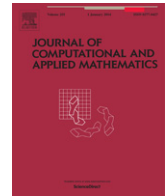




Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

A genetic algorithm for optimization of integrated scheduling of cranes, vehicles, and storage platforms at automated container terminals

Seyed Mahdi Homayouni^{a,*}, Sai Hong Tang^b, Omid Motlagh^c^a Department of Industrial Engineering, Lenjan Branch, Islamic Azad University, Isfahan, Iran^b Mechanical and Manufacturing Engineering Department, Universiti Putra Malaysia (UPM), Selangor, Malaysia^c Faculty of Manufacturing Engineering, Universiti Teknikal Malaysia Melaka (UTeM), Melaka, Malaysia

HIGHLIGHTS

- A new genetic algorithm for the integrated scheduling of handling and storage equipment was developed.
- The control parameters of the GA were examined.
- The proposed GA was compared against a previously developed algorithm.
- Integrated and non-integrated scheduling methods were examined in large scale test cases.
- Three various dwell point policies for the storage platforms were examined.

ARTICLE INFO

Article history:

Received 18 July 2013

Received in revised form 22 November 2013

Keywords:

Integrated scheduling

Genetic algorithm

Automated guided vehicles

Quay cranes

Split-platform automated storage/retrieval system

ABSTRACT

Commonly in container terminals, the containers are stored in yards on top of each other using yard cranes. The split-platform storage/retrieval system (SP-AS/RS) has been invented to store containers more efficiently and to access them more quickly. The integrated scheduling of quay cranes, automated guided vehicles and handling platforms in SP-AS/RS has been formulated and solved using the simulated annealing algorithm in previous literatures. This paper presents a genetic algorithm (GA) to solve this problem more accurately and precisely. The GA includes a new operator to make a random string of tasks observing the precedence relations between the tasks. For evaluating the performance of the GA, 10 small size test cases were solved by using the proposed GA and the results were compared to those from the literature. Results show that the proposed GA is able to find fairly near optimal solutions similar to the existing simulated annealing algorithm. Moreover, it is shown that the proposed GA outperforms the existing algorithm when the number of tasks in the scheduling horizon increases (e.g. 30 to 100).

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

As the world trade market resolves more barriers every day, containers become more important to transport goods from countries and continents to anywhere in the world. Many researchers believe that all over the world, container terminals are

Abbreviations: ACT, Automated container terminal; AGV, Automated guided vehicle; ASC, Automated stacking crane; EAV, Earliest available vehicle; FCFS, First come first service; GA, Genetic algorithm; IS, Integrated scheduling; L/U, Loading/unloading; NIS, Non-integrated scheduling; QC, Quay crane; SAA, Simulated annealing algorithm; SP-AS/RS, Split-platform automated storage/retrieval system.

* Corresponding author. Tel.: +98 9133259168.

E-mail addresses: homayouni@iauln.ac.ir (S.M. Homayouni), saihong@eng.upm.edu.my (S.H. Tang), omid@utem.edu.my (O. Motlagh).

0377-0427/\$ – see front matter © 2013 Elsevier B.V. All rights reserved.

<http://dx.doi.org/10.1016/j.cam.2013.11.021>

Please cite this article in press as: S.M. Homayouni, et al., A genetic algorithm for optimization of integrated scheduling of cranes, vehicles, and storage platforms at automated container terminals, Journal of Computational and Applied Mathematics (2013), <http://dx.doi.org/10.1016/j.cam.2013.11.021>

facing steadily increasing pressure for an upsurge in their transportation capacities and demands for a quick turnaround of vessels [1–3]. Automation of container handling equipment is a response to this ever-increasing container traffic. Automated container terminals (ACTs) have been introduced to reduce operational costs, especially the labor costs, and to increase throughput of the container terminals. In recent years, modern layouts of the container terminals are considered as a solution to increase their performance. For example, Imai, Nishimura and Papadimitriou [4] have developed two new layout designs for container terminals to serve mega container ships; and Zheng, Lu and Sun [5] have developed a heuristic algorithm for yard allocation and stowage plans in addition, which is especially designed for automated container handling systems invented by ZMPC®, in which the vehicles have been removed from their layouts. However, Yu, Cai and Zhao [6] stated that in addition to implementing advanced handling and storage equipment, new strategies and scheduling methods would improve the performance of the ACTs.

In ACTs, the vessels dock on the berths and a predetermined number of quay cranes (QCs or simply cranes) are assigned to the vessel. On the other side of the terminal, in storage yard, commonly the containers stack on top of each other using the automated stacking cranes (ASCs), while the automated guided vehicle (AGVs or simply vehicles) connect the berth and the storage areas. Chen, Huang, Hsu, Toh and Loh [7] proposed split-platform automated storage/retrieval system specially designed to handle containers. This storage system uses separate platforms for high-speed vertical and horizontal movements, called VPs, and HPs, respectively [8]. It has been calculated that this storage system is more efficient in using the land space compared to the conventional storage yards. Hu, Huang, Chen, Hsu, Toh, Loh and Song [9] and Vasili, Tang, Homayouni and Ismail [10] calculated the expected travel time for the SP-AS/RS using three various dwell point policies (i.e. stay in place, return to middle, and return to start policies). Vasili, Tang, Homayouni and Ismail [10] showed that stay in place dwell point policy gained a lower expected travel time compared against other policies.

One of the most recent strategies to improve the performance of the ACTs is the integrated scheduling method. Integrated scheduling of two equipment in the container terminal is a common research area in the literature. For example, Wu, Luo, Zhang and Dong [11] have proposed integrated scheduling of storage operations and vehicles; Homayouni and Tang [12] have developed a mixed integer-programming (MIP) model and a genetic algorithm for the integrated scheduling of cranes and vehicles; and Chen, Langevin and Lu [3] have developed an integrated method for the crane scheduling and yard truck operations. On the other hand, the integrated scheduling of all three equipments engaged in handling and storage operations of ACTs gained fewer considerations in the literature. Meersmans and Wagelmans [13] seem to be the first among researchers considering integrated scheduling of cranes, vehicles, and ASCs. An MIP model for the integrated scheduling of cranes, yard trucks, and yard cranes was formulated by Chen, Xi, Cai, Bostel and Dejax [14]. Furthermore, Liang, Lu and Zhou [15] proposed an MIP model for the integrated scheduling of cranes, inner trucks and yard cranes. They assumed that trucks are not the bottleneck of the scheduling processes. Therefore, the famous Johnson's rule for scheduling of two independent sets of processes was applied to this integrated scheduling problem. Zeng and Yang [16] developed a hybrid simulation–optimization method for this problem. The required simulation time would be decreased using a neural network algorithm in which objective functions of the proposed sequences are predicted, and potentially bad solutions are filtered out.

In the above-mentioned literature, it is assumed that the loading and unloading tasks for the vessels are executed separately [13–16]. However, Lau and Zhao [17] stated that to decrease the empty travelling of the vehicles, loading and unloading tasks can be performed concurrently for a vessel according to a predetermined list of tasks for the cranes. Lau and Zhao [17] have formulated the scheduling of cranes, vehicles, and ASCs minimizing delays in tasks of cranes, and travel time of ASCs, and vehicles. First, the SP-AS/RS involved in the integrated scheduling of container handling and storage equipment by Homayouni, Vasili, Kazemi and Tang [18]. They formulated the problem as an MIP model, and a simulated annealing algorithm (SAA) was developed to minimize the travel time of platforms in the SP-AS/RS, and the vehicles, in addition to the delays in loading and unloading tasks of the cranes.

The main objective of the current paper is to propose a genetic algorithm to optimize the integrated scheduling of cranes, vehicles, and the platforms of the SP-AS/RS. The proposed GA finds near optimal solutions for the problem in relatively low computational time. The control parameters for the proposed GA are well investigated, and the performance of the GA is compared against the optimal solutions found by the MIP model developed by Homayouni, Vasili, Kazemi and Tang [18] and their proposed SAA. In the last step, dwell point policies suggested by Vasili, Tang, Homayouni and Ismail [10] are examined in the proposed scenario of this paper.

The integrated scheduling of cranes, vehicles and platforms of the SP-AS/RS is described in Section 2. Section 3 is dedicated to describe a brief review on the principals of the genetic algorithms. Moreover, the proposed GA to optimize the integrated scheduling problem is stated in this section. Several tests on the parameters used in the proposed GA have been executed and their results are reported in Section 4. Moreover, near optimal solutions found by the proposed GA has been compared with those from literature, and the results are described in Section 4. Final notes and conclusion remarks in addition to the recommendations for further researches are presented in Section 5.

2. Problem definition

Vessel operations primarily consist of the unloading and loading tasks. The unloading task is a set of operations for discharging a container from the vessel and storing it in the storage yard. In unloading tasks, the crane moves from its dwell point to the vessel. Cranes may shuffle the containers in the hold, and pick up the desired container. The container is

transferred ashore to be loaded on a vehicle. Alongside, the vehicle moves from its dwell point to the crane. Once the crane delivers the container to the vehicle, it can start the next assigned task. In its second part of travel, the vehicle moves to the SP-AS/RS. The destination of the containers in unloading tasks is a storage cell in the SP-AS/RS. The location of this cell indicates which HP is dedicated to the current task. In unloading tasks, the VP travels from its dwell point to the load/unload station (L/U) to receive the container from the vehicle. After delivering the container, the vehicle is ready to perform its next assigned task. The loaded VP travels to the dedicated row and transfers the container to the HP. Concurrent to this travel, the HP moves from its dwell point to receive the container. The final journey of the container is performed by the HP, which moves to the predetermined storage cell, and places the container there. In the reverse direction, the loading task is the set of operations to discharge a container from the storage yard and load it into the vessel. It is noted that the tasks should be performed based on the precedence relations for the tasks of the cranes. For example, the second task of a crane cannot be performed before completion of its first task.

The integrated scheduling problem of the cranes, vehicles, and the platforms of the SP-AS/RS has been formulated as an MIP model by Homayouni, Vasili, Kazemi and Tang [18]. They have examined three various heuristic rules to assign the vehicles to the tasks of cranes, and concluded that earliest available vehicle (EAV) gains the best performance among three. Therefore, in this paper, the EAV heuristic rule is applied to assign the vehicles, as well. The effectiveness of the scheduling method is highly sensitive to the assumptions made for it. In agreement to Homayouni, Vasili, Kazemi and Tang [18], the following are the most important assumptions that must be kept in mind while optimizing the problem.

- The set of loading and unloading tasks, their starting point, and destination in the SP-AS/RS and the vessel are predetermined.
- The predetermined storage cells in the SP-AS/RS for unloading tasks are empty.
- The containers that are meant to be loaded in the vessel have been placed in the respective storage cells in the SP-AS/RS, in advance.
- Transportation times of loaded and empty cranes, vehicle, HPs, and VPs are the same.
- Congestions of the vehicles in the guide path are not considered.
- Transferring time between various types of equipment (e.g. cranes and vehicles) in the ACT is assumed deterministic, and it is small enough to be ignored.
- The dwell point policy for all types of handling equipment (i.e. the crane, HPs, VPs, and the vehicles) is “stay in place” where their last task finished.

3. The genetic algorithm

Optimization of the integrated scheduling of container handling and storage equipment is so important to the port authorities to ensure that they use the whole capacity of the equipment. However, the integrated scheduling is an NP-hard problem [13,17], i.e., no systematic method exists to find the optimal solution of this problem, especially for relatively large instances. The mathematical methods are not able to find optimum solutions for the large-scale cases of integrated scheduling problem in a reasonable computation time. Therefore, meta-heuristic algorithms (i.e. the genetic algorithm) are proposed to find near optimal solutions for this problem.

3.1. Principles of the GAs

The genetic algorithm is a well-known meta-heuristic algorithm, following the natural evolution processes. Haupt and Haupt [19] stated that like any other meta-heuristic algorithm, GAs do not guarantee to find the optimal solution. The GAs commence by defining optimization variables, objective functions, and control parameters [20]. Usually, GAs receive works on an initial population involving the individual solutions represented by “chromosomes”, which are strings include all the genes (i.e. variables) involved in a possible solution. The chromosomes are evaluated based on the “objective function”, which is the desired objective of the problem.

In any iteration of the GA, some of the current individuals are replaced with new generated offspring. “Crossover rate” is defined as ratio of the number of offspring produced in any iteration to the population size. A pair of individuals from the current population is selected as the parents of a pair of offspring. Highly fitted individuals, relative to the whole population, have a higher chance to be selected as parents in next generation, while less fitted individuals have a correspondingly low probability of being selected [21].

One of the most widely used selection schemes is called the “biased roulette wheel scheme” in which each current string in the population has a roulette wheel slot sized in proportion to its fitness. The roulette wheel scheme can be described as follows [19,21]:

- Sum the fitness of all the population members; the result is called total fitness.
- Generate a random number, θ , between zero and total fitness.
- Return the first population member whose cumulative fitness is greater than, or equal to θ .

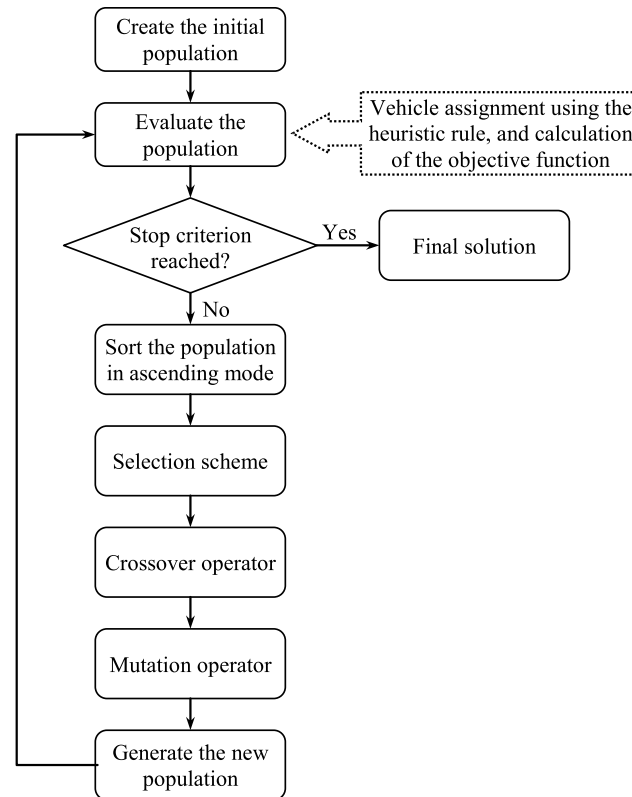


Fig. 1. The proposed genetic algorithm.
Source: Modified from [19].

The aforementioned procedure is repeated to select as many parents as required for the crossover operator. Another popular selection scheme is the “tournament” procedure, in which, two sets of the individuals with n members are selected randomly. The best individual of each set is selected as a parent. This procedure is repeated for any number of parents required for the crossover operator [20].

Mutation is another genetic operator that is applied to offspring and parents. The mutation operator is implemented to ensure that every subspace of the problem solution space is subjected for selection [21]. The “mutation rate” is defined as the percentage of the total number of genes (parents and offspring together with) in the population. In GAs, mutation operator has a vital role, firstly, to replace the genes lost from the population during the selection process; and secondly, to provide the genes that were not present in the initial population. Frequently, a swap mutation operator has been used for the GA application in scheduling literature (e.g. [1,17]). However, problems arising are fixed by using a repair operator. “Crossover” and “mutation” operators are implemented widely to reproduce new offspring, which are in charge of exploration and exploitation of the feasible solution space, respectively. It is stated that exploration and exploitation abilities of GAs are the most essential tools to find better solutions for the NP-hard problems [22].

The proposed GA in this paper finds the order of tasks for the cranes and operations in the SP-AS/RS. However, as mentioned earlier, the vehicles are assigned to the tasks by using EAV heuristic rule. The objective function for the proposed GA was proposed by Homayouni, Vasili, Kazemi and Tang [18] to minimize summation of travel times of the platforms of the SP-AS/RS, and vehicle and delays in loading and unloading tasks of the cranes. Fig. 1 illustrates various operators of the proposed GA. Details of these operators are described in following sections. The number of iterations is selected as the stop criterion for the GA.

3.2. The chromosomes and initial population

The chromosome of the proposed GA is a feasible string of tasks executed by the cranes. Tasks are sorted based on the crane number (i.e. one to K) and then the task number for each crane (i.e. one to Q_k). Next, the tasks are numbered by integers from one to N (calculated by Eq. (1)), in which T_{ki} is the i th task of QC_k , and O_{mn} is the n th operation of the m th rack of the SP-AS/RS. Fig. 2 illustrates an example of this encoding system. A chromosome in this encoding system is a feasible sequence of tasks of the cranes. In this method, the tasks are numbered sequentially, and the same integer number represents the respective SP-AS/RS operations. With reference to Fig. 2, for example integer number “1” represents both T_{11} and O_{51} in the

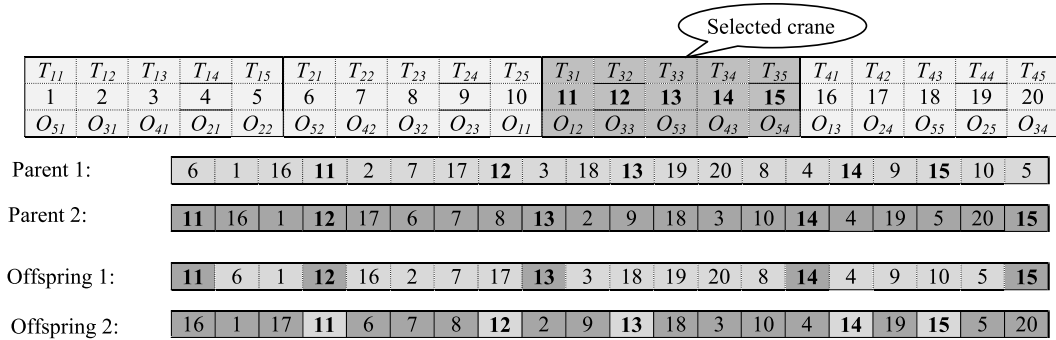


Fig. 2. An illustrative example for the proposed crossover operator.

solution chromosome.

$$N = \sum_{k=1}^K \sum_{i=1}^{Q_k} T_{ki} = \sum_{m=1}^M \sum_{n=1}^{N_m} O_{mn}. \quad (1)$$

The most important constraint of the integrated scheduling of cranes, vehicles, and storage platforms is the precedence relations of the tasks of cranes. Therefore, a solution is feasible if it observes this constraint. Referring to Fig. 2, as an example, there are four cranes, each one performing five tasks, and five storage racks, each one performing four set of operations. In this sample chromosome (e.g. parent 1), the integers are sorted based on the precedence relations of tasks of cranes (e.g., task 1 is before task 2, task 14 should be after task 13 and before task 15, and so on).

To create the initial population for the GA, a random string of numbers ranging from one to N is produced. This string might not be feasible, because of violence of the precedence relations of tasks. A repair algorithm is proposed to sort the tasks and make the string a feasible one. In the first step, all the cells in the string search for the tasks of any of the cranes. The value of the cells is set to the integer numbers that represents the first task of the respective crane. For example, all the tasks of the first crane are set to 1. In the next step, these replaced initial numbers are found from left to right, and add 1 to the second number, 2 to the third number, and so on, and finally add $Q_k - 1$ to the last one. The outcome of such an algorithm is a random feasible random string of the loading/unloading tasks, which observed the precedence relations of the cranes' tasks. The proposed algorithm to create feasible chromosomes is presented in pseudo codes in Fig. 3. The algorithm is repeated to create as many chromosomes as the number of initial population.

3.3. Selection scheme, crossover, and mutation operators

Once all the chromosomes have been evaluated by using the objective function, the current population is sorted in ascending mode. Based on the crossover rate, some of the least fitted solutions are replaced by a new set of offspring. To breed any pair of offspring, a pair of chromosomes from the current population is selected as the parents. The roulette wheel or the tournament schemes are applied in the proposed GA. The best selection scheme is designated based on its performance prior to any further analysis for the proposed GA.

The proposed crossover operator is designed to mate the parents and breed the offspring observing precedence relations of tasks of cranes. In the crossover operator, a crane is selected randomly. Then the tasks belonging to this crane are highlighted in the parent chromosomes. The tasks of this crane in parent 1 are copied to the matching positions of offspring 2, and the same tasks in parent 2 are copied to the matching positions of offspring 1. The rest of the positions in offspring 1 are fulfilled by the tasks of the unselected cranes from left to right according to their order of appearance in parent 1. The same procedure creates the second offspring.

In next step of the GA, the mutation operator is used to ensure that all diversities in the search space are subject to search. While the solution space of the integrated scheduling problems is too rough, the mutation operator is important to avoid premature convergence (converging to local optima). In the proposed mutation operator of this thesis, basics of the swap mutation operator are used and modified, in which, one of the chromosomes belonging to the current population is selected randomly. In this chromosome, two genes are selected randomly. The mutation operator is performed only if swapping the selected genes does not violate the precedence relations of the tasks of cranes. For example, it is obvious that if both tasks belong to one crane, they cannot be swapped. The following conditions are used to qualify the selected genes:

- If the first selected gene is the last task of a crane, it can be substituted to the second gene.
- Otherwise, examine whether the subsequent task of first gene is located after the second gene. If yes, the task can be substituted to the second gene.
- If the second selected gene is the first task of a crane, it can be substituted to the first gene.

```

GET N as total number of tasks
GET K as total number of QCs
GET cell as a random string of integers 1 to N
SET i to 1
SET c to 1
GET  $Q_i$  as total number of tasks for  $QC_i$ 
FOR j = 1 to N
    FOR k = 1 to N
        IF cellk = j THEN
            SET cellk to c
        END IF
    END FOR
    IF j >= c +  $Q_i$  - 1 THEN
        SET c to c +  $Q_i$ 
        SET i to i + 1
    END IF
END FOR
SET j to 1
SET c to 1
FOR i = 1 to K
    FOR k = 1 to N
        IF cellk = j THEN
            SET cellk to c
            SET c to c + 1
        END IF
    END FOR
    SET j to j +  $Q_i$ 
END FOR

```

Fig. 3. Pseudo codes for creating a feasible random string of the tasks.

- Otherwise, examine whether the precedent task of first gene is located prior to the second gene. If yes, the task can be substituted to the first gene.

To avoid missing the fittest chromosome of the current population, the elitism operator keeps three of the best chromosomes and avoids them being mutated. This operator assures that the GA would never lead to a worse solution. After a predetermined number of iterations, a near optimal solution for the integrated scheduling problem is obtained.

4. Results and discussions

In this section, performance of the proposed GA is evaluated through three sets of test, including tests on control parameters of the GA, comparison of the GA and the existing methods to solve the integrated scheduling problem, and sensitivity analysis of the critical factors of the problem. As noted earlier in Section 1, Homayouni, Vasili, Kazemi and Tang [18] have developed a simulated annealing algorithm to optimize the integrated scheduling problem. They used the layout data reported by Lau and Zhao [17] to design 10 small size test cases (called S1 to S10) including eight to 18 loading/unloading tasks assigned to two to four cranes, two to six vehicles in the connecting area, and two to four L/U stations in the storage yard. In this paper, the same data was used to design 10 medium (called M1 to M10) and 10 large size (called L1 to L10) test cases. The medium size test cases include 30–60 tasks on three or four cranes, and 3–6 vehicles in the connecting area. Moreover, the large size test cases include 64–100 tasks on four or five cranes, and 4–6 vehicles. The general specifications of the test cases including number of tasks, cranes, L/U stations, and vehicles (shown as T-Q-L-A) is presented in Tables 1, 3, and 5 for the medium, small, and large size test cases, respectively. Any test case was solved using the proposed GA for 10 replications; because Lau and Zhao [17] stated that 10 replications provide 95% confidence that the solutions are 0.1% away from the mean of the solutions with a power of 80%.

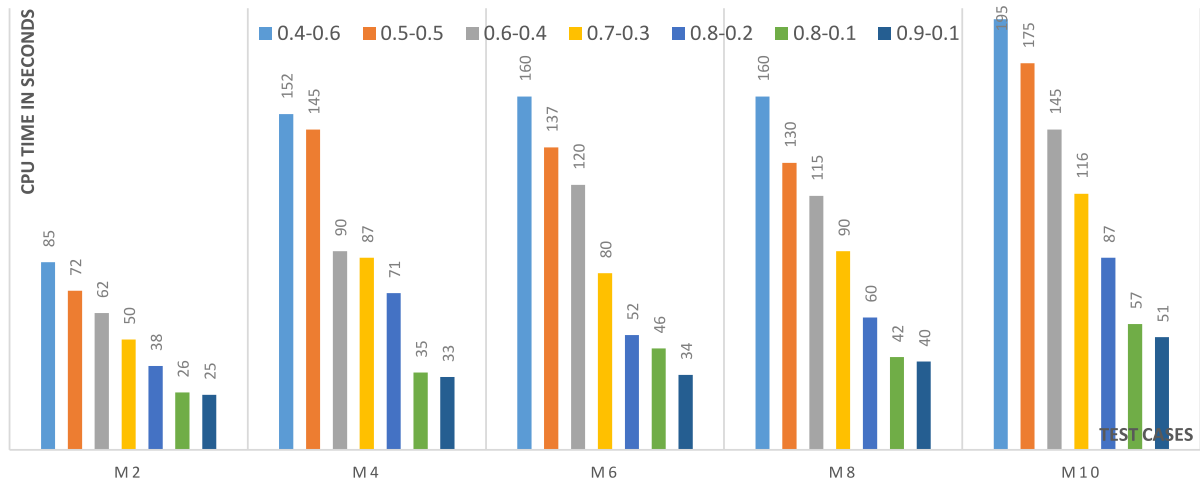
4.1. Control parameters of the proposed GA

Haupt and Haupt [19] believed that deciding on the optimum population size becomes more important and more controversial in NP-hard problems. To determine the best combination of number of iterations, as the stop criteria of the

Table 1

Performance of selection schemes—tournament against roulette wheel.

Method	Test cases										Average
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	
T-Q-L-A	30-3-4-6	30-3-3-7	40-4-5-5	42-3-5-6	48-3-3-7	48-4-5-6	52-4-6-7	52-4-5-8	60-4-6-7	60-4-6-10	
RW Mean	2966.3	2920.6	4149.1	4454.8	3397.9	3969.2	4633.1	5059.9	4931.6	4886.1	
Best	2945	2891	4009	4257	3336	3842	4423	4888	4729	4824	
T Mean	2945	2860.4	3972.5	4250.2	3301.2	3890.5	4468.3	4967.6	4701.2	4814.8	
Best	2879	2820	3766	4072	3258	3717	4426	4697	4489	4750	
MeanRW MeanT	= 1.007	1.021	1.045	1.048	1.029	1.021	1.037	1.019	1.049	1.015	1.029

**Fig. 4.** CPU time to solve selected medium size test cases under various combinations of crossover and mutation rates.

proposed GA, and population size, a preliminary test has been conducted on the set of medium size test cases. Eight different combinations of population size and number of iterations (100*250–50*2000) have been tested. The results demonstrated that 50 as the population size and 2000 as the number of iterations shows the best performance among other combinations. Sivanandam and Deepa [21] discussed that larger population sizes make it easier for the GA to explore the search space. However, very large population sizes (like 100 in this research) increase the computation time for the GA and would not be recommended. The results of this test illustrated that neither large population sizes (e.g. 100) nor small ones (e.g. 20) are recommended for the proposed GA. Kozan and Preston [23] and Hartmann [24] have conducted similar tests for their scheduling problems in container terminals. Their conclusions support findings of the current research, in which both small and large population sizes were rejected for this scheduling problem in the CTs.

To choose the selection scheme for this research, all the medium size test cases have been optimized by using the tournament “T” and roulette wheel “RW” selection schemes. Results in Table 1 shows a ratio of the mean objective values found under the roulette wheel selection scheme to those found under the tournament selection scheme. It is obvious that on average, the tournament selection scheme find better solutions for the large test cases. Consequently, the tournament has been selected as the selection scheme of this research. Lee, Soak, Kim, Park and Jeon [25] have examined the similar procedure, and concluded that tournament selection schemes provide more diversity in the population than the roulette wheel. It seems that a comparatively high selection pressure and high sampling accuracy which is done by the tournament selection scheme, can provide good performance in the selection strategy [25].

The third test has been conducted to select the best control parameters for the proposed GA. The selected medium size test cases have been optimized using seven combinations of crossover and mutation rates. The results presented in Table 2 indicate that the combination of 0.4 and 0.6 for crossover and mutation rates have the weakest performance among all the tested combinations. This can be interpreted as producing more offspring in each generation is more efficient at achieving better solutions than exploiting new search areas. The results of this test support the findings of the researches conducted by Lau and Zhao [17] and He, Chang, Mi and Yan [26] who used similar crossover and mutation rates for their proposed GA to optimize integrated scheduling. They explained that a higher crossover rate compared to the mutation rate is more suitable to explore and exploit the feasible solution space of the integrated scheduling problem. Fig. 4, illustrates the CPU time in seconds to find a near optimal solution for the medium size test cases, under various combinations of crossover and mutation rates. As shown in this figure, the higher the mutation rate, the greater the CPU time. This is due to the difficult procedure of finding two tasks that are eligible for the mutation process. Overall, with reference to the results of Table 2 and Fig. 4, the combination of 0.7 and 0.3 has been selected as the most proper crossover and mutation rates, respectively.

Table 2

Objective values obtained by the GA under various combinations of crossover and mutation rates.

No.		Crossover rate–mutation rate						
		0.4–0.6	0.5–0.5	0.6–0.4	0.7–0.3	0.8–0.2	0.8–0.1	0.9–0.1
M2	Mean	2872.9	2860.4	2851.1	2852.4	2819.2*	2828.4	2828.8
	Best	2835	2820	2818	2808	2793	2791	2801
M4	Mean	4372.3	4250.09	4244.8	4241.1*	4269.4	4309.9	4287.1
	Best	4232	4072.5	4076	4038	4126	4124	4131
M6	Mean	3924.5	3890.5	3872.5	3847.9*	3897.6	3953.7	4014.5
	Best	3721	3717	3679	3635	3644	3753	3858
M8	Mean	4973.2	4967.6	4935.8	4923.3*	4932.5	5030	4953.7
	Best	4679	4697	4639	4549	4836	4711	4641
M10	Mean	4756.8	4814.8	4816.2	4771.6	4735.4	4704.2*	4786.4
	Best	4683	4750	4656	4634	4595	4608	4563

Table 3

Comparison of the performance of MIP model, GA, and SA algorithm in small size test cases.

No.	T-Q-L-A	MIP model			SA algorithm				Genetic algorithm			
		Dec. Var.	Optimal	CPU time (s)	Best	Mean	CPU time (s)	Opt. gap	Best	Mean	CPU time (s)	Opt. gap
S1	8-2-2-3	371/294	873.3	22	890.3	938.2	7	1.9	983.3	994.54	8	12.6
S2	9-3-3-2	463/344	951.2	202	1024.4	1024.4	6	7.7	1053.1	1075.3	6	10.7
S3	10-2-2-5	520/418	729.5	26	735.5	742.4	9	0.8	836.2	856.2	9	14.6
S4	12-2-2-6	672/576	948	185	954.2	990.8	10	0.6	1078.3	1091.8	14	13.7
S5	12-3-3-4	588/516	1296.4	1 143	1303.8	1305.8	8	0.6	1342.9	1360.8	10	3.6
S6	12-3-4-4	582/518	1198.9	499	1254	1278.8	7	4.6	1288.1	1298.3	10	7.4
S7	12-4-2-3	582/518	2568.3*	32 000	2688.2	2774.5	10	4.7	2739.3	2756.4	8	6.7
S8	15-3-3-5	826/692	1175.6	3 103	1192.8	1217.4	10	1.5	1267.8	1267.8	13	7.8
S9	16-4-4-4	904/747	2388.5*	32 000	2400.7	2501.3	10	5.1	2485.6	2520.1	12	4.1
S10	18-3-3-6	1071/741	1624.4*	32 000	1721	1788.4	12	5.9	1767.2	1834.1	16	8.7

* indicates that the optimal solution for the problem was not obtained in more than 9 hours of computation.

4.2. Performance evaluation of the proposed GA

The optimal solutions of the small size test cases suggested by Homayouni, Vasili, Kazemi and Tang [18] are presented in Table 3. According to Homayouni, Vasili, Kazemi and Tang [18] “Dec. Var” shows number of decision variables/number of integer decision variables for each test case. The (near)optimal solutions of the cases and the CPU time are reported for the MIP model and the SAA proposed by Homayouni, Vasili, Kazemi and Tang [18]. These small size test cases have been solved using the proposed GA. The best and the mean objective values found by the proposed GA, and its CPU time in seconds for the test cases are shown in Table 3. It is noted that in cases S7, S9, and S10 the MIP model was not able to find the optimal solution in reasonable CPU time (i.e. 9 h for this test). Therefore, the best available solutions in the limited computational time were recorded [18]. The asterisk sign marks the very last feasible solution found by the MIP model. The optimality gap (2), shown by “Opt. Gap” in Table 3, is the differences between the best solution found by the GA (the SAA) and the optimal solution found by the MIP model, expressed as a percentage.

$$\text{Optimality Gap\%} = \frac{\text{Best} - \text{Optimal}}{\text{Optimal}} \times 100. \quad (2)$$

Looking at the results of the meta-heuristic algorithms indicates that the GA is not able to find near optimal solutions in test cases S1 to S4. This would be due to the operators used to generate new solutions in this algorithm. In test cases S1, S2, and S4, the tasks are scheduled for two cranes. Since the proposed crossover operator substitutes the position of a group of tasks belonging to the same crane, the GA is not able to explore the search space. Nevertheless, the SAA shows its superior performance in these test cases. This is due to using a swap neighbor search structure proposed for the SAA in [18], which substitutes the tasks individually. Therefore, SAA is able to search in a more widely contrasted manner than the GA. In test cases S5 to S10, the means of the optimality gaps for the GA and SAA are 6.4% and 3.7%, respectively, which are acceptable optimality gaps compared against the meta-heuristic algorithms proposed in the literature, for the scheduling problems. For example, the average optimality gap for the heuristic algorithm proposed by Nguyen and Kim [27] is 0.6%. Moreover, the algorithms proposed by Lau and Zhao [17] are almost 1.4% and 5.1% worse than the optimal solutions. The results of this test indicate that if the number of cranes increases the GA would be able to search the feasible solution space of the problem more efficiently. Moreover, on average, the mean of the objective values for the GA are 1.7% greater than the best-obtained objective value for each test case, meaning that it can find solutions close to each other in its various runs. In conclusion, the proposed GA, for the scale of small size test cases is effective and precise.

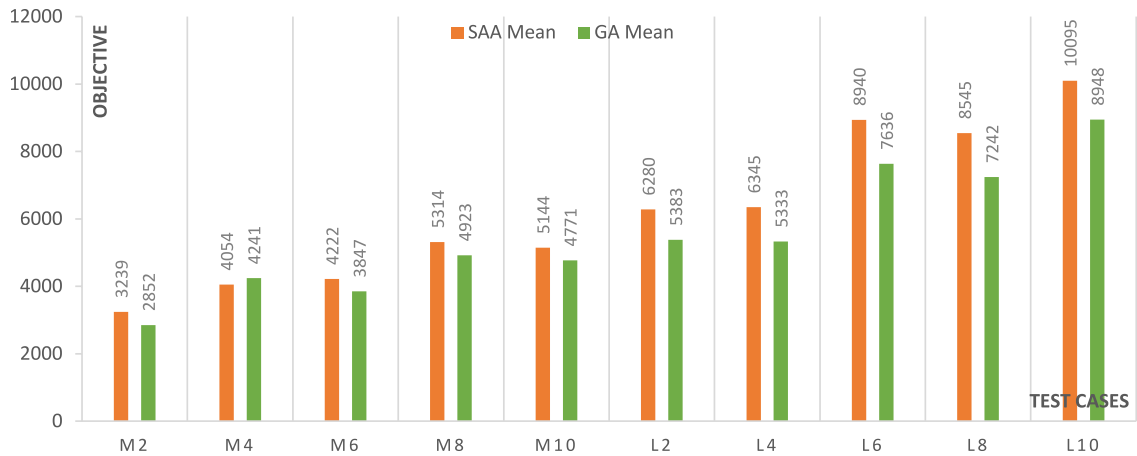


Fig. 5. Comparison of the GA and SAA in the medium and large size test cases.

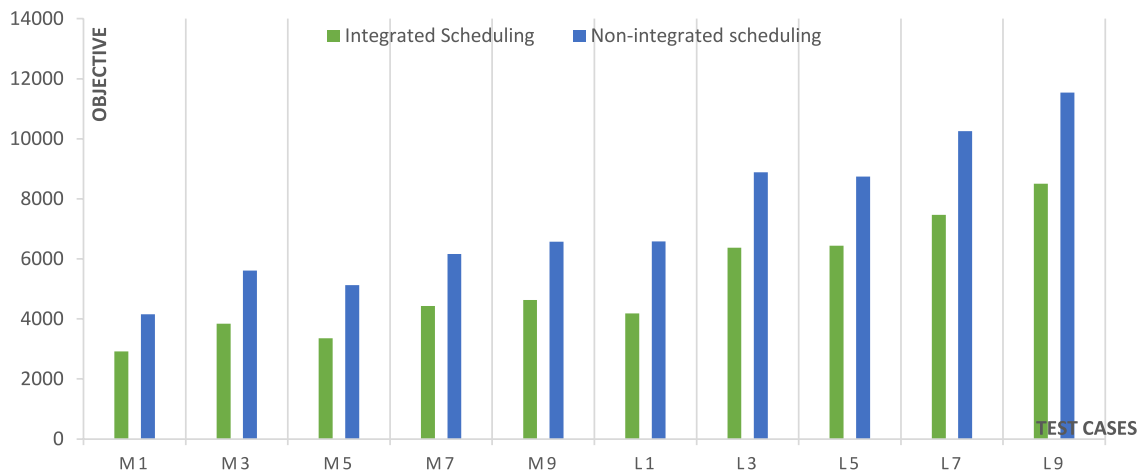


Fig. 6. Non-integrated against integrated scheduling method in medium and large test cases.

According to the results obtained in the previous test, the proposed GA would be able to find better solutions in large test cases compared against the SAA. Therefore, the medium and large size test cases have been solved by using the proposed GA, and the results are compared to those driven by the SAA. Fig. 5 illustrates the mean objective value of each test case obtained using the GA and the SAA. According to the results of Fig. 5, the near optimal solutions found by the GA are on average 13.5% better than those found by the SAA. These results are only valid in the scale of the medium and large size test cases. As a rule of thumb the GA shows superior performance when the number of cranes are three and more, and the number of tasks per crane is ten and more. In these cases, a relatively large number of tasks allows the crossover and mutation operators to explore and exploit the solution space more efficiently.

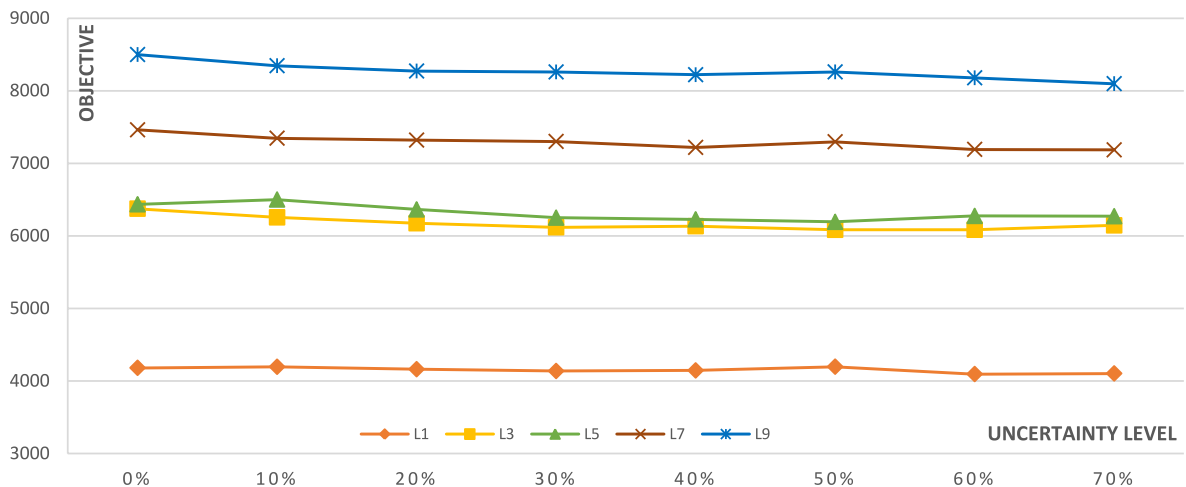
In the third test of this section, a selected number of medium and large size test cases have been solved using both the integrated scheduling (IS) method by using the proposed GA and the non-integrated scheduling (NIS) method. The mean of objective values of IS method, in addition to the objective of the NIS method are presented in Fig. 6. In both the GA and the NIS method, vehicles, HPs, and VPs are assigned to the task using the EAV heuristic rule. In the GA the sequences of tasks are changed considering the availability of the vehicles, VPs, and HPs. The non-integrated scheduling method works on the basis of a first-come-first-service (FCFS) rule for the loading/ unloading tasks, meaning that the NIS method schedules any equipment separately. Thus, the proposed integrated scheduling method achieves better schedules for the problem compared with the NIS method. Vis and Roodbergen [28] have made a comparison of their proposed scheduling method for the storage operations against the FCFS heuristic rule. The objective function they defined for this scheduling method is similar to the second and third objectives of this research. It is interesting that their proposed method, on average, achieved 46% shorter travel distances compared with the FCFS heuristic method.

The last set of tests in this section is for evaluating the effects of various number of operating vehicles on both IS and NIS methods. The results are tabulated in Table 4, in which the mean objective values for the selected large size test cases solved by the GA, and the objectives of the NIS method are presented. With reference to Table 4, one can conclude that on

Table 4

Impact of number of vehicles on integrated and non-integrated scheduling methods.

No.	Method	Number of vehicles							
		6	8	10	12	14	16	18	20
L1	IS	5926	4905	4409	4153	4019	3799	3789	3777
	NIS	8280	7031	6464	6384	6128	6149	6190	6220
L3	IS	8938	7729	6478	6148	6008	5975	6035	5930
	NIS	11252	10159	9425	9041	8557	8491	8518	8334
L5	IS	9541	7833	6801	6198	5708	5594	5497	5316
	NIS	12835	10424	8906	8093	7972	7482	7308	7132
L7	IS	10997	8831	7702.5	7000	6532	6198	5937	5924
	NIS	14789	11500	10432	10025	8982	8755	8600	8560
L9	IS	12354	10137	8648	7718	7150	6817	6444	6297
	NIS	15741	12890	11530	10458	9605	8829	8532	8049

**Fig. 7.** Effects of uncertainty in operational time of cranes on the mean objective values.

average, half of the vehicles are required under the IS method, to achieve a similar objective value to the NIS method. For example, in test case L3, the objective value of the IS method using six vehicles is similar to the objective value of the NIS method using 12 vehicles. Moreover, in test case L7, the IS method using eight vehicles achieves a similar objective value to the NIS method when implementing 16 vehicles. Such cases are bolded for both IS and NIS methods for each test case in Table 4. A similar test has been conducted by Vis and Roodbergen [28] based on the FCFS heuristic rule for the operations of the straddle carriers. They concluded that based on their optimal scheduling method almost 60% of the number of straddle carriers are required to perform the same sequence of tasks compared with the FCFS heuristic rule.

4.3. Sensitivity analysis of the integrated scheduling method

In previous tests, it was assumed that the operational time of the cranes is deterministic. However, due to different reshuffling times required to load/unload containers to/ from the vessels, in real world application, the operational time of the cranes would vary from task to task. This test was designed to investigate how uncertainty in operational time of the cranes would affect the objective function of this integrated scheduling problem. The test was performed on selected large size test cases under various uncertainty levels. Mean objective values of the GA are drawn in Fig. 7. As an example, an uncertainty level of 0.4 means that the operational time of the cranes would vary in a time interval of 20 ± 40 s (i.e. 12–28 s). The operational time was produced using a uniform random function. The results illustrate that increasing the uncertainty level results in a slight decrease in the objective value. However, on average, the largest objective value under 70% of uncertainty level is only 4% lower than that under a 0% uncertainty level. Zhou and Kang [29] stated that uncertainty in the operational time of the cranes affect the overall performance of loading/unloading tasks in CTs. However, according to the results figured out in this test, it can be inferred that using an average operational time for the cranes would not affect the solutions of the integrated scheduling problem, impressively. Because the integrated scheduling problem focuses on the overall performance of the handling and storage equipment (instead of their individual performance), using the average operational/travelling time of the equipment is merely suitable for this scheduling problem.

Table 5

Effects of various dwell point policies for the SP-AS/RS on the objective values of the integrated scheduling method.

No.	T-Q-L-A	Stay in place policy				Return to middle policy				Return to start policy			
		TT	TD	THV	Obj.	TT	TD	THV	Obj.	TT	TD	THV	Obj.
L1	64-4-4-10	10 492	3555	2890	4182	2.25	0.38	26.44	3.47	0.98	1.77	15.61	2.20
L2	68-4-4-10	14 622	4524	3316	5413	3.96	0.79	24.61	4.36	1.90	0.49	13.03	2.20
L3	72-4-5-10	16 378	5401	4167	6374	1.63	−0.74	17.42	2.07	0.87	0.07	6.94	1.08
L4	80-4-5-10	15 604	4240	3765	5328.3	1.77	0.48	18.94	2.62	1.11	0.70	9.51	1.59
L5	75-5-6-10	16 765	5488	3784	6435	1.44	0.40	21.91	2.53	0.80	−0.18	17.92	1.71
L6	80-5-6-10	17 399	6731	4112	7535	1.90	0.49	21.89	2.68	0.80	0.30	16.32	1.55
L7	85-5-6-10	18 309	6460	4623	7461	0.94	0.33	17.50	1.82	0.42	0.23	11.25	1.05
L8	90-5-6-10	17 241	6594	4077	7377	2.35	0.30	25.80	3.58	0.67	0.67	14.62	1.84
L9	95-5-6-10	19 674	7617	4386	8499	1.25	0.33	16.35	1.82	0.89	−0.22	10.21	1.13
L10	100-5-6-10	21 158	7839	4453	8832	1.10	0.45	20.17	1.91	0.23	−0.08	16.48	0.99
						1.86	0.32	21.1	2.69	0.87	0.38	13.19	1.53

Designing the operational level of the SP-AS/RS, Vasili, Tang, Homayouni and Ismail [10] investigated in three different dwell point policies for the platforms of the SP-AS/RS, and theoretically concluded that “stay in place” where the last task of the platform was finished is the best dwell point policy for the SP-AS/RS. Here, in this test, the claim has been examined in the proposed large size test cases. Thus, these test cases have been solved using the GA under three different dwell point policies. The mean of travel time of the vehicles, total delays for tasks of cranes, travel time of the horizontal and vertical platforms, and overall objective (shown as TT, TD, THV, and Obj., respectively) are tabulated in Table 5. This table presents the deviation of objectives under the return to middle and return to start dwell point policies compared to those under the stay in place policy. On average, return to middle and return to start dwell point policies result in 21% and 13% longer travel time for the platforms of the SP-AS/RS compared against the stay in place dwell point policy, which supports the facts found by Vasili, Tang, Homayouni and Ismail [10]. However, the overall objective of the problem would not increase, substantially. On average, the objective values obtained under return to middle and return to start policies are 2.7% and 1.5% greater than this obtained under the stay in place policy. As discussed in previous sections, the most important objective for this problem is to minimize the delays in tasks of cranes. With reference to Table 5, evidently, longer routes for the platforms have no effect on the performance of the other two handling equipments involved in the integrated scheduling method.

5. Conclusions

The most updated automated container terminal, Euromax® started its services in June 2008. This terminal is equipped with AGVs, ASCs, and QCs. However, they still stack the containers in the storage yards. Stacking cranes evidently results in lower land utilization. Moreover, the handling time, including shuffling time, would increase total service time. SP-AS/RS is an alternative to the stacking storage system, whose physical and economic feasibility has been proved, theoretically.

It is demonstrated that tournament selection scheme shows a better performance compared against the roulette wheel in the proposed GA. On the other hand, it is shown that the crossover operator is more important because of finding better solutions compared with the mutation operator, in this problem. That is why the higher mutation rate and lower crossover rates have been rejected for the proposed GA.

Although the integrated scheduling has been approved as a major improvement in container terminals, this problem has not been investigated well in past literature. Many researchers using the simulation analysis support the idea that automated lifting vehicles might decrease the complexity of the problem, and improve the performance of the ACTs. However, rarely are scheduling methods found for this equipment. It is recommended for future studies to include this kind of vehicle in the integrated scheduling problems. For the future researches in this field of study, it is highly recommended to develop methods for rescheduling in cases that an unpredicted phenomenon (e.g. failure in any of the equipment) avoids continuing the current schedule. Moreover, implementing a hybrid of meta-heuristic algorithms to improve the performance of optimization methods in the integrated scheduling problem, (e.g. using fuzzy genetic algorithms, a hybrid of GA and SAA, etc.) is of interest.

References

- [1] C.-J. Liang, M. Chen, M. Gen, J. Jo, A multi-objective genetic algorithm for yard crane scheduling problem with multiple work lines, *J. Intell. Manuf.* (2013) 1–12.
- [2] B. Skinner, S. Yuan, S. Huang, D. Liu, B. Cai, G. Dissanayake, H. Lau, A. Bott, D. Pagac, Optimisation for job scheduling at automated container terminals using genetic algorithm, *Comput. Ind. Eng.* 64 (2013) 511–523.
- [3] L. Chen, A. Langevin, Z. Lu, Integrated scheduling of crane handling and truck transportation in a maritime container terminal, *European J. Oper. Res.* 225 (2013) 142–152.
- [4] A. Imai, E. Nishimura, S. Papadimitriou, Marine container terminal configurations for efficient handling of mega-containerships, *Transp. Res. E-Log.* 49 (2013) 141–158.
- [5] K. Zheng, Z. Lu, X. Sun, An effective heuristic for the integrated scheduling problem of automated container handling system using twin 40' cranes, in: *Second International Conference on Computer Modeling and Simulation, ICCMS'10*, 2010, pp. 406–410.

- [6] M. Yu, Y. Cai, Z. Zhao, Research on cooperative scheduling at container terminal under uncertainties, in: Q. Zu, B. Hu, A. Elçi (Eds.), *Pervasive Computing and the Networked World*, Springer, Berlin, Heidelberg, 2013, pp. 769–774.
- [7] C. Chen, S.Y. Huang, W.J. Hsu, A.C. Toh, C.K. Loh, Platform-based AS/RS for container storage, in: *IEEE International Conference on Robotics and Automation*, ICRA'03, 2003, pp. 181–187.
- [8] M.R. Vasili, S.H. Tang, M. Vasili, Automated storage and retrieval systems: a review on travel time models and control policies, in: R. Manzini (Ed.), *Warehousing in the Global Supply Chain*, Springer-Verlag International Publisher, London, 2012, pp. 159–209.
- [9] Y.H. Hu, S.Y. Huang, C. Chen, W.-J. Hsu, A.C. Toh, C.K. Loh, T. Song, Travel time analysis of a new automated storage and retrieval system, *Comput. Oper. Res.* 32 (2005) 1515–1544.
- [10] M.R. Vasili, S.H. Tang, S.M. Homayouni, N. Ismail, A statistical model for expected cycle time of SP-AS/RS: an application of Monte Carlo simulation, *Appl. Artif. Intell.* 22 (2008) 824–840.
- [11] Y. Wu, J. Luo, D. Zhang, M. Dong, An integrated programming model for storage management and vehicle scheduling at container terminals, *Res. Transport. Econ.* 42 (2013) 13–27.
- [12] S.M. Homayouni, S.H. Tang, Multi objective optimization of coordinated scheduling of cranes and vehicles at container terminals, *Math. Probl. Eng.* 2013 (2013) 1–9.
- [13] P.J.M. Meersmans, A.P.M. Wagelmans, Effective algorithms for integrated scheduling of handling equipment at automated container terminals, in: *Technical Report of Erasmus Research Institute of Management, ERIM, Erasmus University, Rotterdam, The Netherlands*, 2001.
- [14] L. Chen, L.-F. Xi, J.-G. Cai, N. Bostel, P. Dejax, An integrated approach for modeling and solving the scheduling problem of container handling systems, *J. Zhejiang Univ. Sci. A* 7 (2006) 234–239.
- [15] L. Liang, Z.-Q. Lu, B.-H. Zhou, A heuristic algorithm for integrated scheduling problem of container handling system, in: *International Conference on Computers & Industrial Engineering, CIE 2009*, 2009, pp. 40–45.
- [16] Q. Zeng, Z. Yang, Integrating simulation and optimization to schedule loading operations in container terminals, *Comput. Oper. Res.* 36 (2009) 1935–1944.
- [17] H.Y.K. Lau, Y. Zhao, Integrated scheduling of handling equipment at automated container terminals, *Int. J. Prod. Econ.* 112 (2008) 665–682.
- [18] S.M. Homayouni, M.R. Vasili, S.M. Kazemi, S.H. Tang, Integrated scheduling of SP-AS/RS and handling equipment in automated container terminals, in: *42nd International Conference on Computers and Industrial Engineering, CIE42*, Cape Town, South Africa, 2012, pp. 1–12.
- [19] R.L. Haupt, S.E. Haupt, *Practical Genetic Algorithms*, John Wiley and Sons Inc., New Jersey, 2004.
- [20] A. Sokolov, D. Whitley, Unbiased tournament selection, in: *The Conference on Genetic and Evolutionary Computation, GECCO'05*, ACM, Washington DC, 2005, pp. 1131–1138.
- [21] S. Sivanandam, S. Deepa, *Introduction to Genetic Algorithms*, Springer-Verlag, Berlin, 2008.
- [22] M. Bazzazi, N. Safaei, N. Javadian, A genetic algorithm to solve the storage space allocation problem in a container terminal, *Comput. Ind. Eng.* 56 (2009) 44–52.
- [23] E. Kozan, P. Preston, Mathematical modelling of container transfers and storage locations at seaport terminals, in: K.H. Kim, H.-O. Günther (Eds.), *Container Terminals and Cargo Systems*, Springer-Verlag, Berlin, Heidelberg, 2007, pp. 87–105.
- [24] S. Hartmann, A general framework for scheduling equipment and manpower at container terminals, in: H.-O. Günther, K.H. Kim (Eds.), *Container Terminals and Automated Transport Systems*, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 207–230.
- [25] S. Lee, S. Soak, K. Kim, H. Park, M. Jeon, Statistical properties analysis of real world tournament selection in genetic algorithms, *Appl. Intell.* 28 (2008) 195–205.
- [26] J. He, D. Chang, W. Mi, W. Yan, A hybrid parallel genetic algorithm for yard crane scheduling, *Transp. Res. E-Log.* 46 (2010) 136–155.
- [27] V.D. Nguyen, K.H. Kim, A dispatching method for automated lifting vehicles in automated port container terminals, *Comput. Ind. Eng.* 56 (2009) 1002–1020.
- [28] I.F.A. Vis, K. Roodbergen, Scheduling of container storage and retrieval, *Oper. Res.* 57 (2009) 456.
- [29] P.-F. Zhou, H.-G. Kang, Study on berth and quay-crane allocation under stochastic environments in container terminal, *Syst. Eng. Theory Pract.* 28 (2008) 161–169.