	×	×	×	×	×	×
Karam et al. [32]	given range of QCs for each vessel		given range of YTs for each QC		×	×	×	×
Vahdani et al. [33]	×		×		×	×	×	
This study	×	×	×	×	×	×	×	×

In this study, a novel integrated mixed integer programming model considering main real world assumptions is proposed based on flexible jobshop problem concept. To the best of our knowledge, there is no published work in the related literature that models this problem, considering the below mentioned assumptions, like the approach which is proposed in this study. Also a new grouping imperialist competition algorithm based on grouping concept is presented for solving this problem especially for the large-sized problems.

3. PROBLEM STATEMENT

Simultaneously quay cranes and yard trucks scheduling leads to better productivity in container ports. In terminal operations planning problems, usually a fixed number of trucks is assigned to each crane. Quay cranes are usually scheduled to process the inbound and outbound containers and then trucks are assigned to QCs. In this approach, the QCs and YTs may need to wait for each other. For example, consider the inbound process. If a container unloaded by a QC, but none of the YTs are ready to carry the container, the QC has to wait. In another way, if a QC has not finished unloading procedure of a container when a yard truck comes, the YT has to wait. Therefore, scheduling of these two resources (QCs and YTs) simultaneously may enhance the productivity of operations. The results have indicated that integrated methods lead to better solutions in comparison with non-integrated approach.

In this research, we specify allocation of containers to quay cranes and trucks. We also obtain the sequence and scheduling associated with processing of containers on quay cranes and trucks. The goal of the research is to find an integrated allocation and scheduling plan with the following assumptions to minimize the total completion time of ships:

- Each quay crane or truck serves only one container at the same time.
- The quay cranes can be allocated to any range of bays, but it should be considered that quay cranes cannot cross each other (non-crossing assumption).
- Once a quay crane starts working on a container, it will complete the process before starting another container.
- Both inbound and outbound containers have been considered.
- All of the vessels are not available in the beginning of the planning.
- There are precedence relationships among the containers. For example, containers in lower rows should be unloaded after the containers in upper rows.
- The number of quay cranes and trucks, process time of containers, berth allocation, container position in vessel and yard are considered as given.
- The same quay cranes and yard trucks have been established in the terminal.
- There is a sufficient number of yard cranes and there is not any bottleneck in the land-side.

The main objective in this study is finding a sequence of processing containers on QCs and YTs which leads to minimum total completion time of vessels. Table 2 is an example of a vessel with 8 bays and 6 containers and related processing times.

Figure 2 shows an example with two QCs, without considering non-crossing assumption (a) and with considering this assumption (b). As shown in this figure, considering non-crossing assumption leads to an idleness in the schedule.

Yard trucks operation can be handled in two ways: one-cycle strategy, where each truck serves a specific quay crane and two-cycle strategy, where trucks can work with different quay cranes and minimize the empty moves. The performance of yard trucks based on these two strategies has been shown schematically in Figure 3. In this study, in order to better utilize the yard trucks, the two-cycle strategy has been selected.

TABLE 2. Example 1

Job NO.	1	2	3	4	5	6
Related bay	1	2	4	6	7	8
Processing time	12	20	22	18	28	8

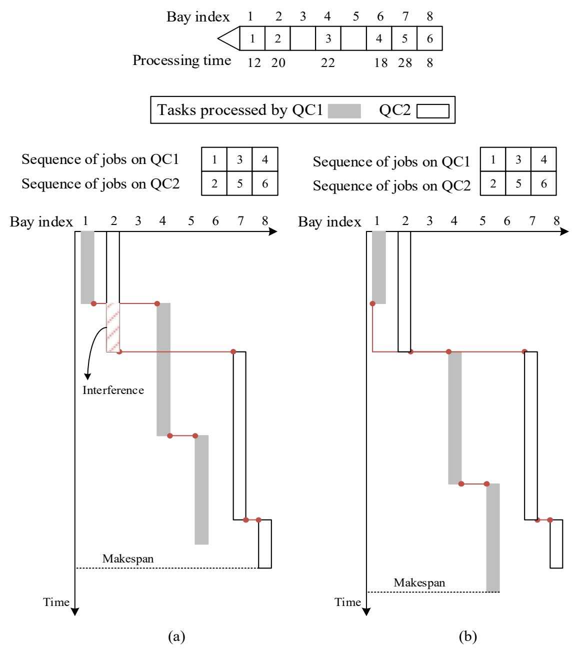


Figure 2. A solution for the example considering non-crossing

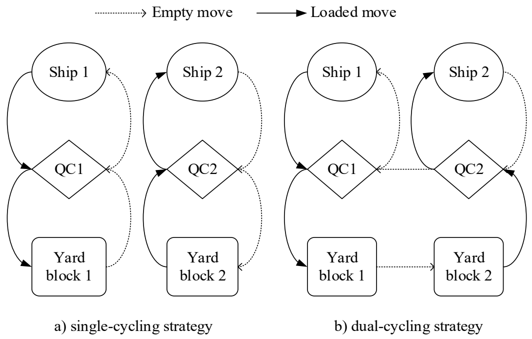


Figure 3. Yard truck operation strategy

4. MATHEMATICAL MODEL

In this study, the quay cranes and yard trucks scheduling problem has been formulated based on flexible jobshop problem concepts. In a flexible jobshop environment, n jobs with varying orders of process should be scheduled on m machines. The difference between flexible jobshop scheduling problems and jobshop scheduling problems is to allow each operation to be processed on one or more than one machine. The quay crane scheduling is a scheduling problem with spatial constraints, i.e. crane interference, which makes this problem more complex than the general jobshop problem.

A mathematical model is proposed in this study based on flexible jobshop concepts and considering the real world assumptions. Each container has two operations shown as O_{hj} , which means the h^{th} operation of container j. While for inbound containers, O_{1j} is unloading on a quay crane and O_{2j} is on a truck, outbound containers should be first carried by an YT and loaded on a vessel by a QC.

In this model, a dummy container, called reference container, is planned to be processed as the first container on each QC and YT. The parameters, decision variables, and the model are as follows:

4. 1. Parameters

m	Number of machines (sum of number of QCs and YTs)
n	Number of jobs (containers)
V	Number of vessels
i, j	Jobs index
l, h, l'	Operations index
k	Quay crane index
v	Vessels index
q_j	Location of job j
Ω	Set of precedence constrained tasks
Θ_v	Set of containers belonging to vessel v
J_1	Set of inbound containers
J_2	Set of outbound containers
Φ	Set of operations which should be processed on QCs
f_{hjk}	1, if machine k is capable to process the O_{hj} 0, o.w.
r_v	The readiness time of vessel v
M	A large number
p_{hjlik}	The process time of O_{hj} on machine k if it is processed immediately after O_{li} on machine k

4. 2. Variables

X_{hjlik}	1, if operation O_{hj} is processed immediately after operation O_{li} on machine k 0, o.w.	$h, l = 1,2$ $i, j = 1, \dots, n$ $k = 1, \dots, m$
s_{hj}	Start time of operation O_{hj}	$h = 1,2$ $j = 1, \dots, n$
c_{hj}	Completion time of operation O_{hj}	$h = 1,2$ $j = 1, \dots, n$
C_v	Completion time of the last job of vessel v	
Y_{hjk}	1, if operation O_{hj} is processed on machine k 0, o.w.	$h = 1,2$ $j = 1, \dots, n$ $k = 1, \dots, m$
Y'_{hjli}	1, if operation O_{hj} is processed after operation O_{li} (not immediately) 0, o.w.	$h, l = 1,2$ $i, j = 1, \dots, n$

4. 3. The model

$$Min \sum_{v=1}^{V} C_v$$

$$\sum_l \sum_{i=0} \sum_k X_{hjlik} = 1 \quad \forall h = 1,2 \quad \forall j = 1, \dots, n \tag{1}$$

$$\sum_h \sum_j \sum_k X_{hjlik} \le 1 \quad \forall l = 1,2 \quad \forall i = 1, \dots, n \tag{2}$$

$$\sum_l \sum_i X_{hjlik} \le f_{hjk} \quad \forall h = 1,2 \quad \forall j = 1, \dots, n \quad \forall k = 1, \dots, m \tag{3}$$

$$\sum_h \sum_j X_{hj10k} \le 1 \quad \forall k = 1, \dots, m \tag{4}$$

$$\sum_h \sum_j X_{hjlik} \le \sum_{l'} \sum_{i'} X_{lil'i'k} \quad \forall k = 1, \dots, m \tag{5}$$

$$c_{hj} \ge c_{(h-1)j} + \sum_l \sum_i \sum_k X_{hjlik} \cdot p_{hjlik} \quad \forall h = 1,2 \quad \forall j = 1, \dots, n \tag{6}$$

$$c_{hj} \ge c_{li} + \sum_k X_{hjlik} \cdot p_{hjlik} - M(1 - \sum_k X_{hjlik}) \quad \forall h, l = 1,2 \quad \forall i, j = 1, \dots, n \tag{7}$$

$$s_{hj} \ge c_{li} \quad \forall (i, j) \in \Omega \quad \forall h, l = 1,2 \tag{8}$$

$$\sum_l \sum_i X_{hjlik} = Y_{hjk} \quad h = 1,2;\ j = 1,\dots,n;\ k = 1,\dots,m \tag{9}$$

$$M\left(Y'_{hjli} + Y'_{lihj}\right) \geq \sum_k k \times Y_{hjk} - \sum_{k'} k' \times Y_{lik'} + 1 \quad h,l = 1,2;\ j,i = 1,\dots,n;\ \forall q_j < q_i;\ O_{hj}, O_{li} \in \Phi \tag{10}$$

$$s_{hj} + \sum_l \sum_i \sum_k X_{hjlik} \times p_{hjlik} = c_{hj} \quad h = 1,2;\ j = 1,\dots,n \tag{11}$$

$$s_{1j} \geq r_v \quad \forall v = 1,\dots,V;\ \forall j \in J_1 \cap \Theta_v \tag{12}$$

$$s_{2j} \geq r_v \quad \forall v = 1,\dots,V;\ \forall j \in J_2 \cap \Theta_v \tag{13}$$

$$c_{li} - s_{hj} + M \times Y'_{hjli} \geq 0 \tag{14}$$

$$c_{li} - s_{hj} - M \times (1 - Y'_{hjli}) \leq 0 \quad \forall h,l = 1,2;\ \forall i,j = 1,\dots,n \tag{15}$$

$$C_v \geq c_{hj} \quad v = 1,\dots,V;\ j \in \Theta_v;\ h = 1,2;\ O_{hj} \in \Phi \tag{16}$$

$$s_{hj}, c_{hj}, C_v \geq 0 \quad \forall h = 1,2,\ \forall j = 1,\dots,n,\ v = 1,\dots,V \tag{17}$$

As mentioned before, the objective function is minimization of the total completion time of the vessels. Constraint set (1) ensures that operation O_{hj} is processed after exactly one operation. Constraint set (2) guarantees that at most one operation can be processed after the previous completed operation. Constraint set (3) ensures that the operation O_{hj} is processed on the machine that is capable to process the operation. For example, the first operation of outbound containers ($O_{1j}, j \in J2$) cannot be processed on a quay crane. Based on constraint set (4), after the dummy jobs, at most one job can be processed. Constraint set (5) guarantees that if the operation O_{li} is not processed on machine *k*, any other operation cannot be processed after this operation on machine *k*. The completion time of the operation O_{hj} is calculated based on constraint set (6). Constraint set (7) ensures that each machine processes at most one operation at the same time. The precedence relationship between containers is considered in the constraint set (8). Constraint set (9) is incorporated into the model to determine which operation is processed on which machine.

Constraint set (10) is quay crane non-crossing constraints. For two containers, if $Y'_{hjli} + Y'_{lihj} = 0$ (means that O_{li} and O_{hj} are processed simultaneously), the QC that processes O_{hj} is on the left side of the QC which processes O_{li} if the location of container *j* is on the left side of the location of the container *i*. Constraint set (11) computes the start time of operations. Constraint sets (12) and (13) restrict the starting time of operations based on the earliest time when the related vessel is on the berth. Constraint sets (14) and (15) are incorporated into the model to compute the Y'_{hjli} variable, showing the simultaneous processing of two operations. Constraint set (16) calculates the completion time of the last container on the vessel *v*.

The proposed research problem in this study can be easily reduced to a minimum makespan jobshop scheduling problem by considering only a vessel with inbound containers. Based on Pinedo [34], this problem is known to be NP-hard in the strong sense. Thus, the proposed research problem is NP-hard too, and a suitable metaheuristic algorithm should be developed to solve the problem efficiently, especially for large sized problems [35].

5. ICA ALGORITHM

Imperialist competitive algorithm (ICA) is a metaheuristic optimization algorithms which has been recently developed based on sociopolitical concepts. This algorithm can be classified as evolutionary algorithms, first presented by Atashpaz-Gargari and Lucas [36].

Based on this algorithm, the problem solutions are defined as country, which improve over the iterative course of the algorithm, eventually giving the solution. The countries are divided into two groups: imperialist and colony countries. Each imperialist consists of an emperorship and several colony countries. ICA algorithm moves according to movement of colonies towards the imperialist as well as the competition between imperialist countries towards the optimal solution. During the algorithm, each country tends to move towards its imperialist. Further, imperialists tend to absorb the colonies of other countries towards themselves by empowering their own Empire [37].

5. 1. Primary Empires In this algorithm, we start with a number of initial population (called country). "N_imp" number of the solutions with minimum objective function (powerful countries) are selected as imperialist. Each imperialist country will have a number of other countries (N_col) as colony. The number of colonies of each imperialist will be determined according to its relative power (p_n) [38]:

$$p_n = \frac{C_n}{\sum_1^{n_{imp}} C_n} \tag{18}$$

$$C_n = c_n - \max_i c_i \tag{19}$$

As the problem is minimization, the power of each country will be calculated as $c_n - \max_i c_i$, where c_i is the value of objective function related to that solution. Further, in the proposed algorithm, the solutions (countries) are represented as follows:

The first section represents process sequence and allocation of containers on quay cranes, while the second section deals with process sequence and allocation of containers on the yard trucks. In the first part of each section, the process sequence is specified, and in the second part it is determined what equipment will process each container.

5. 2. Calculation of the Power of Countries The objective function of the solution should be calculated to evaluate the power of each country. As in the proposed ICA algorithm the sequence of tasks is determined, to calculate the objective function the schedule of processing the containers on machines should also be specified. Considering different real-world assumptions such as precedence/succession of task processing and quay cranes non-crossing, sequence does not easily lead to the schedule. a procedure has been proposed for this aim. Based on this, containers are allocated according to the earliest time available for QCs and YTs. Then, given the operational constraints that should be applied, the machines may have to wait (for example, for outbound jobs, the QC should wait, until the container operations on YT finish). This duration when the QC has to wait in spite of equipment availability is called forced idleness. These waiting times will be added to the time of start of jobs processing according to the proposed procedure. The details of how operational constraints are applied to the schedule of containers are described.

- **The second operation of each container should begin after completion of the first operations of that container.** Evidently, the machine related to process of the second operations of each container should wait until completion of the container processing time on the previous machine. For example, for inbound jobs, YT (even if it is ready and available) cannot initiate the container process operations and should wait until the container processing operations on QC is finished.
- **The starting time of each container on QC should be after the availability of that container (ship).** In this research, considering the time of availability of ships in the berth, the operations of containers on QC should be scheduled after entrance of the relevant ship to the berth.
- **Delay in start of operations can occur due to precedence relations.** In precedence relations (A,B), which are mostly determined based on arrangement of containers in ships, processing of the container B will not begin, unless operations of the container A have been finished. In this way, we may have a forced idleness on QC. This waiting time will be calculated according to completion time of container A.
- **Considering the constraint of non-crossing of cranes across each other, a QC may have to wait for some moment.** According to the Figure 4, assume that QC1 is assigned to bay 3 and is going to process a container in bay 12. If QC2 is processing containers in bay 8, then QC1 has to wait until completion of operations on this container. Accordingly, QC1 should be kept idle while a container in bay 12 is waiting to be processed.

When applying these constraints, some sequences may be infeasible due to the operational constraints. The objective function value for these solutions is assumed to be a large number.

5. 3. Inner Competition The original ICA algorithm for continuous optimization problems has been presented. As shown in Figure 5, the colony country moves by x units along the line connected to the imperialist with an angular deviation of θ.

In the proposed problem we are trying to determine the sequence of containers and allocate them into QCs and YTs. Accordingly, the algorithm should be modified in a way that it become applicable to this problem. To search the solution space, one point crossover and selection and replacement operators are used. Here, the colony country moves towards achieving the characteristics of the imperialist country.

According to crossover operator, one part of the solution of the imperialist country is chosen randomly and incorporated in the sequence of the colony country randomly. Note that this operator should generate feasible solutions.

Through selection and replacement operator, first we select a random number of elements we want to replace (r1). Then, we choose rl jobs among the imperialist's jobs and place them in that position. Thereafter, we embed the rest of relevant colony elements with the same order in the new solution such that feasibility of the solution is still preserved. When colonies are moving, some of them may achieve a better position than the imperialist, in which case these two countries swap their positions.

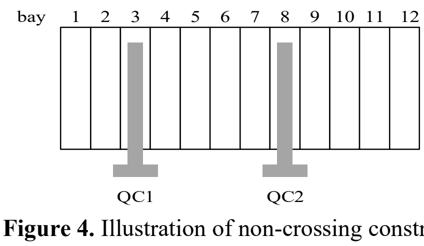


Figure 4. Illustration of non-crossing constraint

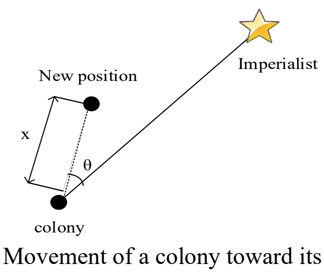


Figure 5. Movement of a colony toward its imperialist

5. 4. Diversifying Technique To prevent the algorithm from being entrapped in a local optimality point, a diversifying technique should be applied to the solutions. In the imperialist competitive algorithm, this technique is equivalent to occurrence of revolution in countries and altered components of that country.

According to the technique proposed in this research, in each iteration of the algorithm, each country may experience revolution with a specific probability, represented by prev. In case of revolution, all components of a solution are abandoned and a new solution will be generated randomly.

5. 5. Outer Competition The main competition in this algorithm occurs among different empires. In this competition, each imperialist is seeking to enhance its empire power and the number of colony countries. Power of each empire is determined based on the power of its imperialist and plus a proportion of its average colonies power.

$$total\ power(imp) = power(imperilasit) + \varepsilon \times mean(power(colonies\ of\ that\ imp)) \quad (20)$$

where ε is a positive coefficient that indicates participation proportion of colonies in the empire power.

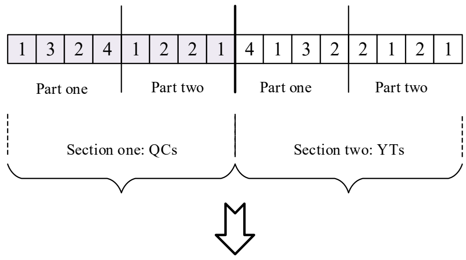
In each iteration of this algorithm, the weakest empire is determined, the weakest country of that empire is allocated to another empire. During the algorithm process, weak empires with reducing numbers of colonies will collapse, and eventually all countries will remain as an empire. The remaining imperialist will be the optimal or near to optimal solution resulting from the algorithm.

5. 6. Grouping Version of the Algorithm (G-ICA) In grouping optimization problems, the aim is to divide the members of set G to a number of different groups, such that each member belongs a group. The main assumption in these problems is that the order of groups is not important. There are various problems in combinatorial optimization, known as grouping problems, including graph coloring problem, bin packing problem, batch–machine scheduling problem and packing/partitioning problems.

In this study, allocation of trucks can be considered as a grouping problem. Here, the set of containers are divided into several groups, each of which is allocated to a yard truck. In this problem, the order of allocation of container groups to trucks will not be different.

As mentioned before, the components of each country consist of two main sections. As shown in Figure 6, the first section represents process sequence and allocation of containers on quay cranes (QC), however the section two shows allocating containers on yard trucks (YT). Figure 7 shows an example of grouping. In the grouping based representation, the second section consists of item part and group part. In the item part, it is specified to which group each container belongs. For example, a grouping problem with 2 containers, 2 quay cranes and yard trucks is represented as Figure 8.

In the grouping version of algorithm (G-ICA), to update the allocation and sequence of containers on QCs in each iteration, crossover as well as selection and replacement operators are used according to Figures 9 and 10. The container group's allocation and sequence on the YTs will be updated according to the method proposed by Kashan et al. [39] based on Figure 11.



QC No.	Contianer sequence	YT No.	Contianer sequence
1	1,4	1	1,2
2	3,2	2	4,3

Figure 6. Solution representation

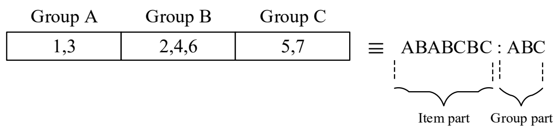


Figure 7. A sample of grouping

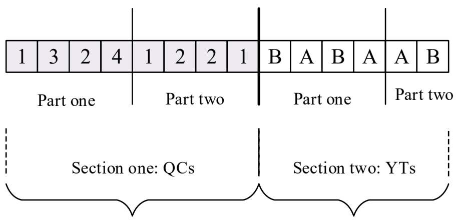


Figure 8. Grouping solution representation

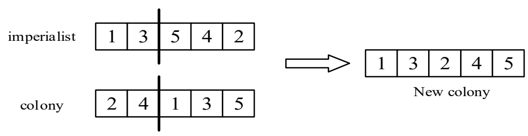


Figure 9. Crossover operator

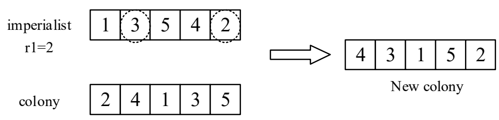


Figure 10. Selection and replacement operator

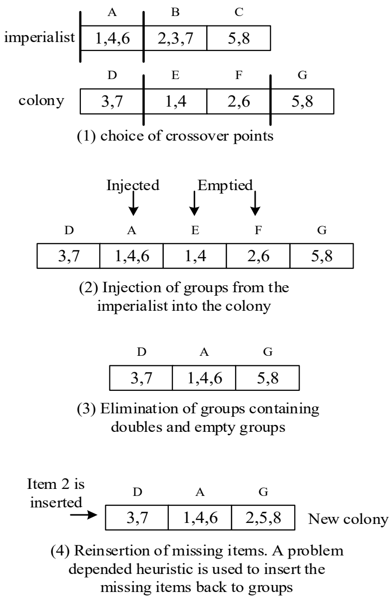


Figure 11. G-ICA crossover

6. NUMERICAL RESULTS

In order to investigate the efficiency of the presented mixed integer programming model as well as the performance of the proposed algorithms, 76 instances have been generated based on the random parameters proposed by Chen et al. [27] ranging from small to large-sized problems.

- The processing time of containers by QC: uniform random distribution [105,161]
- The departure time of trucks unloaded: uniform random distribution [60,130]
- The departure time of trucks loaded: uniform random distribution [100,170]
- The number of quay cranes: {2,3,4}
- The number trucks: {4,6,8,10,12,14}
- The number vessels: {2,3,4}
- The number of containers: {6,8,12,16,18,20,24,30,32,40,50,75}

The proposed mathematical model has been coded based on concert technology and by C++ using Microsoft visual studio 2010 and IBM ILOG CPLEX 12.6. The sample problems have been solved on a laptop with 2.53 GH CPU and 4 G RAM. Further, the proposed ICA algorithm has also been coded by Matlab 2015 software and solved on the same system.

6. 1. Parameters Tuning

The value of design parameters of the proposed ICA (number of countries, number of primary empires, ε and probability of revolution occurrence) affects the performance and results of the algorithm. Accordingly, to design a robust and efficient algorithm, the suitable level of the mentioned factors was determined. Response surface methodology (RSM) has been employed in this study to adjust the parameters related to the proposed algorithms. Table 3 reveals the different levels of each factor.

Considering the number of factors and level of each factor, 31 experiments were performed and based on the obtained objective function, a regression equation has been achieved. Using Minitab software and regression equation extraction, and optimizing it given the level of each factor, the optimal level of each parameter has been obtained. In this research, these values for the number of countries, number of primary empires, ε and probability of revolution occurrence will be $15n$, 25, 0.9, and 0.01, respectively.

6. 2. Performance Assessment

Objective function values obtained from the proposed ICA and solving the mathematical model by the CPLEX for small-sized problems are presented in Table 4. Due to NP-hard nature

TABLE 3. Different levels of algorithm parameters

Level	-2	-1	0	1	2
N_{col}	6n*	9n	12n	15n	18n
N_{imp}	5	10	15	20	25
ε	0.1	0.3	0.5	0.7	0.9
p_{rev}	0.01	0.0125	0.016	0.025	0.05

n=number of containers

TABLE 4. Comparison of the solution methods for small-sized problems

No.	Problem size*	CPLEX		ICA			G-ICA		
		Obj.	Time(s)	Obj.	Time(s)	Gap** (%)	Obj.	Time(s)	Gap (%)
1	6×2×4×2	676	2	680	7	0.59	676	8	0
2	6×2×6×2	617	5	617	9	0	617	11	0
3	6×2×8×2	648	4	648	11	0	648	15	0
4	8×2×4×2	816	58	819	16	0.36	824	17	0.98
5	8×2×6×2	761	38	784	22	3.02	769	21	1.05
6	8×2×8×2	833	153	859	27	3.12	845	25	1.44

* No. of containers × No. of vessels × No. of YTs × No. of QCs
** Gap= (obj. - CPLEX obj.)/CPLEX obj.×100

of the problem, solving large-sized problems within a reasonable time is not possible. Though, the proposed model is efficient for solving small-sized instances using CPLEX. The dimensions of the problem for each instance are determined based on the number of QCs, YTs and containers.

As reported in Table 4, the proposed G-ICA found the optimal solution for three out of six instances. For remaining 3 instances, the gap ranges from 0.98 to 1.44. Whereas the proposed ICA algorithm found the optimal solution for two of instances. Mean gap for the proposed ICA and G-ICA algorithm is 1.18 and 0.57, respectively. Results for larger instances with three and four QCs are shown in Tables 5-7. The first and second columns in Tables 5 and 6 show the experiment number and problem size. It is worthy to note that CPLEX runs are time-restricted and the best solution obtained in 2 hours is reported in this table.

If one the following criteria is occurred, the proposed algorithms will be stopped:

- Weak empires are collapsed and all countries remained as an empire.
- Algorithm reached maximum time limit of 210 seconds.
- An improvement does not observed during 30 consecutive iterations.

TABLE 5. Comparison of ICA and G-ICA against CPLEX for instances with 3 QCs

No.	Problem size*	CPLEX	ICA			G-ICA		
		Obj.	Obj.	Time(s)	Gap** (%)	Obj.	Time(s)	Gap (%)
7	12×2×6×3	921	889	27	-3.47	795	29	-13.68
8	16×2×6×3	1522	1344	53	-11.70	1233	46	-18.99
9	20×2×6×3	2686	2248	85	-16.31	2124	74	-20.92
10	18×3×6×3	2027	1769	71	-12.73	1655	69	-18.35
11	24×3×6×3	4669	3702	97	-20.71	3842	86	-17.71
12	30×3×6×3	4870	4143	76	-14.93	3656	90	-24.93
13	24×4×6×3	5912	4955	108	-16.19	4843	105	-18.08
14	32×4×6×3	6975	6149	81	-11.84	5953	74	-14.65
15	40×4×6×3	9995	8453	70	-15.43	8507	77	-14.89
16	12×2×8×3	808	808	36	0.00	805	35	-0.37
17	16×2×8×3	1544	1371	69	-11.20	1491	69	-3.43
18	20×2×8×3	2238	1941	75	-13.27	1726	83	-22.88
19	18×3×8×3	2884	2472	107	-14.29	2279	115	-20.98
20	24×3×8×3	3875	3421	112	-11.72	3147	116	-18.79
21	30×3×8×3	5323	4527	105	-14.95	4101	91	-22.96
22	24×4×8×3	5131	4138	66	-19.35	4356	77	-15.10
23	32×4×8×3	7080	6130	90	-13.42	5958	92	-15.85
24	40×4×8×3	9180	8106	103	-11.70	7728	104	-15.82
25	12×2×10×3	963	904	51	-6.13	862	53	-10.49
26	16×2×10×3	1643	1433	68	-12.78	1370	74	-16.62
27	20×2×10×3	1955	1675	108	-14.32	1559	105	-20.26
28	18×3×10×3	2646	1969	96	-25.59	1842	104	-30.39
29	24×3×10×3	3692	2884	114	-21.89	3102	98	-15.98
30	30×3×10×3	5548	4626	108	-16.62	3703	112	-33.26
31	24×4×10×3	5981	5055	95	-15.48	4072	109	-31.92
32	32×4×10×3	6367	5231	112	-17.84	5657	103	-11.15
33	40×4×10×3	8918	7849	123	-11.99	8065	117	-9.56

* No. of containers× No. of vessels × No. of YTs × No. of QCs
** Gap= (obj. - CPLEX obj.)/CPLEX obj.×100

TABLE 6. Comparison of ICA and G-ICA against CPLEX for instances with 4 QCs

No.	Problem size*	CPLEX	ICA			G-ICA		
		Obj.	Obj.	Time(s)	Gap** (%)	Obj.	Time(s)	Gap (%)
34	12×2×6×4	747	747	32	0.00	679	27	-9.10
35	16×2×6×4	1056	957	87	-9.38	945	63	-10.51
36	20×2×6×4	1741	1435	115	-17.58	1529	94	-12.18
37	18×3×6×4	1949	1753	95	-10.06	1571	81	-19.39
38	24×3×6×4	3518	3045	77	-13.45	2995	90	-14.87
39	30×3×6×4	4464	3752	74	-15.95	3525	76	-21.03
40	24×4×6×4	6296	5233	77	-16.88	4413	103	-29.91
41	32×4×6×4	6477	5241	81	-19.08	5460	86	-15.70
42	40×4×6×4	8621	7577	105	-12.11	6756	112	-21.63
43	12×2×8×4	876	851	45	-2.85	775	44	-11.53
44	16×2×8×4	1805	1530	69	-15.24	1304	63	-27.76
45	20×2×8×4	1979	1676	85	-15.31	1540	89	-22.18
46	18×3×8×4	2105	1911	117	-9.22	1842	99	-12.49
47	24×3×8×4	3193	2670	91	-16.38	2684	91	-15.94
48	30×3×8×4	5549	4685	97	-15.57	4153	110	-25.16
49	24×4×8×4	4646	3904	72	-15.97	3421	71	-26.37
50	32×4×8×4	6179	5190	99	-16.01	5363	91	-13.21
51	40×4×8×4	7894	6965	121	-11.77	6975	115	-11.64
52	12×2×10×4	883	844	73	-4.42	819	77	-7.25
53	16×2×10×4	1256	1128	95	-10.19	1172	93	-6.69
54	20×2×10×4	1635	1377	122	-15.78	1438	114	-12.05
55	18×3×10×4	2087	1798	84	-13.85	1680	85	-19.50
56	24×3×10×4	3979	2788	105	-29.93	3240	94	-18.57
57	30×3×10×4	4391	3717	100	-15.35	3405	119	-22.46
58	24×4×10×4	4651	4024	115	-13.48	3560	126	-23.46
59	32×4×10×4	6255	5537	120	-11.48	5024	111	-19.68
60	40×4×10×4	8448	7388	135	-12.55	6724	128	-20.41

* No. of containers× No. of vessels × No. of YTs × No. of QCs
** Gap= (obj. - CPLEX obj.)/CPLEX obj.×100

Table 7 compares the ICA and G-ICA performance for large size instances. From the results we observe that in 11 instances the G-ICA has a better performance regarding to ICA algorithm.

In this study for comparison among solution methods a comparative factor called Relative Percentage Deviation (RPD) has been used. This performance measure is determined based on the following equation where obj_i represents the value of objective function of the given algorithm for the *i*th instance and obj_i^{min} denotes the best objective function obtained for the i th instance. Also the number of instances is indicated by *n* [40]:

$$RPD = (\sum_i \frac{obj_i - obj_i^{min}}{obj_i^{min}})/n \times 100 \quad (21)$$

Table 8 shows the RPD of the proposed algorithms for larger instances with three and four QC.

An experimental design based on Montgomery is used to assess the performance of the developed algorithms [41]. The significance level is assumed to be 5% and the experiments coded with IBM SPSS Statistics software.

The comparison among the proposed algorithms has been conducted for three and four machine (QC) problems, separately (Tables 9 and 10).

TABLE 7. Performance of algorithms for large size instances

No.	Problem size*	ICA	G-ICA
		Obj.	Obj.
61	150×3×12×3	35369	31151
62	150×3×14×3	34781	29918
63	150×3×12×4	30571	26491
64	150×3×14×4	29837	25274
65	200×4×12×3	41298	39650
66	200×4×14×3	37092	38292
67	200×4×12×4	36790	35833
68	200×4×14×4	35861	32472
69	225×3×12×3	48251	50231
70	225×3×14×3	47790	45466
71	225×3×12×4	44544	42835
72	225×3×14×4	39582	41138
73	300×4×12×3	64088	63452
74	300×4×14×3	61704	62964
75	300×4×12×4	55712	56866
76	300×4×14×4	55352	53967

* No. of containers× No. of vessels × No. of YTs × No. of QCs

TABLE 8. The RPD for the presented algorithms

No. of QCs	No. of YTs	No. of instances	Algorithms RPD (%)	
			ICA	G-ICA
Three-QC	6	9	5.83	0.49
	8	9	5.35	1.55
	10	9	8.09	2.05
	12	4	4.67	1.02
	14	4	5.34	1.31
	total	*35*	*6.09*	*1.32*
Four-QC	6	9	6.85	1.19
	8	9	7.40	0.44
	10	9	5.81	2.72
	12	4	5.51	0.51
	14	4	7.76	0.98
	total	*35*	*6.67*	*1.29*

TABLE 9. ANOVA table for problems with three QCs

Effect	Degree of freedom	Sum of Squares	Mean Square	Sig.
Treatments	1	2418001	2418001	0.093<0.05
Blocks	34	24284002559	714235369	
Error	34	27568729	810844	
Total	69	24313989290		

Moreover according to the significance of *p-value* (less than 0.05) for four machine problems in Table 10, the G-ICA is better than the proposed ICA algorithm.

6. 3. Simulated Annealing (SA) Algorithm

For further evaluation of the proposed algorithms, the performance of the G-ICA has been compared with a general simulated annealing (SA) algorithm. The SA algorithm starts from an initial solution and consecutively moves to new neighboring solutions via algorithm loops. At every step, the simulated refrigeration considers a neighboring state and decides between proceeding to the new state and remaining the previous state probabilistically. If the new solution outperforms the current solution in terms of the value of objective function, the current solution is replaced by the new one. To avoid confinement by a local optimal solution, the algorithm accepts the new solution with a specific probability.

Table 11 shows the average deviation of the results of SA algorithm with the proposed G-ICA and ICA algorithms.

It shows that ICA and G-ICA algorithms performs better results than SA algorithm with average gap of 52 and 59%, respectively.

6. 4. Sensitivity Analysis

For further evaluation of the proposed metaheuristics, the impact of the number of YTs on the objective function was investigated.

TABLE 10. ANOVA table for problems with four QCs

Effect	Degree of freedom	Sum of Squares	Mean Square	Sig.
Treatments	1	4682211	4682211	0.019<0.05
Blocks	34	18800325791	552950758	
Error	34	26229447	771454	
Total	69	18831237450		

TABLE 11. Performance of the SA algorithm

No. of QCs	No. of YTs	No. of instances	Deviation (%)	
			(SA-ICA)/ICA	(SA-GICA)/GICA
Three-QC	6	9	52.99	62.05
	8	9	45.99	49.58
	10	9	44.51	55.62
	total	27	47.83	55.27
Four-QC	6	9	63.97	72.35
	8	9	53.32	64.00
	10	9	52.56	57.45
	Total	27	56.62	64.60

The value of objective function and utilization rate for YTs and QCs for different instances with 15 containers and a specific number of QCs in relation to different numbers of YTs is shown in Figure 12.

A significant relation between total completion time of vessels and the number of YTs is indicated based on this figure. Increasing the number of YTs leads to improvement in the value of objective function and better utilization of QCs which are main equipment of container terminals. However, as it is expectable, utilization rate of yard trucks decreases by increasing number of the YTs. It shows a tradeoff between more utilization of QCs and improvement in objective function and number of assigned YTs. Decision makers should find an optimal point which is the balance of cost of more YTs assignment and completion time related penalties.

In Figure 13, the total completion time obtained of G-ICA algorithm considering different number of quay cranes is shown. Each line in this figure shows the objective function value for different number of yard trucks. The increase in the number of quay cranes would not lead to a considerable improvement in the value of objective function. It shows that increasing number of yard trucks for this problem size leads to better performance rather than increasing number of quay cranes. However, this proposition should be tested for other instances with different number of jobs. This sensitivity analysis is useful to find out the bottleneck of loading/unloading process and determine the optimal number of QCs or YTs in a container terminal.

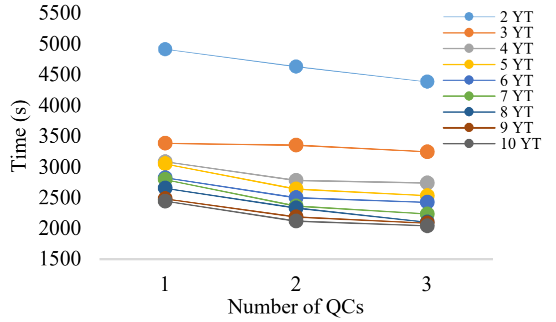


Figure 13. Effect of increasing number of QCs on objective function

Figure 12. Effect of increasing number of YTs on objective function and utilization of machines for instances with (a) 2 QCs, (b) 3 QCs and (c) 4 QCs

As mentioned in section 3, two-cycle strategy for yard trucks assignment has been considered in this study. The effectiveness of this approach in yard trucks operation has been examined. For this purpose 9 instances with two quay cranes and different number of yard trucks have been solved. The results which are reported in Figure14 shows that two-cycle strategy leads to faster loading and unloading procedure.

As shown in Figure14 using dynamic assignment of yard truck to quay cranes helps to find solutions with average 9.7% better objective function. Evidently the gap between the solutions of these two approaches is tighter for the instances with the more number of yard trucks.

6. 4. A Real World Case The proposed algorithms are applied on the real data of a port in Iran to show the applicability of the research problem. This port is located in the north of Persian Gulf with the capacity of about 5 million TEUs annually.

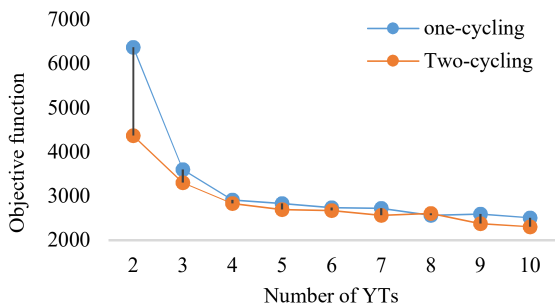


Figure 14. Comparison of the YT assignment strategies (two-cycling vs. one cycling)

The allocation of containers to QCs and YTs and also the schedule of processing of containers on QCs and YTs for a real case with the following input (Table 12) is determined.

The main assumptions of the proposed problem in this paper such as non-crossing and precedence are also practical in real world. The results are summarized in Table 13.

As shown in Table 13, the G-ICA has a better performance in comparison with the other proposed algorithms. Moreover the resources, including QCs and YTs, are utilized in a balanced manner. Based on the comments of experts of the port, the obtained schedule for processing the containers provides better solution to the above-mentioned instance compared to the regular scheduling procedure in the port.

TABLE 12. Parameters of the real case

Parameter	Value	
Number of QCs	6	
Number of YTs	30	
The processing time of containers by a QC	144	
The departure time of trucks	200	
Number of containers	365	
	Vessel A	**Vessel B**
Inbound containers	138	103
Outbound containers	0	124
Bays	19	19

TABLE 13. Analysis of the results for the real case

	SA	ICA	G-ICA
Objective function	48816	32256	31752
Completion time of vessel A	23328	15768	15696
Completion time of vessel B	25488	16488	16056
Maximum number of containers processed by a QC	67	68	68
Minimum number of containers processed by a QC	55	55	54
Maximum number of containers processed by a YT	28	22	19
Minimum number of containers processed by a YT	1	5	5

7. CONCLUSION

In this study, joint quay crane and yard truck scheduling problem in a container terminal was investigated. A mathematical model is extended for considering main real-world assumptions such as quay crane non-interference, precedence constraints and variable berthing times for vessels. Due to NP-hard nature of the problem, two versions of imperialist competitive algorithm (ICA) are proposed to heuristically solve the problem. The proposed metaheuristic algorithms can find optimal or near to optimal solutions, especially for small and medium size problems. Moreover, the G-ICA algorithm is proposed which is considering the YTs assignment problem as a grouping problem. The numerical results showed that G-ICA outperformed the ICA. Moreover, a simulated annealing (SA) algorithm is developed to solve the problem and evaluate the performance of the proposed ICA algorithms. It shows that performance of the proposed ICA algorithms is significantly better. Also an instance of a container terminal in Iran has been investigated which shows that the proposed model and solution methods are applicable in real world problems.

Considering uncertainty in processing times of containers on the QCs and YTs, non-identical equipment (QCs with different productivity rates) and considering multi objective optimization, are some of future studies that may be conducted.

8. REFERENCES

1. UNCTAD, Review of the maritime transport 2013, United Nations Conference on Trade and Development, Geneva: Switzerland, 2013.
2. Ng, W.C. and Mak, K. L., “Quay crane scheduling in container terminals”, ***Engineering Optimization***, Vol. 38, No. 6, (2006), 723–737.
3. Zegordi, S.H. and Nahavandi, N., “Measuring productivity indexes and efficiency in the container terminal at port Rajaei”, ***Scientia Iranica***, Vol. 9, No. 3, (2002), 248–254.
4. Tongzon, J. and Heng, W., “Port privatization, efficiency and competitiveness: Some empirical evidence from container ports (terminals)”, ***Transportation Research Part A: Policy and Practice***, Vol. 39, No. 5, (2005), 405–424.
5. Tang, L., Zhao, J. and Liu, J., “Modeling and solution of the joint quay crane and truck scheduling problem”, ***European Journal of Operational Research***, Vol. 236, No. 3, (2014), 978–990.
6. Daganzo, C. F., “The crane scheduling problem”, ***Transportation Research Part B: Methodological***, Vol. 23, No. 3, (1989), 159–175.
7. Kim, K.H. and Park, Y. M., “A crane scheduling method for port container terminals”, ***European Journal of Operational Research***, Vol. 156, No. 3, (2004), 752–768.
8. Moccia, L., Cordeau, J.F., Gaudioso, M. and Laporte, G., “A branch-and-cut algorithm for the quay crane scheduling problem in a container terminal”, ***Naval Research Logistics (NRL)***, Vol. 53, No. 1, (2006), 45–59.
9. Bierwirth, C. and Meisel, F., “A fast heuristic for quay crane scheduling with interference constraints”, ***Journal of Scheduling***, Vol. 12, No. 4, (2009), 345–360.
10. Tavakkoli-Moghaddam, R., Makui, A., Salahi, S., Bazzazi, M. and Taheri, F., “An efficient algorithm for solving a new

mathematical model for a quay crane scheduling problem in container ports", ***Computers & Industrial Engineering***, Vol. 56, No. 1, (2009), 241–248.

11. Nguyen, S., Zhang, M., Johnston, M. and Tan, K. C., "Hybrid evolutionary computation methods for quay crane scheduling problems", ***Computers & Operations Research***, Vol. 40, No. 8, (2013), 2083–2093.
12. Kaveshgar, N., Huynh, N. and Rahimian, S. K., "An efficient genetic algorithm for solving the quay crane scheduling problem", ***Expert Systems with Applications***, Vol. 39, No. 18, (2012), 13108–13117.
13. Liu, J., Wan, Y.W. and Wang, L., "Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures", ***Naval Research Logistics (NRL)***, Vol. 53, No. 1, (2006), 60–74.
14. Emde, S., "Optimally scheduling interfering and non-interfering cranes", ***Naval Research Logistics (NRL)***, Vol. 64, No. 6, (2017), 476–489.
15. Sammarra, M., Cordeau, J.F., Laporte, G. and Monaco, M. F., "A tabu search heuristic for the quay crane scheduling problem", ***Journal of Scheduling***, Vol. 10, No. 4–5, (2007), 327–336.
16. Liang, C., Fan, L., Xu, D., Ding, Y. and Gen, M., "Research on coupling scheduling of quay crane dispatch and configuration in the container terminal", ***Computers & Industrial Engineering***, Vol. 125, (2018), 649–657.
17. Sun, D., Tang, L. and Baldacci, R., "A Benders decomposition-based framework for solving quay crane scheduling problems", ***European Journal of Operational Research***, Vol. 273, No. 2, (2019), 504–515.
18. Kasm, O.A. and Diabat, A., "The quay crane scheduling problem with non-crossing and safety clearance constraints: An exact solution approach", ***Computers & Operations Research***, Vol. 107, (2019), 189–199.
19. Bierwirth, C. and Meisel, F., "A survey of berth allocation and quay crane scheduling problems in container terminals", ***European Journal of Operational Research***, Vol. 202, No. 3, (2010), 615–627.
20. Bierwirth, C. and Meisel, F., "A follow-up survey of berth allocation and quay crane scheduling problems in container terminals", ***European Journal of Operational Research***, Vol. 244, No. 3, (2015), 675–689.
21. Bish, E. K., "A multiple-crane-constrained scheduling problem in a container terminal", ***European Journal of Operational Research***, Vol. 144, No. 1, (2003), 83–107.
22. Nishimura, E., Imai, A. and Papadimitriou, S., "Yard trailer routing at a maritime container terminal", ***Transportation Research Part E: Logistics and Transportation Review***, Vol. 41, No. 1, (2005), 53–76.
23. Ng, W.C., Mak, K.L. and Zhang, Y. X., "Scheduling trucks in container terminals using a genetic algorithm", ***Engineering Optimization***, Vol. 39, No. 1, (2007), 33–47.
24. Hu, Z.H., Sheu, J.B. and Luo, J. X., "Sequencing twin automated stacking cranes in a block at automated container terminal", ***Transportation Research Part C: Emerging Technologies***, Vol. 69, (2016), 208–227.
25. Sidoti, D., Avvari, G.V., Mishra, M., Zhang, L., Nadella, B.K., Peak, J.E., Hansen, J.A. and Pattipati, K. R., "A multiobjective path-planning algorithm with time windows for asset routing in a dynamic weather-impacted environment", ***IEEE Transactions on Systems, Man, and Cybernetics: Systems***, Vol. 47, No. 12, (2016), 3256–3271.
26. Yang, Y., Zhong, M., Dessouky, Y. and Postolache, O., "An integrated scheduling method for AGV routing in automated container terminals", ***Computers & Industrial Engineering***, Vol. 126, (2018), 482–493.
27. Chen, L., Bostel, N., Dejax, P., Cai, J. and Xi, L., "A tabu search algorithm for the integrated scheduling problem of container handling systems in a maritime terminal", ***European Journal of Operational Research***, Vol. 181, No. 1, (2007), 40–58.
28. Cao, J., Shi, Q. and Lee, D. H., "Integrated quay crane and yard truck schedule problem in container terminals", ***Tsinghua science and technology***, Vol. 15, No. 4, (2010), 467–474.
29. Chen, L., Langevin, A. and Lu, Z., ", 2013. Integrated scheduling of crane handling and truck transportation in a maritime container terminal", ***Research, European Journal of Operational***, Vol. 225, No. 1, (2013), 142–152.
30. Kaveshgar, N. and Huynh, N., "Integrated quay crane and yard truck scheduling for unloading inbound containers", ***International Journal of Production Economics***, Vol. 159, (2015), 168–177.
31. He, J., Huang, Y., Yan, W. and Wang, S., "Integrated internal truck, yard crane and quay crane scheduling in a container terminal considering energy consumption", ***Expert Systems with Applications***, Vol. 42, No. 5, (2015), 2464–2487.
32. Karam, A., ElTawil, A.B. and Harraz, N. A., "Simultaneous assignment of quay cranes and internal trucks in container terminals", ***International Journal of Industrial and Systems Engineering***, Vol. 24, No. 1, (2016), 107–125.
33. Vahdani, B., Mansour, F., Soltani, M. and Veysmoradi, D., "Bi-objective optimization for integrating quay crane and internal truck assignment with challenges of trucks sharing", ***Knowledge-Based Systems***, Vol. 163, (2019), 675–692.
34. Pinedo, M., Scheduling, Vol. 29, New York: Springer, 2012.
35. Yegane, B.Y., Kamalabadia, I.N. and Khanlarzadeb, N., "Critical Path Method for Lot Streaming Problem in Flexible Job Shop Environment", ***International Journal of Engineering - Transactions B: Applications***, Vol. 30, No. 2, (2017), 261–269.
36. Atashpaz-Gargari, E. and Lucas, C., "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition", In 2007 IEEE congress on evolutionary computation, IEEE, (2007), 4661–4667.
37. Hosseinirad, S. M., "A Hierarchy Topology Design Using a Hybrid Evolutionary Algorithm in Wireless Sensor Networks", ***International Journal of Engineering - Transactions A: Basics***, Vol. 31, No. 10, (2018), 1651–1658.
38. Azizmohammadi, R., Amiri, M., Tavakkoli-Moghaddam, R. and Mohammadi, M., "Solving a redundancy allocation problem by a hybrid multi-objective imperialist competitive algorithm. , 26(9), pp..", ***International Journal of Engineering - Transactions C: Aspects***, Vol. 26, No. 9, (2013), 1031–1042.
39. Kashan, A.H., Kashan, M.H. and Karimiyan, S., "A particle swarm optimizer for grouping problems", ***Information Sciences***, Vol. 252, (2013), 81–95.
40. Behjat, S. and Salmasi, N., "Total completion time minimisation of no-wait flowshop group scheduling problem with sequence dependent setup times", ***European Journal of Industrial Engineering***, Vol. 11, No. 1, (2017), 22–48.
41. Montgomery, D.C., Design and analysis of experiments, John wiley & sons, 2017.

A Mathematical Model and Grouping Imperialist Competitive Algorithm for Integrated Quay Crane and Yard Truck Scheduling Problem with Non-crossing Constraint

S. Behjat, N. Nahavandi

Industrial & Systems Engineering Faculty, Tarbiat Modares University, Tehran, Iran

PAPER INFO

Paper history:
Received 23 June 2019
Received in revised form 13 August 2019
Accepted 12 Septemaber 2019

Keywords:
Grouping
Imperialist Competitive Algorithm
Mathematical Model
Non-crossing
Quay Crane Scheduling
Yard Truck Scheduling

چکیده

در تحقیق پیش رو یک رویکرد یکپارچه به منظور حل همزمان زمان‌بندی جرثقیل‌های اسکله‌ای و کامیون‌های محوطه ارائه شده است. یک مدل ریاضی با در نظر گرفتن فرضیات دنیای واقعی مانند عدم عبور جرثقیل‌های اسکله‌ای از یکدیگر، روابط پیشنیازی و چندین کشتی با زمان پهلوگیری مختلف با هدف کمینه کردن مجموع زمان تکمیل کارها ارائه شده است. براساس آزمایشات عددی، مدل پیشنهادی برای حل مسائل در ابعاد کوچک از کارایی مناسبی برخوردار است. با توجه به ماهیت NP-hard مساله، الگوریتم رقابت استعماری (ICA) برای حل مساله به صورت فراابتکاری ارائه شده است. همچنین یک الگوریتم رقابت استعماری مبتنی بر مفهوم گروه‌بندی (G-ICA) نیز برای حل مساله توسعه داده شده است. عملکرد الگوریتم‌های پیشنهادی برای مسائل با ابعاد کوچک، با جواب‌های حاصل از مدل ریاضی و برای مسائل با ابعاد بزرگ با یکدیگر مقایسه شده است. همچنین یک الگوریتم شبیه‌سازی تبرید (SA) به منظور ارزیابی کیفیت جواب‌های الگوریتم‌های پیشنهادی توسعه داده شده است. براساس نتایج عددی و آزمایشات آماری الگوریتم رقابت استعماری مبتنی بر مفهوم گروه‌بندی عملکرد بهتری نسبت به الگوریتم رقابت استعماری ساده وشبیه‌سازی تبرید دارد. علاوه بر این یک مساله واقعی از یک بندر در ایران نیز مورد بررسی قرار گرفته که نتایج حاصله نشان می‌دهد، مدل و روش‌های حل پیشنهادی در این تحقیق در مسائل دنیای واقعی قابل استفاده و پیاده‌سازی می‌باشند.

***doi**: 10.5829/ije.2019.32.10a.16*