Storage Allocation and Yard Trucks Scheduling in Container Terminals Using a Genetic Algorithm Approach

Z.X. Wang, Felix T.S. Chan, and S.H. Chung

Abstract—Storage allocation and yard trucks scheduling are two important problems that influence the efficiency of a container terminal. This paper integrates the two problems as a whole and attempts to minimize the weighted summation of total delay and total yard trucks travel time. Genetic Algorithm (GA) is proposed to deal with the problem. The crossover operation is based on the knowledge of job's ready time and due time, and a strategy of mutation which combines three methods of mutation is proposed for the GA. The optimal solution obtained by using CPLEX is used as a comparison with the performance of the proposed GA. Results of computational experiments show that the proposed GA is able to obtain a near optimal solution in reasonable time.

Keywords—Container Terminal, Genetic Algorithm, Storage Allocation, Yard Truck Scheduling

I. INTRODUCTION

With rapid development in techniques and trade globalization, marine transportations have played a very important role in logistics networks. Container terminals are essential spots to comprise the networks, and the efficiency of container terminals is the most important factor that influences marine transportations. Terminal turnaround time, which means the average time a ship stays in a terminal, is the most vital factor in determining terminal efficiency. The study of operations in container terminals to shorten the turnaround time is therefore necessary.

Container terminals have put much effort on shortening the turnaround time by developing various decision support techniques to impact terminal operations [1]. In general, these operations include berth allocation, quay crane scheduling, yard truck scheduling, yard crane scheduling, and storage allocation [2]. When vessels come into a terminal, the operator would first allocate berths and quay cranes for each vessel. Locations for the discharging containers are then allocated. Finally, yard cranes and a fleet of trucks will be dispatched to accomplish the corresponding loading and discharging operations.

This paper focuses on the yard truck scheduling and storage allocation problems. Researchers usually consider only one of these two problems. In the area of containers storage allocation, Kim and Kim [3] addressed the problem of allocating storage space for import containers. A segregation policy has also been considered, in which the author analyzed the cyclic and dynamic arrival rates. The aim of the problem was to minimize the expected number of rehandles. Zhang et al. [4] studied the storage allocation problem and a rolling-horizon approach was proposed. In the area of yard truck scheduling, Kim and Bae [5] discussed the ways to dispatch Automated Guided Vehicles (AGVs) by utilizing information on locations and times of future delivery tasks. A mixed-integer programming model was proposed and the problem was solved by a heuristic algorithm. Ng et al. [6] considered the problem of scheduling a fleet of trucks in a container terminal to minimize makespan. Sequence-dependent processing times and different ready times were considered in the loading and discharging tasks. The problem was formulated as a mixed integer program (MIP) and solved with a genetic algorithm.

Recently, some researches considered the two problems as a whole. Lee et al. [7] proposed an integer programming model to deal with the problems of yard truck scheduling and storage allocation. It can be considered as the first paper in studying the two problems as a whole instead of solving them separately. The objective is to reduce the congestion and idle time of yard trucks to reduce the makespan of discharging containers. Lee et al. [2] further extended the previous study and proposed another integrated model for the yard truck scheduling and storage allocation problems. A hybrid insertion algorithm was proposed.

In this paper, we propose a Genetic Algorithm (GA) to deal with yard truck scheduling and storage allocation problems as a whole. This paper is organized by this introductory section and following sections: Section 2 provides a mathematical model and problem description. Section 3 presents the proposed GA. Section 4 presents the computational experiments results and section 5 concludes the paper.

II. PROBLEM DESCRIPTION AND FORMULATION

In this study, the movement of a container from its origin to destination is defined as a request. There are two types of...
requests in this paper: the loading requests and the discharging requests. The origin of a loading request is the location where the container is stored in the yard side, while the origin of a discharging request is the location of quay crane by which the container is unloaded from the vessel. The destination of a loading request is the location of the quay crane by which the container is loaded onto the vessel, and the destination of a discharging request is the location where the container is allocated to in the yard side for storage. As this paper focuses on the situation that several vessels are served by the same fleet of trucks in a planning period, the origin and destination of loading requests are known, and the origin of discharging requests is also given.

A soft time window for service, say \([a_i, b_i]\) of each container is generated by an operator in the planning horizon. It is a period of time which consists of the earliest possible time \(a_i\) and the due time \(b_i\). A container cannot be served before the earliest possible time, and \(b_i\) can be viewed as a penalty.

The processing time \(t_i\) is the period of time that a truck processes request \(i\) from its origin to destination, and the setup time \(s_{ij}\) is the period of time that a truck travels from the destination of current request \(i\) to the origin of next request \(j\). The start time \(w_i\) of request \(i\) is the time when it is started, and the completion time of a request is the time when it is finished. The difference between completion time and \(b_i\) of request \(i\) is the delay \(d_i\) of request \(i\). In this paper, we assume there are limited numbers of trucks with each truck serving only one route, which means each vehicle route is a one-to-one assignment of truck. Let \(R\) be the set of routes, \(K\) be the set of storage locations and \(J\) be the set of requests.

The following decision variables are used to describe the problem studied in this paper:

\[
x_{ik} = 1, \text{if container } i \text{ is allocated to storage location } k; \quad 0, \text{otherwise.}
\]

\[
y_{ij} = 1, \text{if request } i \text{ is connected to request } j \text{ in the same route}; \quad 0, \text{otherwise.}
\]

In this paper, the problem is to determine how to schedule yard trucks and allocate both loading and discharging containers with the aim of minimizing the weighted summation of total delay and total yard trucks travel time. The problem formulation is shown as follows.

**Objective Function:**

\[
Z = \alpha_1 \sum_{i \in J} d_i + \alpha_2 \left( \sum_{i \in J} t_i + \sum_{i,j \in J} s_{ij} y_{ij} \right)
\]  

(1)

**Subject to:**

\[
\sum_{i \in J} x_{ik} \leq 1 \quad \forall k \in K
\]

(2)

\[
\sum_{k \in K} x_{ik} = 1 \quad \forall i \in J
\]

(3)

\[
\sum_{j \in J} y_{ij} = 1 \quad \forall i \in J
\]

(4)

\[
\sum_{i \in J} w_i = 1 \quad \forall j \in J
\]

(5)

\[
w_j \geq a_i \quad \forall i \in J
\]

(6)

\[
d_i \geq w_i + t_i - b_i \quad \forall i \in J
\]

(7)

\[
w_j + M(1 - y_{ij}) \geq w_i + t_i + s_{ij} \quad \forall i \in J \text{ and } \forall j \in J
\]

(8)

In the objective function (1), \(\alpha_1\) and \(\alpha_2\) represent the weights of total delay and total travel time of yard trucks, respectively. Constraints (2) ensure that each storage location will be assigned with at most one discharging container. Constraints (3) ensure that each discharging container will be assigned with one storage location. Constraints (4) ensure that \(y_{ij} = 1\) if yard truck processes request \(j\) after request \(i\). Constraints (5) ensure that \(y_{ij} = 1\) if yard truck processes request \(i\) before request \(j\). Constraints (6) ensure that requests can only be served after earliest possible time. Constraints (7) calculate the delay of each request. Constraints (8) give the relationship of the starting time of a request and that of its successor. \(M\) is a large positive number. Due to the integrated problem of yard truck scheduling and storage allocation, it is considered as an NP-hard problem [8]. In the next section, we therefore propose a genetic algorithm to obtain near optimal solution for the integrated problem.

### III. A Proposed Genetic Algorithm

This paper proposes a genetic algorithm (GA) to solve the yard truck scheduling and storage allocation problems. The details of the proposed GA are given as follows.

#### A. Chromosome Representation

In the proposed GA, permutation encoding (real number coding) is used. A chromosome of the GA represents a solution of the yard truck scheduling and storage allocation problems. Each chromosome consists of \(|J|+|R|\) genes. Two types of genes are used, positive number gene and negative number gene. A positive number gene represents a request, and the sequence of the gene, from the left to the right, represents the sequence of the requests being performed. A positive number gene contains information of container ID, origin location, destination location and the time window of the request. A negative number gene represents the route number, while the positive number genes between two adjacent negative number genes represent the corresponding requests in the same route. A chromosome of the proposed GA can be generated using the following steps.

**Step 1:** Randomly allocate a storage location for each discharging request, so that each gene contains the information of origin, destination and sequence of each request.

**Step 2:** Randomly allocate all negative number genes into the chromosome, then the number of request in each route can be calculated.

**Step 3:** Randomly allocate all the requests to all routes. Then the requests and the requests’ sequence in each route can be obtained.
B. Fitness Value and Selection

The objective of the proposed GA is to minimize the weighted summation of total delay and total yard truck travel time. Thus, the fitness value of chromosome can be the reciprocal of its objective function value, as shown in (9). In this way, the best chromosome, corresponding to the scheduling of trucks and allocation of discharging containers with minimum weighted summation of total delay and total travel time, can be found. In this paper, the initial pool is randomly generated and the roulette wheel selection proposed by Holland [9] is used for selection operator.

\[
\text{Fitness} = \frac{1}{Z} \quad (9)
\]

C. Crossover Operation

Many studies (Whitley et al. [10], Blanton [11]) have shown that instance-specified information can make GA searching process more effective. In the yard truck scheduling and storage allocation problems, both the earliest possible time ai and the due time bi are important instance-specified information. In the proposed GA, the two kinds of instance-specified information are tried to be inherited with crossover operation. Considering the crossover operation of using two parents, P1 and P2, to reproduce two offspring, O1 and O2, the procedure of the proposed crossover operation is shown in the following steps.

Step 1: Add all the requests in route one in both parent P1 and parent P2 into set \( \Omega_1 \). Delete the duplicated requests in \( \Omega_1 \). Let set \( \Omega_2 \) is the same as \( \Omega_1 \).

Step 2: Rank the requests in \( \Omega_1 \) in non-decreasing order of their earliest starting time and let set \( \Phi \) be the ranked set. Rank the requests in \( \Omega_2 \) in non-decreasing order of their due time and let set \( \Psi \) be the ranked set.

Step 3: Insert the requests in the corresponding route in O1 according to their order in set \( \Phi \) and delete the inserted requests from set \( \Omega_1 \). Then, insert the requests in the corresponding route in O2 according to their order in set \( \Psi \) and delete the inserted requests from set \( \Omega_2 \).

Step 4: Add all the requests in the next route in both parent P1 and parent P2 into set O1. Then, delete the duplicated requests and the requests that have been inserted in O1. Add all the requests in the next route in both parent P1 and parent P2 into set O2. Then, delete the duplicated requests and the requests that have been inserted in O2.

Step 5: Repeat steps 2-4 until all the routes are assigned.

D. Mutation Operation

Mutation operation can help GA to prevent premature convergence and obtain the global optimal solution. In the proposed GA, three types of information, storage locations of discharging containers, sequence of requests in each route, and amount of requests in each route, in each chromosome are able to be changed. Thus, each chromosome can be mutated in three ways. The first way is to randomly choose a discharging request and change the request’s storage location into another empty storage location. The second way is to randomly select two positions and swap the requests on these positions. The last way is to change the amount of requests in two routes. In the following section, a well performed mutation strategy, which is a combination of these three mutation methods, is proposed based on the results of a series of computational experiments.

IV. COMPUTATIONAL EXPERIMENTS

In this section, a series of computational experiments are used to examine the performance of the proposed GA. The GA is coded with the use of Java Language and executed on a PC with Intel Core i7 3.4 GHz and 8 GB RAM. Instances used in the experiments are created based on the following criteria:

1. Both the origin and destination of loading containers, the origin of discharging containers, and the destination of storage locations are generated through a two–dimensional uniform distribution in the square from (0,0) to (1500,1500) (unit: meter).
2. The earliest start time of the requests are randomly generated from a uniform distribution of U(0, 1500) (unit: second), and the length of time window of the requests are generated from a uniform distribution of U(200, 500) (unit: second).
3. The trucks travel at the speed of 11.11 m/s (40 km/h).

We assume the two weight parameters have the relation of \( \alpha_1 + \alpha_2 = 1 \) and \( \alpha_1 \) equals to 0.6 as described by Lee [2].

A. A Strategy of Mutation

As three different mutation ways have been proposed in section 3, the mutation rate should be decided for each mutation way. An experiment, as shown in Table I, is proposed to evaluate the effect on the objective values of each mutation way. Another experiment, as shown in Table II, is proposed to find out the relation among the three ways of mutation by changing the order of the three mutation ways. In both experiments, there are 20 containers for loading, 20 containers for discharging, 50 storage locations and three trucks to be operated. The parameters used in these two experiments are set as follows: population size = 100; mutation rate \( P_{M_1} \); crossover rate \( P_c \).08.

As we can see in Table I, a smaller objective value can be obtained when using the first and the second ways of mutation than using the third way. With the increment of generation, the objective value of the first two ways become smaller. Thus, we set the mutation rate of the first way \( P_{M_1} \) as 0.9; mutation rate of the second way \( P_{M_2} \) as 0.9; mutation rate of the third way \( P_{M_3} \) as 0.2. We further force on the order of the three ways of mutation operations, shown in Table II.

It is clear from Table II that we can get the best objective value by first taking the first mutation way, second taking the third mutation way, and third taking the second mutation way.

B. Performance of the Proposed GA

To evaluate the performance of the proposed GA, a serious of experiments is conducted. CPLEX is employed to obtain the exact results of random small size instances as a comparison. Then, several large size instances are considered. Based on the preliminary tests, the parameters of GA used in small size instances are set as: population size = 100; mutation rate \( P_{M_1} = 0.9, P_{M_2} = 0.9, P_{M_3} = 0.2 \); crossover rate \( P_c = 0.8 \); and
maximum number of generations is 100. For the large size instances, the population size, mutation rate, crossover rate and maximum number of generations are set as 500, $P_{M1} = 0.9$, $P_{M2} = 0.8$, $P_{M3} = 0.2$, $P_{C} = 0.8$ and 300, respectively.

As shown in Table III, it is obvious that the proposed GA can obtain a near optimal solution in reasonable time when the instance size is small. Due to the interacting of yard truck scheduling problem and storage allocation problem, CPLEX costs hours to solve each single instance, while the proposed GA, as a comparison, only uses a few seconds to solve.

<table>
<thead>
<tr>
<th>Experiment No.</th>
<th>Generation</th>
<th>Objective value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td>997</td>
</tr>
<tr>
<td>2</td>
<td>40</td>
<td>948</td>
</tr>
<tr>
<td>3</td>
<td>60</td>
<td>890</td>
</tr>
<tr>
<td>4</td>
<td>80</td>
<td>856</td>
</tr>
<tr>
<td>5</td>
<td>100</td>
<td>819</td>
</tr>
</tbody>
</table>

### V. CONCLUSIONS

This paper proposed a genetic algorithm to deal with the integrated problem of yard truck scheduling and storage allocation. The crossover operation of the proposed GA is based on the information of job’s ready time and due time. The mutation operator combines three mutation ways. The GA is described in detail and proposed to find effective solutions. It is proved to obtain near optimal solutions in reasonable time through a series of computational experiments for small size instances. Larger size instances will be considered in the future.

### ACKNOWLEDGMENT

The work described in this paper was partially supported by a Teaching Company Scheme project (Project No. ZW1H (TCS162)), and partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 510410). The authors also would like to thank The Hong Kong Polytechnic University Research Committee for the financial and technical support.

### REFERENCES


