

Research Article

Multi Objective Optimization of Coordinated Scheduling of Cranes and Vehicles at Container Terminals

Seyed Mahdi Homayouni¹ and Sai Hong Tang²

¹ Department of Industrial Engineering, Lenjan Branch, Islamic Azad University, Zarinshahr, Esfahan 84741, Iran

² Department of Mechanical and Manufacturing Engineering, Faculty of Engineering, Universiti Putra Malaysia, UPM Serdang, 43400 Selangor, Malaysia

Correspondence should be addressed to Seyed Mahdi Homayouni; homayouni@iauln.ac.ir

Received 20 March 2013; Revised 28 May 2013; Accepted 1 June 2013

Academic Editor: Wen-Jer Chang

Copyright © 2013 S. M. Homayouni and S. H. Tang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

According to previous researches, automated guided vehicles and quay cranes in container terminals have a high potential synergy. In this paper, a mixed integer programming model is formulated to optimize the coordinated scheduling of cranes and vehicles in container terminals. Objectives of the model are to minimize total traveling time of the vehicles and delays in tasks of cranes. A genetic algorithm is developed to solve the problem in reasonable computational time. The most appropriate control parameters for the proposed genetic algorithm are investigated in a medium size numerical test case. It is shown that balanced crossover and mutation rates have the best performance in finding a near optimal solution for the problem. Then, ten small size test cases are solved to evaluate the performance of the proposed optimization methods. The results show the applicability of the genetic algorithm since it can find near optimal solutions, precisely and accurately.

1. Introduction

The containers have revolutionized the seaborne transportation industry. In the past decade, massive containerships and ever increasing demand for containerization had forced the seaport container terminals (CTs) to expand their capacity in loading and unloading the containerships. Imai et al. [1] emphasized that mega containerships have started to handle more than ten thousand of containers since 2007. Commonly, a predetermined time schedule is set for any visiting containership being loaded/unloaded, where violating this schedule is too costly for the container terminals. Therefore, most of the CT authorities seek to improve their performance, by either implementing modern handling equipment or improving their management, planning, and scheduling skills.

CT authorities all over the world are trying to use more enhanced equipment, such as automated guided vehicles (AGVs) and automated stacking cranes (ASCs) which serve the quay cranes (QCs). For example, more than 250 AGVs and 140 ASCs are in service within Europe container terminal (Delta terminal) [2]. Key factors for a CT are efficiency of the stacking and transport of a huge number of containers

to (from) the quayside. Kim and Park [3] stated that one direction to improve productivity of the CTs is to enhance the degree of automation of their handling equipment.

In most of the cases both hardware and software improvements are necessary to decrease waiting times in container terminals. Stahlbock and Voß [4] declared that existing infrastructures and equipment of the CTs can be used more efficiently, for example, by means of powerful information technology and logistics control software systems including optimization methods. According to the literature, coordinated scheduling is a method to improve the performance of existing equipment [5–7]. Lau and Zhao [6] emphasized that the main loss of performance at ACTs is the schedules of various equipment, which are uncoordinated.

Optimization of the coordinated scheduling of cranes and vehicles in CTs is the main concern of this paper. The remainder of the paper is organized as follows, to pursue this aim. A brief literature review of the most related researches is presented in Section 2. A mixed integer programming (MIP) model for this problem is described in Section 3. Next, a genetic algorithm (GA) is proposed and detailed in Section 4 to solve the problem. To compare the results and show the

applicability of the proposed GA method, results of numerical test cases are reported in Section 5. Finally, Section 6 provides the concluding remarks of the research.

2. The Literature Review

Various scheduling problems for handling equipment in CTs have been widely considered in the past decade. For example, the goal of QC scheduling problem is to determine the sequence of unloading/loading tasks, assigned to a crane so that the completion time of a containership tasks is minimized. Kim and Park [3] proposed an MIP model to schedule the cranes. They stated that integrating decision-making process at various levels in CTs improve the efficiency of its scheduling. Liu et al. [8] studied the scheduling of cranes at CTs where incoming containerships have different ready times. They tried to minimize the maximum tardiness of containership departure. The scheduling of cranes is highly affected by availability of the vehicles. The researchers stated that coordinated scheduling of cranes and vehicles might improve the efficiency of CTs [6].

The scheduling of vehicles in CTs and flexible manufacturing systems is similar. Researchers proposed to schedule machines and AGVs simultaneously [9, 10], though vehicle scheduling in container terminals has gained more attention in recent years [6, 11, 12]. In CTs, automated guided vehicles operate in single-load carrier mode. The particular difficulty of AGV dispatching in a CT is that AGV pickup and drop-off times for each container have to coincide with scheduling of the quay and stacking cranes to avoid idle times of these equipment and to guarantee short berthing time of the containerships [13]. According to Lau and Zhao [6], AGVs deliver their loads to buffers once they reached the assigned destination in manufacturing systems. Nevertheless, the AGVs are not able to pick up/drop off containers by themselves. Therefore, there is no buffer between QCs and AGVs in container terminals. This causes a tight and precise schedule for AGVs and QCs. Moreover, in contrast to the manufacturing areas, the large number of AGVs is implemented at container terminals, and outdoor transport systems make it more difficult to schedule, efficiently.

Coordinated scheduling of cranes and vehicles in CTs has been studied in the literature. Lee et al. [7] considered integrated scheduling of cranes and yard trucks. They formulated an MIP model for this problem similar to the two-stage flexible flow shop with sequence-dependent setup time. The objective function includes minimizing the makespan of all the cranes' tasks in the scheduling period. They proposed a GA to obtain near optimal solutions of the problem. They only considered scheduling of unloading tasks; however, they stated that the procedure can be applied to loading tasks, as well. Moreover, Nguyen and Kim [14] proposed coordinated scheduling for cranes and automated lifting vehicles in container terminals. They developed an MIP model in addition to a heuristic algorithm to solve the problem. The objective function is to minimize total travel time of vehicles and total delays in tasks of cranes. As the automated lifting vehicles are able to pick up the containers by themselves, Nguyen and

Kim [14] demonstrated that buffer capacity highly influences performance of this integrated scheduling method.

Cao et al. [15] have formulated an MIP model for the integrated crane and yard truck scheduling, which has been solved by using a GA. In a recent research, Zhao and Tang [16] considered the integrated scheduling of trucks and cranes. They developed a particle swarm optimization algorithm to solve the problem. Their objective was to minimize makespan of unloading tasks for a specific containership. Assuming empty travels of the trucks when they come back from the yard to the quay area is the main drawback of their work. The same objective function has been minimized by Homayouni et al. [17]. They developed a simulated annealing algorithm to find near optimal solutions for the integrated scheduling of vehicles and cranes in container terminals.

Moreover, Skinner et al. [18] presented a GA to improve container handling operations at a container terminal in Australia. Jobs related to QCs, straddle carriers, and trucks (for importing or exporting containers to the land side of the terminal) have been scheduled. They have formulated the problem as a mixed integer linear programming model. They have applied this approach in a trial basis live scheduling in the aforementioned container terminal. Chen et al. [19] have considered the simultaneous scheduling of QCs and vehicles. They formulated problem as a constraint programming model. They also have developed a heuristic three-stage solution method to find high-quality solutions and can efficiently solve large problems.

Most of the literature reviewed in this section has assumed that unloading tasks for any of the containerships should precede the loading tasks (e.g., [7, 16, 19]). However, this process may increase empty travelling time of the vehicles while they come back to the docking area after an unloading task. According to Lau and Zhao [6], the loading and unloading tasks can be mixed to overcome this main limitation of the problems. On the other hand, these literatures have developed mathematical formulations which are not able to calculate the exact real completion time of the tasks. These two main limitations are considered while formulating the problem in this research.

3. Mathematical Modeling of the Problem

The coordinated scheduling of the cranes and the vehicles is formulated as a mixed integer programming model in this section. The general scenario for the containership operations consists of unloading and loading tasks. The scheduling horizon is usually divided to some predetermined scheduling periods. In each period, based on the precedence relations, a list of tasks is planned for each crane.

In unloading tasks, the vehicle starts a journey from its dwell point (where the last task of the vehicle finished) to the assigned crane. Similarly, after its previous task, the crane handles the container from its location in the containership to the pickup/delivery (P/D) point in the quayside of the CT. Once the vehicle reaches P/D point, and if the crane reached there too, the container will be loaded on the vehicle. Afterward, the vehicle starts a journey to the assigned load/unload (L/U) station in the storage yard. The vehicle

TABLE 1: Calculations for travelling time of the vehicles.

T_{ki}	T_{lj}	t_{ki}^{lj}	T_{ki}	T_{lj}	t_{ki}^{lj}
L	L	$= C_k \rightarrow L/U_{lj} + L/U_{lj} \rightarrow C_l$	S	L	$= L/U_{Si} \rightarrow L/U_{lj} + L/U_{lj} \rightarrow C_l$
L	U	$= C_k \rightarrow C_l$	S	U	$= L/U_{Si} \rightarrow C_l$
U	L	$= C_k \rightarrow L/U_{ki} + L/U_{ki} \rightarrow L/U_{lj} + L/U_{lj} \rightarrow C_l$	L	F	$= C_k \rightarrow L/U_{Fj}$
U	U	$= C_k \rightarrow L/U_{ki} + L/U_{ki} \rightarrow C_l$	U	F	$= C_k \rightarrow L/U_{ki} + L/U_{ki} \rightarrow L/U_{Fj}$

delivers the container to L/U station and waits for next assigned operation.

Likewise, in the loading tasks, the vehicle moves from its dwell point to the assigned L/U station, receives the container, and moves to P/D point at the crane. If the crane is free, the container is loaded to the containership. In case that crane is not free, the vehicle has to wait for it. The AGVs are not able to pick up the containers by themselves. Therefore, they directly should be loaded or unloaded by either the crane or the L/U station. It is why there is no buffer space between the crane or L/U stations and the AGVs in container terminals.

It is assumed that handling time between various equipment of the CT is deterministic and predetermined. Congestions of the vehicles in the guide path are not considered in this scenario. Moreover, it is assumed that the vehicles start their tasks from a predetermined L/U stations and return there once finishing all their assignments. There is no acceleration for the vehicles and the velocity of loaded and empty vehicles are equal; therefore, their travelling time is the same.

In the proposed scenario for the coordinated scheduling of cranes and vehicles, T_{ki} represents i th task of the k th crane (C_k). If T_{ki} and T_{lj} are performed consecutively, by using the same vehicle, t_{ki}^{lj} denotes the travelling time of the vehicle from C_k , in T_{ki} , to the C_l in T_{lj} . Table 1 presents formulas to calculate t_{ki}^{lj} based on the specifications of both T_{ki} and T_{lj} . The initial and final events of the i th vehicle are shown by T_{Si} and T_{Fi} , in which “S” and “F” represent the set of initial and final events of the vehicles. In Table 1, L/U_{ki} shows the dedicated L/U station for T_{ki} . Moreover, “L” and “U” represent the sets of loading and unloading tasks, respectively. As an example, if T_{ki} is an unloading task and T_{lj} is a loading one, t_{ki}^{lj} is the travelling time for the vehicle, which starts the journey from C_k and delivers the container to L/U_{ki} , which is the destination of T_{ki} . Next, the vehicle moves from L/U_{ki} to L/U_{lj} , which is the starting point of T_{lj} , receives the container, and moves to C_l , which is the destination of T_{lj} .

In the proposed model, the set of cranes is denoted by K , the number of tasks assigned to C_k is shown by m_k , and the set of the vehicles is shown by V . ϕ_{ki}^{lj} is a binary decision variable, which becomes one if T_{lj} is successively performed after T_{ki} by using the same vehicle, and zero, otherwise. ϕ_{Si}^{lj} denotes a journey started by V_i from its initial position, where $i \in \{1, \dots, V\}$ to the starting point of T_{lj} . Similarly, ϕ_{ki}^{Fj} denotes a journey ended by V_j from destination of T_{ki} to the final position of V_j , $j \in \{1, \dots, V\}$. Moreover, y_{ki} is the real time in which C_k finishes its operation in T_{ki} (place

the container in its predefined location in the containership in loading tasks, and drop the container into the assigned vehicle in unloading tasks). The earliest possible completion time for operations of C_k in T_{ki} is represented as S_{ki} . To calculate S_{ki} it is assumed that vehicles are ready to deliver or receive the container to the crane. The proposed mixed integer programming model for the coordinated scheduling of cranes and vehicles is presented in the following

Minimize

$$Z = \alpha \left(\sum_{k \in \{SUK\}} \sum_{i=1}^{m_k} \sum_{l \in \{KUF\}} \sum_{j=1}^{m_l} t_{ki}^{lj} \phi_{ki}^{lj} \right) + \beta \left(\sum_{k=1}^K (y_{km_k} - S_{km_k}) \right), \quad (1)$$

Subject to

$$\sum_{l \in \{KUF\}} \sum_{j=1}^{m_l} \phi_{ki}^{lj} = 1 \quad \text{for } \forall k \in S \cup K, \quad (2)$$

$$i = 1, \dots, m_k,$$

$$\sum_{k \in \{SUK\}} \sum_{i=1}^{m_k} \phi_{ki}^{lj} = 1 \quad \text{for } \forall l \in K \cup F, \quad (3)$$

$$j = 1, \dots, m_l,$$

$$y_{lj} \geq y_{l(j-1)} + 2TC + OC \quad \text{for } \forall l \in K \cup F,$$

$$j = 2, \dots, m_l, \text{ if } (T_{lj}, T_{l(j-1)} \in L) \quad (4)$$

$$\text{or } (T_{lj}, T_{l(j-1)} \in U),$$

$$y_{lj} \geq y_{l(j-1)} + TC + OC \quad \text{for } \forall l \in K \cup F,$$

$$j = 2, \dots, m_l, \text{ if } (T_{lj} \in L, T_{l(j-1)} \in U) \quad (5)$$

$$\text{or } (T_{lj} \in U, T_{l(j-1)} \in L),$$

$$y_{lj} - (y_{ki} + t_{ki}^{lj}) \geq M(\phi_{ki}^{lj} - 1)$$

$$\text{for } \forall k \in K \cup S, l \in K \cup F, i = 1, \dots, m_k,$$

$$j = 1, \dots, m_l, \text{ if } (T_{lj}, T_{ki} \in L)$$

$$\text{or } (T_{lj}, T_{ki} \in U), \quad (6)$$

$$y_{lj} - (y_{ki} + t_{ki}^{lj} + TC + OC) \geq M(\phi_{ki}^{lj} - 1) \quad (7)$$

$$\text{for } \forall k \in K \cup S, l \in K \cup F, i = 1, \dots, m_k,$$

$$j = 1, \dots, m_l, T_{lj} \in L, T_{ki} \in U,$$

$$y_{lj} - (y_{ki} + t_{ki}^{lj} - TC - OC) \geq M(\phi_{ki}^{lj} - 1) \quad (8)$$

$$\text{for } \forall k \in K \cup S, l \in K \cup F, i = 1, \dots, m_k,$$

$$j = 1, \dots, m_l, T_{lj} \in U, T_{ki} \in L,$$

$$y_{ki} - y_{k(i-1)} \geq S_{ki} - S_{k(i-1)} \quad \text{for } \forall k \in K, \quad (9)$$

$$i = 2, \dots, m_k,$$

$$y_{ki} \geq S_{ki} \quad \text{for } \forall k \in K, i = 1, \dots, m_k \quad (10)$$

$$\phi_{ki}^{lj} = 0 \text{ or } 1 \quad \text{for } \forall k \in S \cup K, \forall l \in K \cup F, \quad (11)$$

$$i = 1, \dots, m_k, j = 1, \dots, m_l.$$

The objective function (1) considers minimizing total travelling time for all the vehicles, in addition to minimizing total delays of the tasks of cranes compared to their earliest possible completion time. α is a constant which shows the cost of travelling time of the vehicles, and β is the cost for delays in crane operations, and α and β are set to 0.1 and 0.8, respectively, according to the assumptions of Lau and Zhao [6]. Moreover, Nguyen and Kim [14] emphasized that objective of minimizing delays of tasks of cranes is more important than decreasing total travel time of vehicles.

In the constraints section, (2) states that only one task should be started directly after any task by using the same vehicle, while (3) states that any task is preceded by one task by using the same vehicle. Constraints (4) to (8) calculate the real completion time of T_{lj} on C_l (i.e., y_{lj}). If both consecutive tasks of a crane are loading ones, the crane should travel to the quayside after placing the container on the containership in previous tasks of crane or T_{lj-1} (takes TC time units). After receiving the container from the vehicle, the crane moves to the containership (takes TC time units) and places the container on the containership (takes OC time units). Therefore, y_{lj} should be greater than y_{lj-1} by " $2TC + OC$." The same occasion happens when both consecutive tasks of a crane are unloading ones. This is represented in constraint (4). Unevenly, if two consecutive tasks of a crane are not of the same type, the crane can start next task immediately. Thus, y_{lj} is greater than y_{lj-1} by " $TC + OC$," which is shown in constraint (5).

On the other hand, y_{lj} is related to the arrival time of the vehicle after performing T_{ki} (takes t_{ki}^{lj}). If both T_{ki} and T_{lj} are of the same type, the vehicle can start its journey only after receiving/delivering the container from/to the C_k . Thus, y_{lj} is greater than y_{ki} by " t_{ki}^{lj} ," which is presented in constraint (6). If T_{ki} is an unloading task and T_{lj} is a loading one, the vehicle can start its new journey after receiving the container. The vehicle finishes its operation in T_{ki} and moves to C_l (takes t_{ki}^{lj}). Once, the vehicle reached C_l , it can start loading operations of the container (taking $TC + OC$). Accordingly, y_{lj} is greater

than y_{ki} by " $t_{ki}^{lj} + TC + OC$," which is stated in constraint (7). In loading tasks, C_k continues its operations on T_{ki} (takes $TC + OC$) after receiving the container from the vehicle. Hence, the starting time for the new journey of the vehicle equals to " $T_{ki} - TC - OC$." If T_{ki} is a loading task and T_{lj} is an unloading one, y_{lj} is greater than y_{ki} by " $t_{ki}^{lj} - TC - OC$." This is presented in constraint (8). From (6) to (8), M is a relatively large positive number. Thus, it is obvious that constraints from (6) to (8) are activated only if ϕ_{ki}^{lj} becomes one.

Constraint (9) states that the real completion time of T_{ki} should be greater than real completion time of $T_{k(i-1)}$ by at least the differences of their earliest possible completion time. Constraint (10) expresses that the real completion time of T_{ki} should be greater than its earliest possible completion time. Finally, (11) declares that ϕ_{ki}^{lj} is a binary decision variable.

4. The Proposed Genetic Algorithm

The coordinated scheduling of vehicles and cranes in CTs have been proved as an NP-hard problem [5]. Genetic algorithms are able to search very large spaces effectively and efficiently. Thus, GAs have been suggested by researchers to optimize scheduling problems. GAs operate on a population of potential solutions, applying the principle of survival of the fittest. Due to this selection, poorly performing individuals have less chance to survive. Consequently, in every generation of the GA, a better performing set of offspring is reproduced [20].

In this paper, a GA was developed to optimize the sequence of tasks for the cranes, and a heuristic rule assigns the vehicles to the sequence of tasks. In the proposed GA, a chromosome is a feasible sequence of the cranes' tasks; each task is located in a gene. A permutation encoding system [20] is selected to represent the tasks of cranes. Figure 1 illustrates the permutation encoding for k number of cranes, in which m_k is the number of tasks for each crane. In this method, a chromosome is a combination of integer numbers in range of 1 to $N = \sum_{i=1}^K m_i$. The chromosome is feasible for the GA operators if and only if the precedence relations of the cranes' tasks are observed.

To create the initial population for the proposed GA, a random string of numbers in range of one to N is produced, in first step. Then, the tasks belonging to each crane are found and highlighted. In the third and last step, the tasks for each crane are sorted. Therefore, the precedence relations of the tasks of cranes are observed here in initial population of the proposed GA. This suggested procedure is repeated to create the required number of chromosomes as the population size. Since in the next iterations of the proposed GA, new offsprings are produced considering the precedence constraints of the tasks; all the produced solutions will be feasible.

Once the initial population is created, the vehicles are assigned to the tasks based on the "nearest available vehicle" heuristic rule. The objective of the heuristic algorithm is to decrease waiting time for both the vehicles and the cranes. Consequently, the utilization and throughput of the cranes and vehicles are improved. Next, the objective function for

C_1				C_2				C_i				C_k			
1	2	\dots	m_1	$m_1 + 1$	$m_1 + 2$	\dots	$m_1 + m_2$	\dots					$N - 2$	$N - 1$	N

FIGURE 1: Permutation encoding of QC tasks.

each chromosome is calculated; which is the same as the objective function of the proposed MIP model (presented in (1)).

In next step of the GA, chromosomes of the current population are sorted based on the objective function in ascending mode. Then, the proposed crossover and mutation operators reproduce a new population that hopefully contains better solutions to the problem. A number of worst chromosomes in the current population are replaced by the reproduced offspring. The proposed operators of the GA are designed to check and to ignore the chromosomes that violate the precedence constraints of the tasks of cranes.

A tournament selection scheme is applied to select the parents for crossover operator. Based on Alvarenga and Mateus [21] to select the parents, tournament selection scheme chooses three chromosomes of the current population, randomly. The fittest chromosome is selected as the first parent. The procedure is repeated to select the other parent.

The number of required parents is determined by the crossover rate. The crossover operator reproduces the offspring for new generations. Task-based crossover operator proposed by Reddy and Rao [10] never offends the precedence constraints. Figure 2 illustrates an example for the modified task-based crossover proposed for this research. In this example, there are four cranes, each one performing five tasks. The third crane has been selected for the crossover operator. For this operator, tasks of the selected crane in the first parent are copied to the same positions in the second offspring. On the other hand, the tasks of the selected crane in the second parent are copied into the same positions in the first offspring. In the next step, the remaining tasks of first parent are copied in the unfulfilled positions of the first offspring, and the remaining tasks of the second parent are copied in the unfulfilled positions of the second offspring.

In next step, the mutation operator is used to ensure that all diversities in the feasible solution space are subject to search. While the solution space of the scheduling problems is too rough, the mutation operator is very important to avoid converging to local optima. In most of the literature in the area of scheduling, usually a swap mutation operator is used, where the infeasibilities arisen from are fixed by using a repair operator. In the proposed task-based mutation operator, the swap operation is performed if it does not violate the precedence constraints. A chromosome is chosen for mutation, in which two random genes are selected for a swap operation. The random genes are sorted ascending; hence, the first selected gene is always located prior to the second one. The mutation operator is performed if and only if both genes are qualified for swap operation. The following conditions are used to qualify the selected genes.

- (i) If the first gene contains the last task of a crane or the subsequent task of the first gene is located after the second gene, it can be substituted to the second gene.

- (ii) If the second gene contains the first task of a crane or the precedent task of the second gene is located prior to the first gene, it can be substituted to the first gene.

In order to keep the fittest chromosomes in current population, for next population, the elitism operator avoids them to be mutated. By using this operator, some of the fittest chromosomes in the current population are directly copied to the next population. Elitism operator assures that GA would never lead to a worse solution. Finally, after a predefined number of iterations, the algorithm stops, and the best found solution is introduced as the near optimal solution for the coordinated scheduling of the cranes and vehicles at container terminals.

5. Evaluation of the Proposed Genetic Algorithm

In order to evaluate the performance of the proposed GA and MIP model in optimization of coordinated scheduling of cranes and vehicles at container terminals, ten small size test cases were designed. These test cases have been solved using branch and bound (B&B) method of a regular solver software.

The sample layout of the CT for the test cases consists of six cranes to load/unload containers of the containerships. The containers are stored in storage yard using six L/U stations. Travel time between any of the cranes and L/U stations followed the data presented by Lau and Zhao [6]. The operational time for the cranes is assumed the same for all the loading/unloading tasks and is equal to 20 seconds. The operational time is the average time required for loading (unloading) operations of the cranes to (from) the containership including the required reshuffling operations. Moreover, travelling time of the cranes between the containership and the P/D point is set to 10 seconds. The specifications of ten numerical test cases are presented in Table 2, including the cranes and L/U stations, the initial positions of the vehicles, type of tasks (L for loading, and U for unloading tasks), and the earliest possible completion time for the cranes tasks. For more analytical results, a medium size test case is designed, where six tasks are assigned randomly to each crane. The specifications of this test case are presented in Table 3. It is noteworthy to say that the specification of the small size and the medium size test cases has been selected randomly by the authors, following the method introduced by Lau and Zhao [6].

In the first test, the effects of crossover and mutation rates on the obtained solutions by the proposed GA are presented in Table 4. This test was executed on the medium size test case using six vehicles. The vehicles are in the L/U stations number 1 to 6, as their initial positions. Every combination of crossover and mutation rate was run for ten replications and the mean of obtained objective values is reported in Table 4.

Parent 1:	6	1	16	11	2	7	17	12	3	18	13	19	20	8	4	14	9	15	10	5
Parent 2:	11	16	1	12	17	6	7	8	13	2	9	18	3	10	14	4	19	5	20	15
Offspring 1:	11	6	1	12	16	2	7	17	13	3	18	19	20	8	14	4	9	10	5	15
Offspring 2:	16	1	17	11	6	7	8	12	2	9	13	18	3	10	4	14	19	15	5	20

FIGURE 2: An illustrative example for task-based crossover operator.

TABLE 2: Specifications of ten numerical small size test cases.

No.	Task-crane- <i>L/U</i> -vehicle	Crane no.	<i>L/U</i> station	Vehicle initial position	Task type	Earliest possible completion time (s)
1	8-2-2-4	1, 2	7, 9, 9, 7 9, 7, 9, 7	12, 11, 12, 7	<i>U, U, U, L</i> <i>U, L, U, U</i>	40, 80, 120, 150 40, 70, 100, 140
2	10-2-2-4	3, 4	8, 9, 8, 9, 9 8, 8, 9, 8, 9	7, 12, 11, 11	<i>U, U, U, U, L</i> <i>L, L, U, U, U</i>	40, 80, 120, 160, 190 30, 70, 100, 140, 180
3	10-2-2-5	4, 5	10, 11, 10, 10, 11 10, 10, 10, 11, 10	7, 12, 10, 9, 11	<i>U, U, L, L, L</i> <i>L, U, L, U, L</i>	40, 80, 110, 150, 190 30, 60, 90, 120, 150
4	12-2-2-4	2, 3	9, 9, 12, 9, 12, 9 12, 12, 9, 12, 12, 12	10, 9, 7, 8	<i>L, U, L, L, U, L</i> <i>U, L, U, U, U, L</i>	30, 50, 80, 120, 150, 180 40, 70, 100, 140, 180, 210
5	12-2-2-5	1, 2	10, 9, 10, 9, 9, 10 10, 9, 9, 9, 10, 10	11, 11, 10, 7, 10	<i>L, U, L, U, U, U</i> <i>L, U, L, L, L, U</i>	30, 60, 90, 120, 160, 200 30, 60, 90, 130, 170, 200
6	12-2-3-5	5, 6	7, 9, 11, 9, 11, 9 11, 7, 7, 11, 9, 7	12, 8, 10, 9, 10	<i>U, U, L, L, U, L</i> <i>U, L, L, U, U, U</i>	40, 80, 110, 150, 180, 210 40, 70, 110, 140, 180, 220
7	14-2-2-4	2, 3	8, 10, 8, 8, 10, 8, 10 8, 8, 8, 8, 10, 8, 10	8, 8, 7, 8	<i>L, L, U, L, L, L, U</i> <i>U, U, L, L, U, U, L</i>	30, 70, 100, 130, 170, 210, 240 40, 80, 110, 150, 180, 220, 250
8	14-2-2-5	4, 5	9, 12, 9, 12, 9, 12, 9 9, 12, 12, 12, 9, 12, 9	12, 7, 12, 8, 11	<i>L, L, L, U, L, U, U</i> <i>U, L, U, L, U, U, U</i>	30, 70, 110, 140, 170, 200, 240 40, 70, 110, 140, 170, 210, 250
9	14-2-3-5	5, 6	7, 8, 10, 10, 8, 8, 7 10, 8, 7, 10, 8, 8, 10	12, 7, 8, 7, 11	<i>L, L, U, U, U, L, U</i> <i>U, U, U, L, L, U, L</i>	30, 70, 100, 140, 180, 210, 240 40, 80, 120, 150, 180, 210, 240
10	16-2-2-4	3, 4	7, 9, 7, 9, 7, 9, 7, 7 9, 7, 7, 9, 7, 9, 7, 7	9, 7, 10, 7	<i>U, U, L, U, U, L, L, U</i> <i>U, U, U, U, U, U, L, L</i>	40, 80, 110, 140, 180, 210, 250, 280 40, 80, 120, 160, 200, 240, 270, 310

TABLE 3: Specifications of the medium size test case (task type, *L/U* station, and earliest completion time).

QCs	Tasks					
	1	2	3	4	5	6
1	<i>L, 9, 30</i>	<i>U, 10, 60</i>	<i>L, 8, 90</i>	<i>L, 12, 130</i>	<i>U, 11, 160</i>	<i>L, 8, 190</i>
2	<i>L, 7, 30</i>	<i>L, 8, 70</i>	<i>U, 10, 100</i>	<i>L, 11, 130</i>	<i>L, 7, 170</i>	<i>U, 12, 200</i>
3	<i>U, 12, 40</i>	<i>U, 11, 80</i>	<i>L, 10, 110</i>	<i>L, 10, 150</i>	<i>U, 9, 180</i>	<i>L, 8, 210</i>
4	<i>U, 10, 40</i>	<i>U, 7, 80</i>	<i>U, 8, 120</i>	<i>L, 12, 150</i>	<i>L, 11, 190</i>	<i>U, 9, 220</i>
5	<i>L, 11, 30</i>	<i>L, 7, 70</i>	<i>U, 8, 100</i>	<i>U, 12, 140</i>	<i>U, 11, 180</i>	<i>U, 10, 220</i>
6	<i>L, 12, 30</i>	<i>L, 11, 70</i>	<i>U, 9, 100</i>	<i>U, 7, 140</i>	<i>U, 8, 180</i>	<i>L, 10, 210</i>

The results show that if both the crossover and mutation rates are equal to 0.5, the best results are obtained. However, the best solution for the medium size test case is obtained by the crossover and mutation rates of 0.6. The best objective value for this test case is equal to 3830. Based on this test, the best scheduling string for this test case is presented in Table 5, in which the assigned vehicle to the tasks is indicated. Moreover, a schematic view of the best scheduling found by the GA is shown in Figure 3. This schematic view shows the first 20 tasks scheduled by using the proposed GA.

The second test was conducted to compare the solutions found by the proposed GA and the MIP model. For the GA

TABLE 4: Mean of obtained solutions under various crossover and mutation rates in medium size test case.

Crossover rates	Mutation rates				
	0.5	0.6	0.7	0.8	0.9
0.5	3984.4	4104.8	4104.2	4206	4158
0.6	4032.6	4006	4046	4018.6	4140
0.7	4000.2	4082.8	4118.6	4061.8	4179.8

method, the crossover and mutation rates were set to 0.5 and 0.5, respectively. Moreover, the population size of the GA and its number of iterations was set to 40 and 1000, respectively. Preliminary tests on the small size test cases indicated that while decreasing the number of iterations would cause worse solutions for the GA method, increasing the number of iterations more than 1000 does not lead to better solutions in these test cases.

Table 6 presents the comparative results for the small size test cases. The number of decision variables, the optimal value, and CPU time required for the solver software are reported for the MIP model. In other columns, the mean and the best solutions found by the GA method are presented, where every test case was solved for 10 replications. Moreover, the CPU time for GA and the deviation between the best

TABLE 5: The best scheduling string, based on the proposed GA method in medium size test case.

		Scheduling string																	
Task no.	T_{21}	T_{31}	T_{22}	T_{32}	T_{33}	T_{51}	T_{23}	T_{41}	T_{34}	T_{42}	T_{43}	T_{44}	T_{52}	T_{24}	T_{45}	T_{35}	T_{53}	T_{25}	
Vehicle no.	V_1	V_1	V_2	V_2	V_4	V_5	V_3	V_4	V_6	V_6	V_5	V_1	V_3	V_2	V_4	V_2	V_3	V_6	
Task no.	T_{11}	T_{26}	T_{46}	T_{36}	T_{54}	T_{55}	T_{61}	T_{56}	T_{12}	T_{62}	T_{13}	T_{63}	T_{14}	T_{15}	T_{16}	T_{64}	T_{65}	T_{66}	
Vehicle no.	V_2	V_6	V_1	V_5	V_4	V_5	V_6	V_2	V_3	V_5	V_1	V_1	V_4	V_4	V_2	V_6	V_5	V_3	

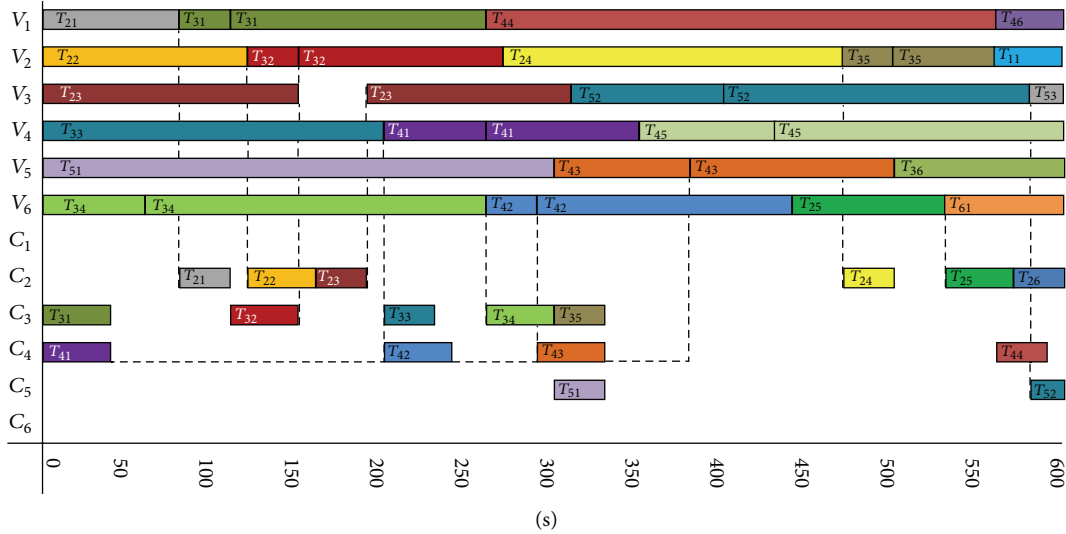


FIGURE 3: A schematic view for the best scheduling of the medium size test case found by the proposed GA.

solution of the GA method and the optimal solution of the MIP model are shown in Table 6. The deviation is defined in the following:

$$\% \text{Deviation} = \frac{\text{Best} - \text{Optimal}}{\text{Optimal}} \times 100. \quad (12)$$

Since the number of integer decision variables for the proposed MIP model increases exponentially by increasing the number of tasks, it is too time consuming to solve and find the optimal solutions for the real applications of the coordinated scheduling problem. In some cases, even the solver software cannot obtain the global optimal value in a reasonable time. Therefore, it was decided to record the best obtained result in a limited CPU time for this test case, and it is marked with an asterisk sign in Table 6.

In test cases numbers 1 and 2, the GA method shows a significant performance, while it found the best solution for each test case in all 10 replications. In other test cases, the mean of 10 replications of the GA method and its best results are near to each other (the mean objective value is 0.75% greater than the best objective value, on average). Therefore, the proposed GA method is able to find a near optimal solution with high precision. On the other hand, in all the small size test cases, the GA found a near optimal solution close to the optimal solution found by the MIP model which shows the accuracy of the proposed GA method (2.17% greater than the global optima). Overall, it is demonstrated that the GA can be used for real applications, and its results are near to the global optima of the problem. Moreover, the

results of 10 test cases illustrate that the required CPU time for the proposed GA method is reasonable for real application.

The results shown in Table 6 illustrate that assigning a predetermined number of vehicles for a specific number of tasks would not guarantee to decrease the objective function. In other words, in addition to the number of tasks per vehicle, the sequence of tasks and their specifications are so important to decrease the objective function. To demonstrate this, look at results of test case 2 and compare it with results of test case 3; the same occasion occurs for test cases 7 and 8. Therefore, for any specific application of the proposed scheduling method, one needs to determine the number of vehicles based on the results of the GA method. Nonetheless, in a specific sequence of tasks, increasing the number of vehicles would result in decreasing the objective function.

The sensitivity of the problem is analyzed with various numbers of working vehicles in medium size test case. The GA is run for five replications under crossover and mutation rates of 0.5. Figure 4 illustrates the mean and the best solutions for this test case under various numbers of vehicles. The results illustrate that employing more numbers of vehicles in a specific scheduling horizon decreases the objective function obtained by the proposed GA. However, it is obvious that in a certain number of vehicles, for example, 9 in this case, there is no significant reduction in the objective value of the coordinated scheduling problem. A cost-benefit analysis is required in such cases to determine the optimal number of vehicles in a specific scheduling horizon. On the other hand, it is obvious that as the number of vehicles increases the mean

TABLE 6: Comparative results for ten small size test cases solved by the MIP model and the GA method.

No.	MIP model			Proposed GA			Deviation %
	Decision variables	Optimal	CPU time	Mean	Best	CPU time	
1	132	471	180 S	471	471	7.4 S	0
2	176	775	1440 S	775	775	9.6 S	0
3	205	806	4200 S	818	818	9.8 S	1.49
4	226	1055	5100 S	1065	1061	12.3 S	0.57
5	259	675	7500 S	689.2	687	12.4 S	1.77
6	259	923	9000 S	1033.5	1016	12.6 S	10.07
7	282	872	11880 S	921.3	916	14.8 S	5.04
8	319	1064	12300 S	1110.5	1089	16.8 S	2.35
9	319	1154	15120 S	1186.2	1162	15.3 S	0.69
10*	344	1137	27000 S	1165.3	1159	20.3 S	1.93

*The best obtained result in a limited CPU time.

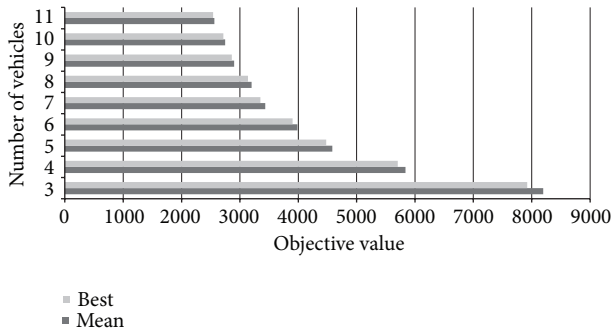


FIGURE 4: Mean and the best objective values for various numbers of vehicles in medium size test case obtained by the proposed GA method.

of replications converge to the best solutions, which means that GA has more chances to find the best solution in each replication.

6. Conclusions

The optimization of coordinated scheduling of cranes and vehicles at container terminals was presented in this paper. Considering scheduling constraints related to both vehicles and cranes improve the performance of container terminals. The problem was formulated as an MIP model solved using B&B method of regular solver software. To decrease the potential empty travels of vehicles, in this paper, a mixture of loading and unloading tasks of vessels are performed in the scheduling horizon. Moreover, the exact finishing time of the tasks of cranes and vehicle travels is calculated in this model. However the mathematical model is able to find optimum of small size cases, the computational time to find the optimal is too high to be implemented in real applications. Therefore, a genetic algorithm was proposed to find near optimal solutions for the problem in a reasonable computational time. The GA modifies the rules in a way that all of its generated chromosomes are feasible according to the precedence relations required for tasks of cranes.

The algorithm searches for tasks of cranes to find a near optimal sequence of tasks. Next, the nearest available vehicle

to the starting point of the tasks is assigned to perform the tasks. The objective function of the problem is to minimize total travelling time of the vehicles and delays in tasks of cranes. The performance of the GA and the MIP model was compared in ten small size test cases. The results show that GA can find near optimal solutions, in relatively low computational time. A medium size test case has been solved by using the GA, showing that the GA can be applied in real world instances. On the other hand, more tests on medium size test case shows that increasing the number of vehicles in a specific case decrease the objective function value. Nevertheless, reduction in objective function is not significant when the number of vehicles exceeds a specific level. Therefore, more analysis is required to decide about the number of vehicles according to their operational costs, in practice.

For further researches, it is highly recommended to include uncertainties in travelling time of the vehicles, their failure, and times for their batteries to be recharged in the coordinated scheduling method. Moreover, application of other metaheuristic optimization methods in the coordinated scheduling problem is an interesting field for future researches.

References

- [1] A. Imai, E. Nishimura, and S. Papadimitriou, "Marine container terminal configurations for efficient handling of mega-containerships," *Transportation Research Part E*, vol. 49, no. 1, pp. 141–158, 2013.
- [2] Europe Container Terminal, "ECT delta terminal," 2013, <http://www.ect.nl/en/content/ect-delta-terminal>.
- [3] K. H. Kim and Y.-M. Park, "A crane scheduling method for port container terminals," *European Journal of Operational Research*, vol. 156, no. 3, pp. 752–768, 2004.
- [4] R. Stahlbock and S. Voß, "Operations research at container terminals: a literature update," *OR Spectrum*, vol. 30, no. 1, pp. 1–52, 2008.
- [5] P. J. M. Meersmans and A. P. M. Wagelmans, "Effective algorithms for integrated scheduling of handling equipment at automated container terminals," Tech. Rep., Erasmus Research

- Institute of Management (ERIM), Erasmus University, Rotterdam, The Netherlands, 2001.
- [6] H. Y. K. Lau and Y. Zhao, "Integrated scheduling of handling equipment at automated container terminals," *International Journal of Production Economics*, vol. 112, no. 2, pp. 665–682, 2008.
 - [7] D.-H. Lee, J. X. Cao, and Q. X. Shi, "Integrated Quay crane and yard truck schedule for inbound containers," in *Proceedings of the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM '08)*, pp. 1219–1223, December 2008.
 - [8] J. Liu, Y.-W. Wan, and L. Wang, "Quay crane scheduling at container terminals to minimize the maximum relative tardiness of vessel departures," *Naval Research Logistics*, vol. 53, no. 1, pp. 60–74, 2006.
 - [9] L. Qiu, W.-J. Hsu, S.-Y. Huang, and H. Wang, "Scheduling and routing algorithms for AGVs: a survey," *International Journal of Production Research*, vol. 40, no. 3, pp. 745–760, 2002.
 - [10] B. S. P. Reddy and C. S. P. Rao, "A hybrid multi-objective GA for simultaneous scheduling of machines and AGVs in FMS," *International Journal of Advanced Manufacturing Technology*, vol. 31, no. 5-6, pp. 602–613, 2006.
 - [11] L. Chen, N. Bostel, P. Dejax, J. Cai, and L. Xi, "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, pp. 40–58, 2007.
 - [12] Q. Zeng and Z. Yang, "Integrating simulation and optimization to schedule loading operations in container terminals," *Computers and Operations Research*, vol. 36, no. 6, pp. 1935–1944, 2009.
 - [13] M. Grunow, H.-O. Günther, and M. Lehmann, "Strategies for dispatching AGVs at automated seaport container terminals," in *Container Terminals and Cargo Systems*, K. H. Kim and H.-O. Günther, Eds., pp. 155–178, Springer, Berlin, Germany, 2007.
 - [14] V. D. Nguyen and K. H. Kim, "A dispatching method for automated lifting vehicles in automated port container terminals," *Computers and Industrial Engineering*, vol. 56, no. 3, pp. 1002–1020, 2009.
 - [15] J. Cao, Q. Shi, and D.-H. Lee, "Integrated quay crane and yard truck schedule problem in container terminals," *Tsinghua Science and Technology*, vol. 15, no. 4, pp. 467–474, 2010.
 - [16] J. Zhao and L. Tang, "The simultaneous quay crane and truck scheduling problem in container terminals," in *Proceedings of the 4th International Conference on Intelligent Computation Technology and Automation (ICICTA '11)*, pp. 279–282, March 2011.
 - [17] S. M. Homayouni, S. H. Tang, N. Ismail, and M. K. A. Ariffin, "Using simulated annealing algorithm for optimization of quay cranes and automated guided vehicles scheduling," *International Journal of Physical Sciences*, vol. 6, no. 27, pp. 6286–6294, 2011.
 - [18] B. Skinner, S. Yuan, S. Huang et al., "Optimisation for job scheduling at automated container terminals using genetic algorithm," *Computers & Industrial Engineering*, vol. 64, no. 1, pp. 511–523, 2013.
 - [19] L. Chen, A. Langevin, and Z. Lu, "Integrated scheduling of crane handling and truck transportation in a maritime container terminal," *European Journal of Operational Research*, vol. 225, no. 1, pp. 142–152, 2013.
 - [20] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*, Springer, Berlin, Germany, 2008.
 - [21] G. B. Alvarenga and G. R. Mateus, "Hierarchical tournament selection genetic algorithm for the vehicle routing problem with time windows," in *Proceedings of the 4th International Conference on Hybrid Intelligent Systems (HIS '04)*, M. Geraldo Robson, Ed., pp. 410–415, Kitakyushu, Japan, December 2004.

