컨테이너 터미날에서의 유전자 해법을 이용한 적하계획법*

Load Scheduling Using a Genetic Algorithm in Port Container Terminals

김갑환** · 김기영** · 고창성***

Kap Hwan Kim · Ki Young Kim · Chang Seong Ko

Abstract

An application of the genetic algorithm(GA) to the loading sequencing problem in port container terminals is presented in this paper. The efficiency of loading operations in port container terminals is highly dependent on the loading sequence of export containers. In order to sequence the loading operation, we have to determine the route of each container handling equipment (transfer crane or straddle carrier) in the yard during the loading operation. The route of a container handling equipment is determined in a way of minimizing the total container handling time. An encoding method is developed which keeps intermediate solutions feasible and speeds up the evolution process. We determine the sequence of each individual container which the container handling equipment picks up at each yard-bay as well as the visiting sequence of yard-bays of the equipment during the loading operation. A numerical experiment is carried out to evaluate the performance of the algorithm developed.

key word : Container terminal, Loading sequence, Genetic algorithm

1. Introduction

The efficiency of loading operations in port container terminal is highly dependent on the loading sequence of export containers. In this paper, we apply a genetic algorithm to determine the route of the handling equipments(straddle carrier(SC), transfer crane(TC)) in a way of minimizing the total container handling time of the container handling equipments.

Cho developed a methodology for containership load planning[4]. In the paper, he formulated an integer programming model to assign a specific container to a specific cell in a bay of the ship. Although his formulation gives us the insight for the load scheduling problem, it is impossible to get the optimal solution for real size problems. Gifford presented a heuristic procedure for a containership load planning in a transtainer-based container port[7]. Chung et al. proposed a strategy of

* This paper was supported(in part) by 1994 Non Directed Research Fund. Korea Research Foundation

** 부산대학교 산업공학과

*** 경성대학교 산업공학과

providing buffer spaces in the apron area in order to increase the utilization of material handling equipments and reduce the total container loading time, which is evaluated by a simulation technique[5].

Other similar problems can be utilized to solve the load sequencing problem, although they are not directly related to port container terminals[9,11,13]. A typical example is the routing problem of a manual picker within a simple warehouse. This problem involves the minimization of the travel distance of the picker for a given list of picking items[8,12]. Other examples are the vehicle routing problem[2,10] and the traveling sales man problem. But the load sequencing problem in this paper is different from the previous researches in that the number of containers to be picked up at each visiting yard-bay has to be determined in addition to the visiting sequence of yard-bays for the handling equipment. And the handling equipment is allowed to revisit a yard-bay at any multiple times.

In the following section, we first define the routing problem of container handling equipment during the container loading operation in port container terminals in more details. In section 3, we suggest a mathematical formulation of the routing problem. In section 4, we shortly review the scheme of the general genetic algorithm and suggest a genetic algorithm for the routing problem of container handling equipment. In section 5, the results of the computational experiment are shown. In section 6, a solution for a practical example is illustrated. Finally, summary and conclusion are provided in section 7.

2. Problem Descriptions

In the following, it is described how load planners determine the loading sequence of export containers:

Based on the load profile provided by the ship company and the availability of gantry cranes, planners firstly determine the unloading and loading sequence of ship-bays in the containership which we call the "work schedule". Major considerations are given here to prevent a gantry crane from interfering with another. The work schedule shows the number of containers to be discharged and loaded in each ship-bay of the containership. And it also shows the sequence of ship-bays in which each gantry crane performs the handling operation. A work schedule is illustrated in Figure 1. The load planning by the work schedule is to pick up sequentially "5 containers of group A", "5 containers of group B", "3 containers of group C" and so on. The first subtour of the container handling equipment corresponds to a visiting sequence of yard-bays for the container handling equipment to pick up 5 containers of group A. By connecting subtours we can get a complete tour for a container handling equipment.

sequence (subtour number)	1	2	3	4	5	6
group	A	B	C	A	D	C
quantity	5	5	3	7	6	4

Figure 1. An illustration of a work schedule

Once the work schedule has been determined, the load planner assigns a specific container on the yard to a specific cell(slot) of the containership and determines the sequence of loading containers. In the process, the planner considers the work schedule of each gantry crane, the load profile, and the yard map which shows the storage locations of containers in the yard. Yard map keeps various informations of containers on the yard which include container number, destination, shipping company, and weight.

In this paper, we try to minimize the travel distance of the container handling equipment by optimizing the yard-bay sequence in which it visits during loading operation and the number of containers to pick up at each visiting yard-bay simultaneously.

The loading schedule has to satisfy the following two constraints: (1) it must satisfy the requirement of the work

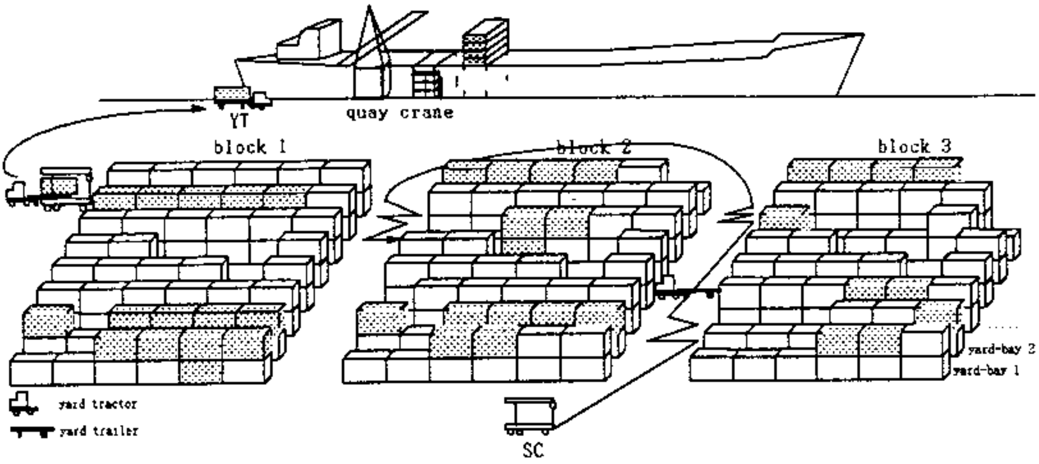


Figure 2. Container loading using straddle carrier in port container terminals

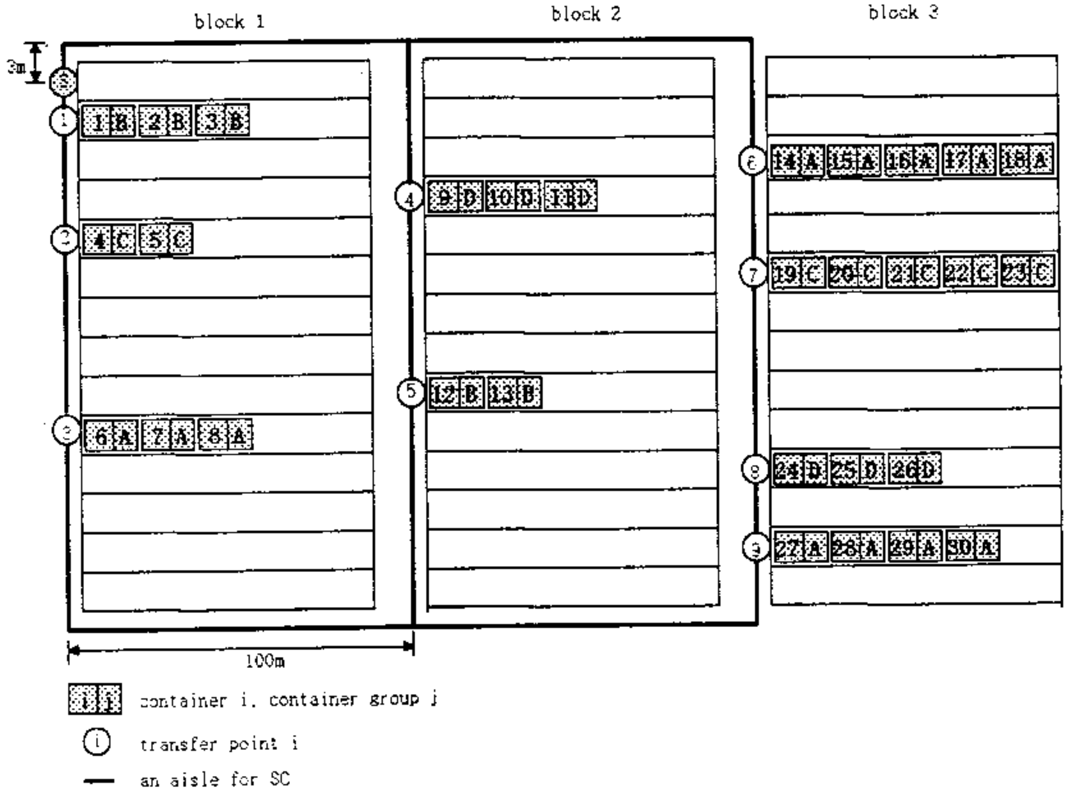


Figure 3. The distribution of containers in SC system

schedule of the corresponding gantry crane and (2) the total number of containers of each group picked up at each yard-bay must be equal to the initial number of containers of the corresponding group.

The method of the load scheduling depends on the type of the container handling equipment. In this paper, we assume that straddle carriers(SC) are used as the handling equipment.

In a container terminal using straddle carrier, a yard-slot is identified by its block number, yard-bay number, row number, and tier number. Figure 2 shows an example of container loading using straddle carrier in port container terminal. A yard-map shows the distribution of containers of each container group in the yard as in Figure 3.

To load a container in the yard onto a ship, the straddle carrier(SC) moves over a yard-bay, picks up a container, moves it to the end of the yard-bay, and loads it onto a yard trailer. Then, the yard tractor/trailer(YT) transports the container to a gantry crane. Finally, the gantry crane picks up the container and loads it onto the ship. Since the YTs may not move over yard-bays, the SC has to move between the location of the target yard-slot and the end point of the yard-bay whenever the SC transfers a container to a YT. Thus, the relative locations of two consecutive containers within the same yard-bay do not affect the total travel distance. But, if two consecutive containers are located in different yard-bays from each other, the SC has to travel between yard-bays which is a main source of inefficiencies in the loading operation. Since the load sequence of containers affects only the total travel time of the SC between yard-bays, we try to reduce the travel time of SC.

3. Mathematical Formulation

We first define a "tour" as a visiting sequence of yard-bays by a SC in order to pick up all the specified containers in the corresponding work schedule. By a "subtour" of a SC, we mean a visiting sequence of yard-bays during which a SC picks up all the containers for a cluster of cells in the ship (See Figure 1). For example, in order to carry out the work schedule in Figure 1, we have to schedule a tour consisting of six subtours each of which corresponds to "picking up five A containers", "five Bs", "three Cs", "seven As" and four Cs, respectively. By connecting subtours, we can get a complete tour for a SC.

The following notations are used to formulate the carrier routing problem:

n=the number of subtours which constitute a complete tour of a SC

B=the set of indices of yard-bays, $B=\{1,2,3....,m\}$

G=the set of indices of container groups, $G=\{1,2,3....,l\}$

c_{hj}=the initial number of containers of group h stacked at yard-bay j

$S(h)$=the set of indices of subtours whose corresponding container group number is h

r_t=the number of containers to pick up during subtour t

d_{ij}=the travel distance between yard-bay i and j,

The tour of a SC may be expressed as a route on a network which may be constructed as follows:

A network may be represented by a set of vertices (V) and a set of arcs (A). In our formulation, a vertex is represented as t^i where t is the subtour number and i is the yard-bay number(See Figure 4). Then, the problem is to find a path on the network which starts from node S and ends at node T and to determine simultaneously the number of containers to pick up at each node during the tour. Of course, the total number of containers to pick up during a subtour should be the same as the number specified in the work schedule. And note that the total number of containers of each group picked up at each yard-bay may not exceed the initial available amount in the bay.

We introduce some more notations for the following explanation:

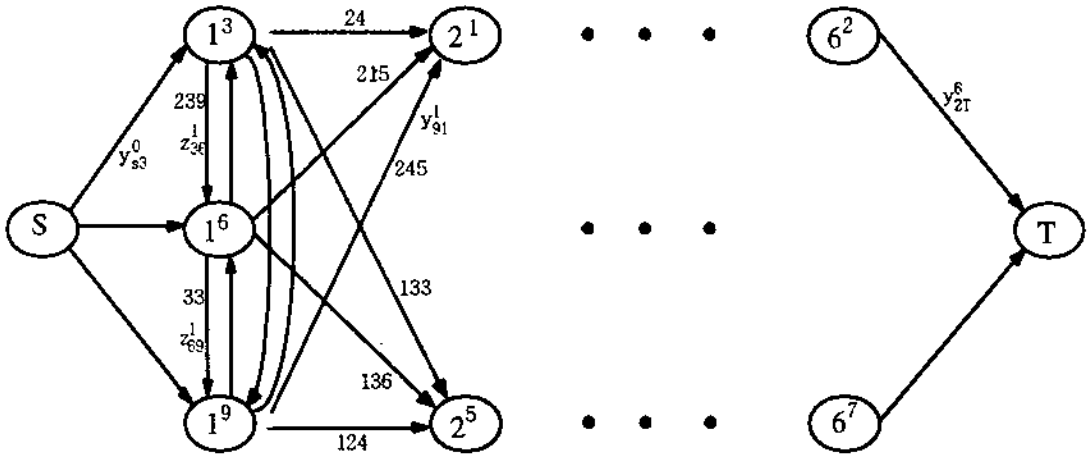


Figure 4. A network representation of the problem

$A'V)$=the set of arcs, $A(V)=\{(i,j) \mid i,j\in V\}$, given a set of vertices V,

t=subtour number, t =0, 1, ..., n, n+1 where t=0 and n+1 at source and terminal vertices in the network representation which implies the initial and the final location of the SC, respectively,

$$y^t_{ij}=\begin{cases}1, \text{ if the SC moves yard-bay } i \textit{ yard-bay } j \textit{ after completing subtour } t\\ 0, \textit{ otherwise (a decision variable),}\end{cases}$$

and, y^0_{sj} and y^m_{jT} denote the first and final movement of the SC during the tour, respectively,

$$Z^t_{ij}=\begin{cases}1, \text{ if the SC moves yard-bay } i \textit{ yard-bay } j \textit{ during subtourt}\\ 0, \textit{ otherwise (a decision variable),}\end{cases}$$

X^t_j=the number of containers picked up at yard-bay j during subtour t (a decision variable).

Then, the problem may be formulated as follows:

$$\underset{x,y^t_{ij},z^t_{ij}}{\text{Min}} \sum_{t=0}^{m-1}\sum_{(i,j)\in A(B)} d_{ij}y^t_{ij}+\sum_{t=1}^{m}\sum_{(i,j)\in A(B)} d_{ij}z^t_{ij} \tag{1}$$

subject to

$$\sum_{j\in B} y^0_{Sj}=1 \tag{2}$$

$$-\sum_{j\in B} y^m_{jT}=-1 \tag{3}$$

$$\sum_{j\in B}(y^{t-1}_{ji}+z^t_{ji})-\sum_{j\in B}(y^t_{ij}+z^t_{ij})=0,\ j\in B,\ t=1,\ 2,\dots,\ m \tag{4}$$

$$\sum_{(i,j)\in A(w)} z^t_{ij}\le |w|-1, \text{ for all } w\subseteq B,\ t=1,\ 2,\ \dots,\ m \text{ where } |w| \text{ denotes the cardinality of set } w, \tag{5}$$

$$x^t_j\le M(\sum_{i\in B} z^t_{ij}+\sum_{i\in B} y^{t-1}_{ij}),\ j\in B,\ t=2,\ \dots,\ m,$$

$$x^1_j\le M(\sum_{i\in B} z^1_{ij}+\sum_{i\in B} y^0_{Si}),\ j\in B,\ ,\ t=1,$$

$$\text{where } M \text{ is a sufficiently large number,} \tag{6}$$

$$\sum_{j\in B} x^t_j=r_t,\ t=1,\ 2,\ \dots,\ m, \tag{7}$$

$$\sum_{t\in S(h)} x^t_j=c_{hj},\ h\in G,\ j\in B, \tag{8}$$

$$y^t_{ij}\in\{0,1\},\ i,\ j\in B,\ t=1,\ 2,\ \dots,\ m-1,$$
$$y^0_{Sj}\in\{0,1\},\ j\in B,$$
$$y^m_{iT}\in\{0,1\},\ i\in B,$$
$$z^t_{ij}\in\{0,1\},\ i,\ j\in B,\ t=1,\ 2,\ \dots,\ m,$$

$x_j^t \geq 0,\ j \in B,\ t=1,\ 2,\ \ldots,\ m.$

The objective function is to minimize the total travel distance of the SC. Since the total travel distance within a yard-bay is constant regardless of the loading sequence of containers, we consider only the inter-bay movements of the SC in order to evaluate the total travel distance.

The constraints (2)-(4) represent the flow conservation. A feasible solution corresponds to a path from the source node to the terminal node on the network. The constraints (2) and (3) represent the gain of flows at source node and the terminal node, respectively, while the constraint (4) represents the flow conservation at the other nodes. The constraint (5) is to prevent the looping of subtours. We may have an isolated cycle in the final solution which is not connected to the path from the source to the terminal node. The constraint (6) comes from the definition of variables. These constraints imply that only when the SC visits a bay, it can pick up containers at the bay. The constraint (7) implies that the number of containers picked up in a subtour should be equal to that of containers requested by the work schedule. The constraint (8) means that the total number of containers picked up during the whole tour should be equal to the initial number of containers at each bay for each specific container group.

LINDO program was used to solve the above mixed integer programming for an example problem. We developed a code generation program to convert the input data into the mixed integer programming code which LINDO can read and process. The computational time increased rapidly as the problem size grows. For example, it took about 9 hours on IBM/PC 486 to solve an example problem, with only 6 yard-bays, 12 container groups and 16 subtours. It is why we suggest a search procedure based on a genetic algorithm.

In the following section, we adopt a genetic algorithm in order to determine the number of containers to be picked up by the SC at a specific yard-bay.

4. A Genetic Algorithm for Routing the Container Handling Equipment

Genetic algorithm(GA) is a heuristic search technique that imitates the natural selection and the biological evolutionary process. GA combines the notion of survival of the fittest, random but yet structured search, and parallel evaluation of nodes in the search space. GA has been successfully applied to various combinatorial optimization problems. The traveling salesman problem (TSP) is one of the typical NP-hard problem which is solved successfully by GA. It has a similar structure to the routing problem of container handling equipment treated in this paper.

A genetic algorithm is characterized by a string representation (genes) of nodes in the search space, a fitness function to evaluate the search nodes, and a stochastic assignment to control the genetic operators.

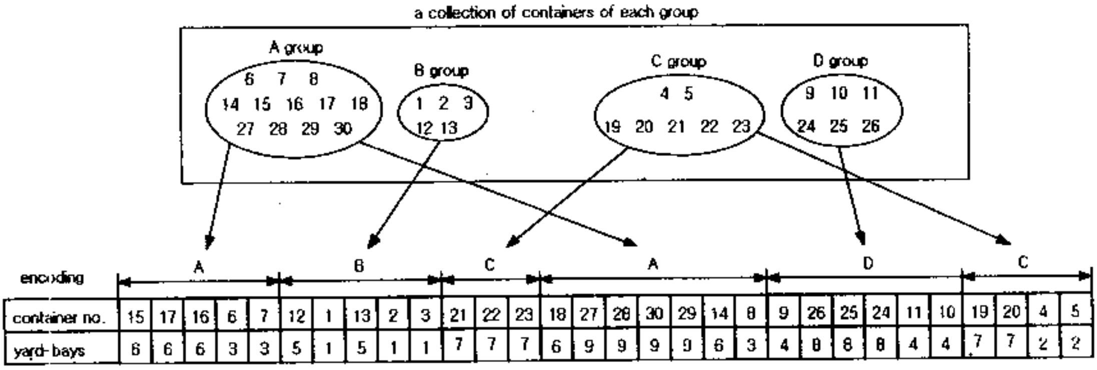
4.1 Encoding

The first step of designing a genetic algorithm is to define an encoding method which enables us to map all the nodes in the search space into a set of symbolic strings. This is a crucial phase that affects all the subsequent steps.

The encoded string in this paper consists of symbols each of which corresponds to a specific container. A permutation of the container numbers is a chromosome which corresponds to a possible sequence. Every symbol (container number) appears only once in the chromosome. This coding method for the routing problem in this paper is similar to the one of TSP. The difference is that in this paper we construct an independent chromosome for each container group (a collection of containers of a same size and with a same destination port). That is, when chromosomes in the initial population are generated, each gene is generated based on the container group. We call this encoding strategy "group based encoding". Figure 5 illustrates a result of the group based encoding. In the encoding method, all the containers of a container group are assigned only to the section for the corresponding container group. For example, all the containers of container group A are assigned only to section A and all containers of container group B are assigned only to section B.

We summarize how to construct the chromosome as follows:

- Each chromosome (string) consists of several sections (subchromosomes). Each section in a string corresponds to a sequence list of containers of a specific container group. In Figure 5, digits from



encoding	A					B					C			A							D						C			
container no.	15	17	16	6	7	12	1	13	2	3	21	22	23	18	27	28	30	29	14	8	9	26	25	24	11	10	19	20	4	5
yard-bays	6	6	6	3	3	5	1	5	1	1	7	7	7	6	9	9	9	9	6	3	4	8	8	8	4	4	7	7	2	2

Figure 5. An illustration of encoding and decoding for the problem in Figure 1 and Figure 3

1-5 and 14-20 constitutes a section.

- We define a run as a subsection of a chromosome which consists of containers of a same group in a same yard-bay that are picked up consecutively. In Figure 5, digits 1-3 corresponds to a run. Then, the movement of the handling equipment depends on only the length of each run and the sequence of runs. The length of a run implies the number of containers of a group to be picked up in a specific yard-bay and the sequence of runs corresponds to the visiting sequence of yard-bays. For example, SC moves from initial position to yard-bay 6 and picks up 3 containers. And then, SC moves to yard-bay 3 and picks up 2 containers, and so on. Notice that each gene(container number) is encoded into an independent digit in the string, and that the relative locations of two consecutive genes within a run do not affect the total travel distance. Thus, the travel distance within a yard-bay is not considered in the evaluation of the fitness function.

4.2 Reproduction Operator

Reproduction operation is a process in which parents are selected from the total parent solution pool according to their fitness values. The parents with higher fitness values have a higher probability of contributing one or more offspring in the next generation. We chose a biased roulette wheel approach (what is called stochastic sampling with replacement) to implement the reproduction operator. In this approach, each string is given a specific percent of the biased roulette wheel. The percent is calculated by dividing its fitness value by the total fitness value of all strings. Each time we require another offspring, a simple spin of the weighted roulette wheel yields the reproduction candidate. In this way, strings with higher fitness values have a larger number of offsprings in the next generation. Once a string has been selected for reproduction, an exact replica of the string is made. This string is then enter into a mating pool, a tentative new population, for further genetic operation.

4.3 Crossover Operator

In the following, we define a crossover operation which keeps the resulting solutions feasible. Partially mapped crossover(PMX), order crossover(OX), and cycle crossover(CX) are commonly used for the permutation problems. The following strings, S1 and S2, are used to illustrate PMX operator:

S1= 2-4-5-3-8-9-6-1-7 and S2= 3-9-8-6-5-4-2-7-1

Firstly, two cut points are chosen at random. Suppose that two cut points are chosen as S1=2-4-5-|3-8-9|-6-1-7 and S2=3-9-8-|6-5-4|-2-7-1. Then, genes bounded by the cut points, |3-8-9| and |6-5-4|, are exchanged each other so that S1 would posses some new partial genetic information from S2 and vice versa. Then, we can get S1'=2-4-5-|6-5-4|-6-1-7 and S2'=3-9-8-|3-8-9|-2-7-1. However, S1' and S2' have illegal structures because of the duplication of genes. Therefore, the final step is to resolve the illegality by replacing those duplicated genes with the genes at the corresponding positions in the original chromosome. Two new offspring are S1''= 2-9-8-|6-5-4|-3-1-7 and S2''=6-4-5-|3-8-9|-2-7-1.

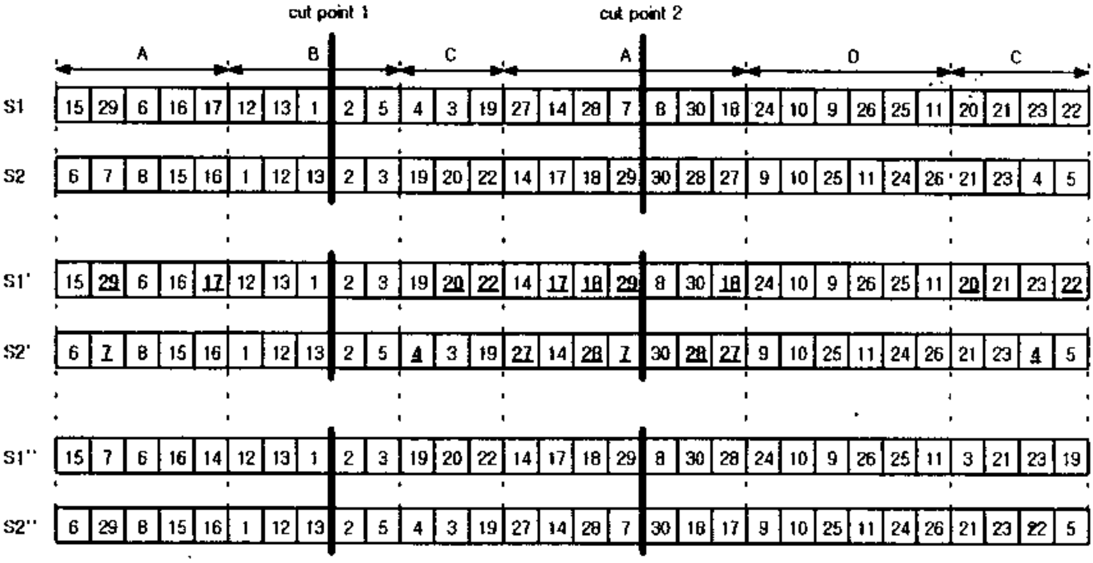


Figure 6. An illustration of crossover using PMX operator

All the PMX, OX, and CX are known to be successful in the traveling salesman problem. PMX tends to reflect the absolute gene positions because of point-to-point mapping, whereas OX tends to reflect the relative gene positions because of slide filling of empty spaces. For the detailed explanation of OX and CX operator, refer to Chan and Tansri's paper[3]. For OX and CX operators, genes for different container groups become mixed with each other after every crossover operation, which results in an infeasible solution. But, when using PMX operator, since the genes for a specific container group remain at the locations for the corresponding container group even after the crossover operation, the feasibility of the resulting solution is maintained. This is why we choose PMX as the crossover operator in this study.

Figure 6 shows an illustration of crossover using the PMX operator. Consider the strings, S1 and S2 in Figure 6. Firstly, two cut points are chosen at random. Then genes bounded by the cut points, (2-5-4-3-19-27-14-20-7) and (2-3-19-20-22-14-17-18-29) are exchanged(S1' and S2'). The illegal structure because of the duplication of genes is resolved by replacing those duplicated genes with the genes at the corresponding positions in the original chromosome(S1'' and S2'').

4.4 Mutation Operator

Mutation is operated on a single chromosome. Firstly, one container group is chosen at random. And then, two elements (container number) in sections of the chromosome corresponding to the chosen container group are selected at random and exchanged with each other. We call this mutation "group based swap mutation".

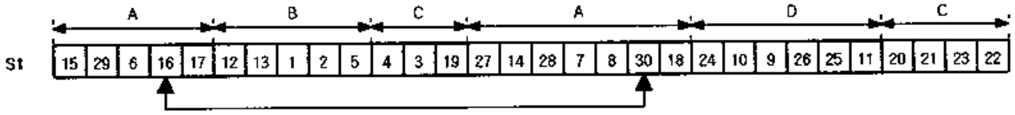
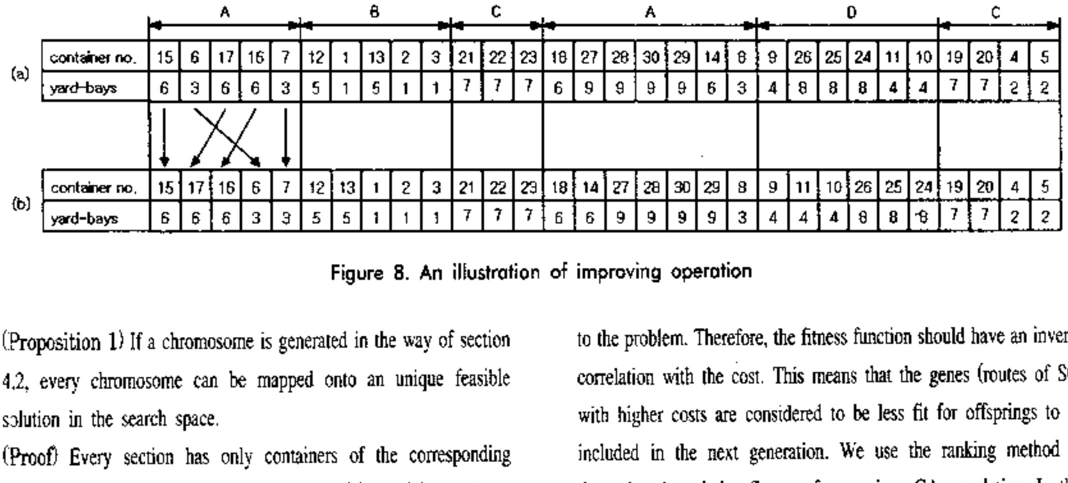


Figure 7. An illustration of class based swap mutation operator

Figure 7 illustrates the process of application of the swap mutation operator. Suppose that container group A is selected at random. Then, two elements in sections corresponding to container group A are selected randomly and are exchanged with each other.

4.5 Improving Operator

Obviously, it is unreasonable to visit a yard-bay multiple times during a subtour as in (a) of Figure 8. Thus, whenever we find a chromosome with the case of multiple visits is found, we modify the chromosome by moving containers in a same yard-bay to the positions succeeding the one of the first occurrence as in (b) of Figure 8.



		A					B					C			A							D						C			
(a)	container no.	15	6	17	16	7	12	1	13	2	3	21	22	23	18	27	28	30	29	14	8	9	26	25	24	11	10	19	20	4	5
	yard-bays	6	3	6	6	3	5	1	5	1	1	7	7	7	6	9	9	9	9	6	3	4	8	8	8	4	4	7	7	2	2
(b)	container no.	15	17	16	6	7	12	13	1	2	3	21	22	23	18	14	27	28	30	29	8	9	11	10	26	25	24	19	20	4	5
	yard-bays	6	6	6	3	3	5	5	1	1	1	7	7	7	6	6	9	9	9	9	3	4	4	4	8	8	8	7	7	2	2

Figure 8. An illustration of improving operation

(Proposition 1) If a chromosome is generated in the way of section 4.2, every chromosome can be mapped onto an unique feasible solution in the search space.

(Proof) Every section has only containers of the corresponding container group in the section and constraint (7) and (8) are always satisfied when chromosomes are generated in the way of section 4.2. Note that any permutation of containers of a same container group in a section doesn't destroy the feasibility of the solution. Q.E.D.

(Proposition 2) When the PMX operation, the group-based swap mutation, and the improving operation are performed in the routing problem of container handling equipment, every child chromosome resulting from parent chromosomes which are feasible to constraint (7) and (8) can also be mapped onto a feasible solution.

(Proof) Refer to the proof of proposition 1. The resulting string after applying the above three operations is also a permutation of parent strings which corresponds to a feasible solution. Q.E.D.

From Propositions 1 and 2, we can conclude that if every chromosome in the initial population is mapped into a feasible solution, every chromosome in a population of the succeeding generations can also be mapped into a feasible solution. Since we consider only the feasible solutions during the search process, the computational time can be significantly reduced.

4.6 Fitness Function

We use the objective function (1) of the routing problem to evaluate the fitness function.

The fitness function in a GA is a measure of goodness of a solution to the problem. Therefore, the fitness function should have an inverse correlation with the cost. This means that the genes (routes of SC) with higher costs are considered to be less fit for offsprings to be included in the next generation. We use the ranking method to determine the relative fitness of genes in a GA population. In this method, the population is sorted by the objective function value. Every individual is then assigned an offspring count that is solely a function of their rank.

5. Computational Experiments

In order to test the performance of the genetic algorithm in this paper, we solved a problem with a single yard block. As a test problem, we chose the work schedule in Figure 1 and the yard map in Figure 9. The work schedule consists of loading tasks for 30 containers which may be classified into 4 container groups. In this test data, the moving time per unit yard-bay distance is assumed to be 1.

For evaluating the performance of the genetic algorithm, we developed a computer program using C language which was run on a Pentium-based IBM-PC compatible. We tested the sensitivity of

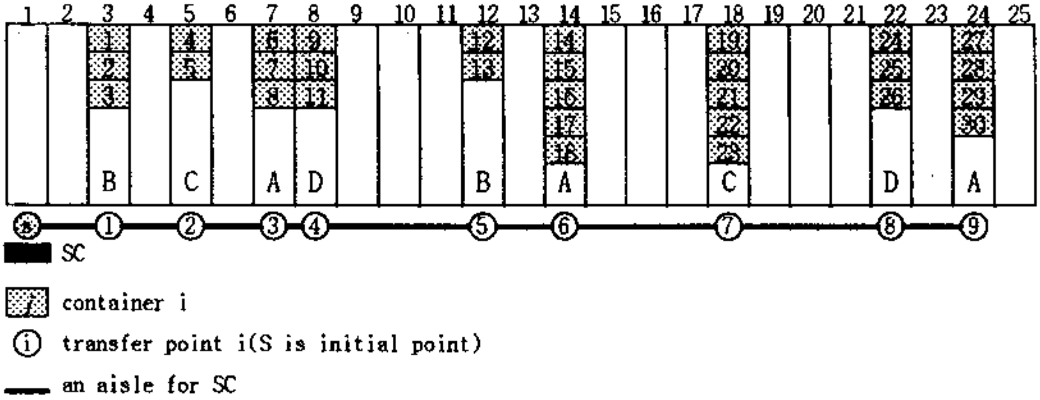
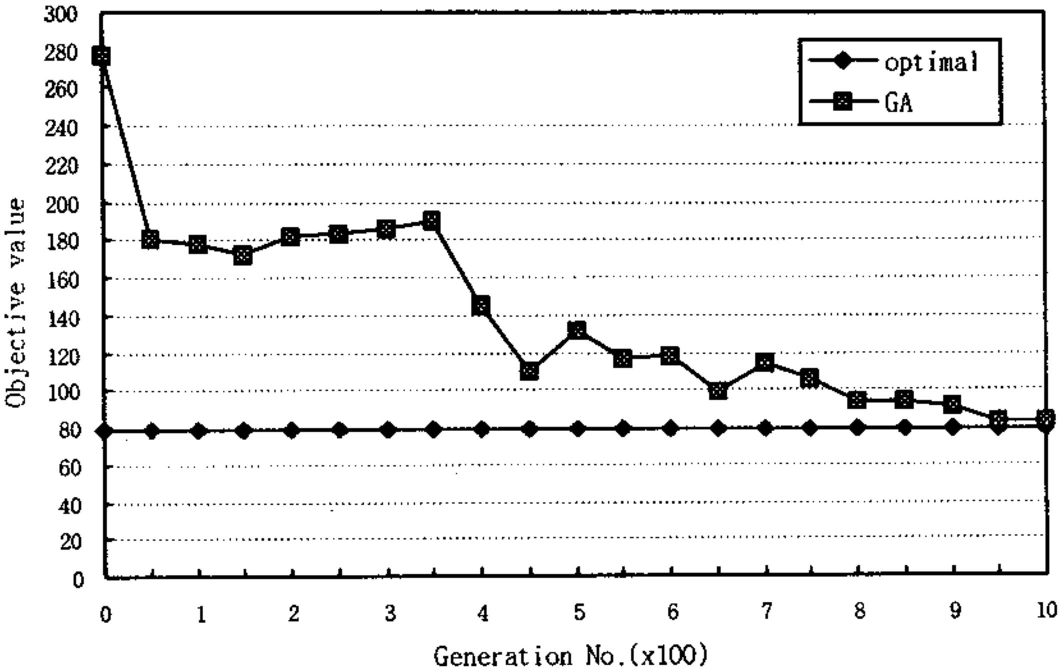


Figure 9. The distribution of containers in a single block



(Population=400, Generation=1000, Crossover=0.7, Mutation=0.1)

Figure 10. The trend of the objective function value in the test problem

Table 1. The average objective function value of solutions for 5 test runs

population size	crossover / mutation	0.5			0.7			0.9		
		mean	worst	best	mean	worst	best	mean	worst	best
200	0.05	111.8	126	99	98.8	122	92	92.0	94	85
	0.15	107.0	125	92	103.8	114	94	103.2	114	95
	0.25	104.4	113	92	101.4	112	92	105.2	114	92
600	0.05	98.2	114	92	93.4	106	81	92.4	94	92
	0.15	93.4	107	81	96.0	105	81	89.2	92	79
	0.25	94.4	100	92	90.4	94	81	98.0	106	92

Table 2. The average run time for 5 test runs

unit=second

	crossover	0.5			0.7			0.9		
	mutation	0.05	0.15	0.25	0.05	0.15	0.25	0.05	0.15	0.25
population size	200	69.8	72.0	78.8	86.4	88.0	93.8	202.2	205.4	173.4
	600	496.5	504.6	533.2	659.6	657.0	643.0	1584.0	1590.2	1260.2

various GA parameters to the solution.

The first parameter tested is the number of strings in every generation, so called the population size, which is the size of the parallel search. It is known that the necessary population size highly depends on the problem characteristics and needs to be determined experimentally. We examined two population sizes, such as 200 and 600units.

The second parameter tested is the number of generations. The solutions normally converge to a near- optimal point as the number of generations increases. The necessary number of generations may also depend on the problem characteristics related to other genetic parameters. Therefore, it has also to be determined experimentally. The performance of the algorithm is evaluated until the number of generations reaches 1000 time units.

The third parameter tested is the percentage of the crossover and the mutation. The probability of the crossover was set to be 0.5, 0.7, and 0.9, respectively. The probability of mutation tested is set to 0.05, 0.15, and 0.25, respectively.

Figure 10 shows the trend of the average value of the objective function as the number of generation increases. We can see that the average of the objective function value converges to the optimal (The optimal solution is 79) or near-optimal solution. Five solutions are obtained for each combination of parameters in Table 1. It shows the mean, the worst and the best of the 5 solutions. The population size turned out to be the most important factors to affect the quality of the solution. Although the best solution was obtained when the probability of the crossover and the mutation is 0.9 and 0.15, respectively, for the population size of 600, the number of samples is not sufficient to conclude with confidence that it is the best combination of the probabilities. The objective function value of the best solution obtained by the genetic algorithm is within 16% above the one of the optimal solution for all cases of the population size of 600. Table 2 shows the average run time over the 5 test runs for each parameters. The average run time is largely dependent on the population size. Thus, as the population size increases, the run time increases rapidly. And the average run time is also dependent on the probability of the crossover and the mutation. Figure 11 compares the objective function values for the different population

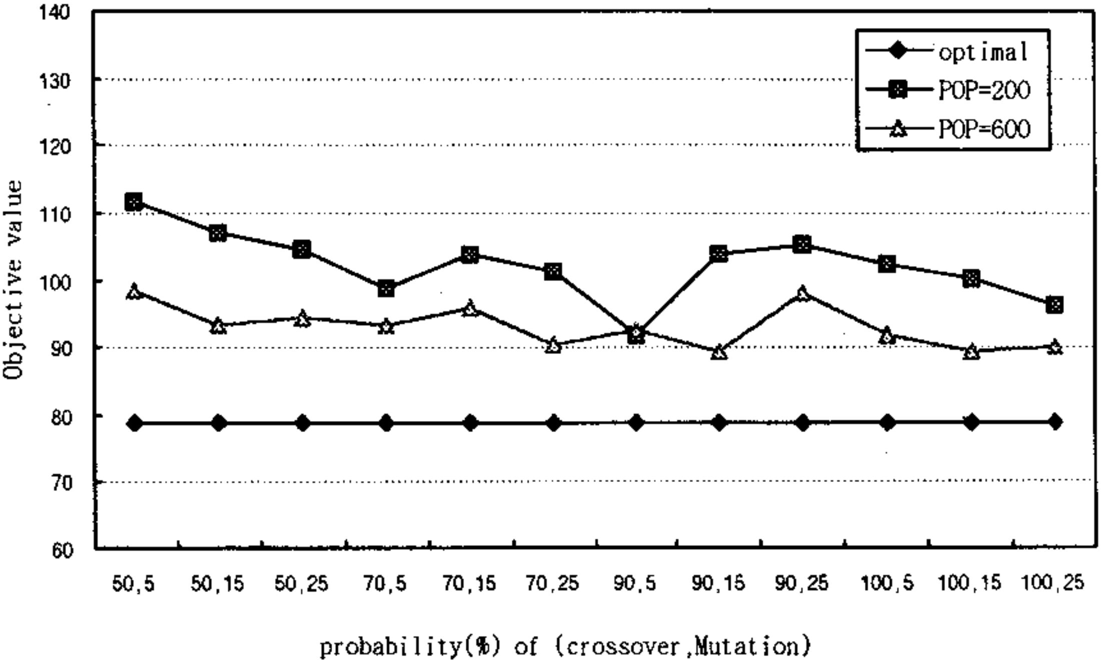


Figure 11. Objective function values for the different population sizes(POP)

sizes. The objective function value for all cases of the population size of 600 was smaller than one for all cases of the population size of 200.

6. A Practical Example

A practical problem using a real data in a Pusan Container Terminal is solved by the algorithm in this paper. Two straddle carriers(SCs) are utilized in this example, which is a typical case of the loading operation in the Pusan container terminal. In Figures 12 and 13, the structure of the containership and the load profile are provided. Based on the load profile, the work schedule for each SC is obtained as in Figure 14. The work schedule is in the form of the sequence of clusters and the number of containers of each cluster

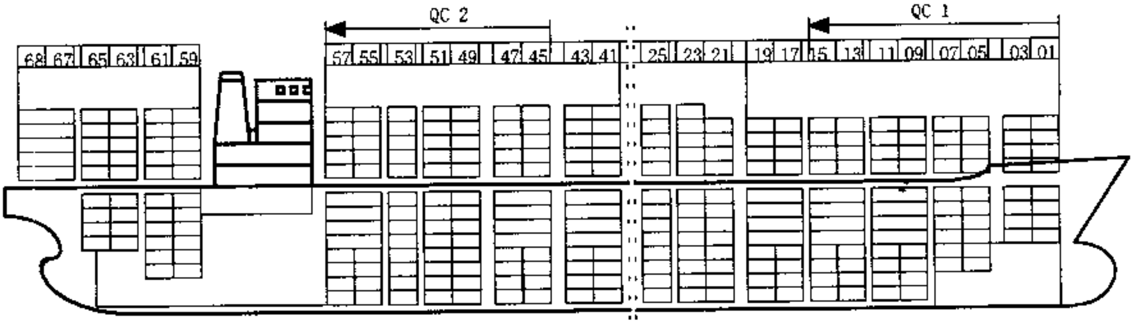


Figure 12. A practical example of containership

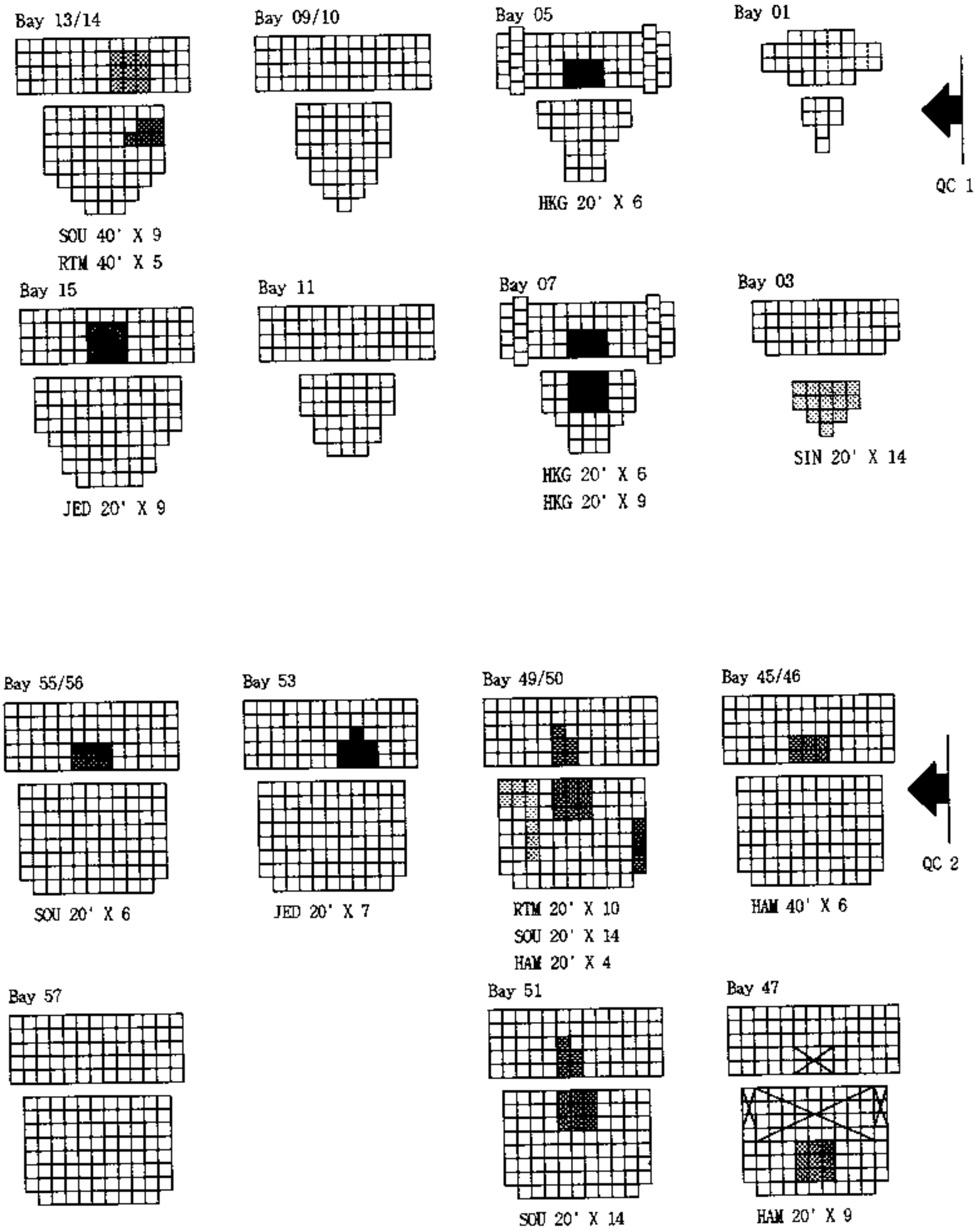


Figure 13. A practical example of load profile

which each SC is requested to pick up. For example, the work schedule of QC 1 is "to pick up 14 containers whose destination port is Singapore and whose length is 20 feet, and then "to pick up 21 containers whose destination port is Hong Kong and whose length is 20 feet and so on". From the work schedule, we get the number of containers for each subtour. The total number of

QC 1 (SC 1)	sequence (subtour number)	1	2	3	4	5
	container group	SIN 20	HKG 20	RTM 40	SOU 40	JED 20
	quantity	14	21	5	9	9

GC 2 (SC 2)	sequence (subtour number)	1	2	3	4	5	6	7
	container group	HAM 40	HAM 20	RTM 20	HAM 20	SOU 20	JED 20	SOU 20
	quantity	6	9	10	4	28	7	6

Figure 14. A work schedule based on the load profile in Figure 11

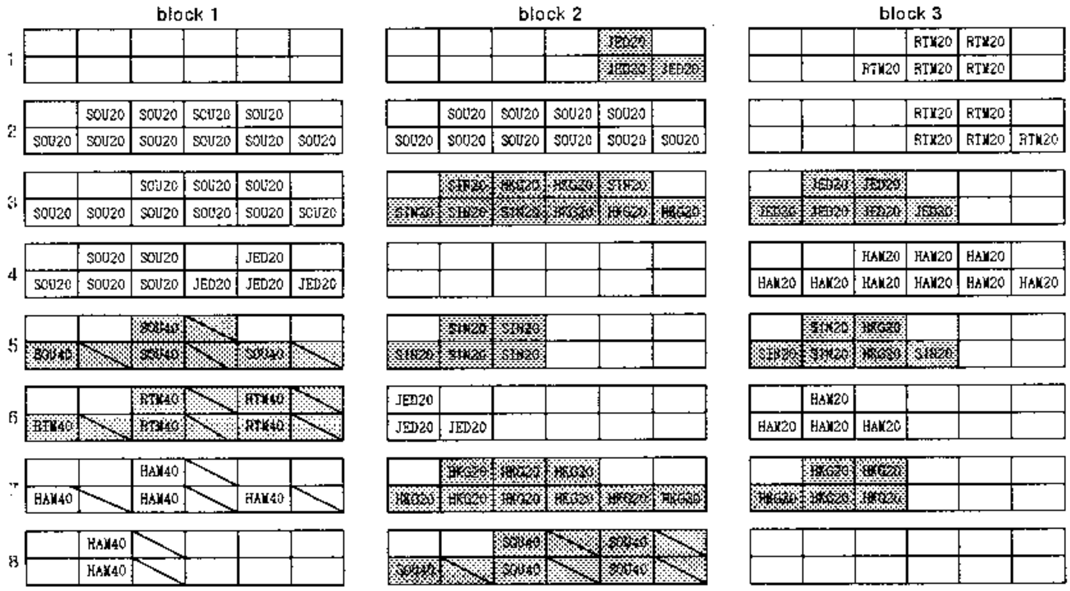


Figure 15. A practical example of yard map simplified somewhat

containers to be loaded is 128. The total number of containers assigned to QC 1 is 58. The total number of containers to be loaded by QC 2 is 70.

The initial number of containers of each group in each yard-bay is provided in Figure 15. The distance between two adjacent blocks is assumed to be 100m while the distance between two adjacent yard-bays is 3m. The moving time per meter and the setup time per positioning are assumed to be 1 and 0, respectively. The layout of the container yard was modified for the convenience of analysis. Each block is assumed to have 8 yard-bays although a block usually consists of 20-22 yard-bays in practice. But, this simplification does not distort the performance evaluation of the algorithm. And each SC is assumed to move on the shortest travel path to get to another position.

sequence		1	2	3	4	5	6	7	8	9	10	11	12
block/bay	1/1	2/3	2/5	3/5	3/5	3/7	2/7	2/3	1/6	1/5	2/8	2/1	3/3
container group	init	SIN20	SIN20	SIN20	HKG20	HKG20	HKG20	HKG20	RTM40	SOU20	SOU20	JED20	JED20
quantity		5	5	4	2	5	9	5	5	4	5	3	6

(a) the route of SC 1

sequence		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
block/bay	1/1	1/7	1/8	3/6	3/4	3/2	3/1	3/4	3/6	2/2	1/3	1/2	1/4	1/4	2/6	2/2	1/2	1/4
container group	init	HAM40	HAM40	HAM20	HAM20	RTM20	RTM20	HAM20	HAM20	SOU20	SOU20	SOU20	SOU20	JED20	JED20	SOU20	SOU20	SOU20
quantity		4	2	2	7	5	5	2	2	7	9	8	4	4	3	3	2	1

(b) the route of SC 2

Figure 16. The final solution for each SC for the practical problem

Firstly, the containers in the yard are split into two groups based on the proximity of containers to the location of the corresponding gantry crane(QC). For example, QC 1 is requested to load 9 containers of JED 20, while QC 2 7 containers of the same group. There are 4, 6, and 6 containers in blocks 1, 2, and 3, respectively. Since QC 1 is supposed to work in the right-hand side of QC 2, 9 containers located rightward are assigned to QC 1, which are marked by shading. And then, we determine the visiting sequence of yard-bays for each SC. The population size and the number of generation were set to be equal to 400 and 1000, respectively. The probability of crossover and the probability of mutation were set to be equal to 0.7 and 0.1, respectively.

Figure 16 shows the result of the genetic algorithm in order to determine the visiting sequence of each SC. The total travel times of SC 1 and SC 2 are 750 and 765, respectively. Thus, the objective function has the value of 1515 for the solution. The run time on a Pentium-based IBM-PC compatible was 809 seconds (389 seconds for SC 1, 420 seconds for SC 2).

7. Conclusions

In this paper, we introduce a genetic algorithm to determine the load sequence of export containers in a port container terminal. The peculiarity of this encoding system is that it always produces feasible chromosomes. We try to minimize the total travel time of the straddle carriers which are a popular material handling equipment in the yard. The sequence of individual containers are determined although only the travel time between different yard-bays is considered in the evaluation of the fitness function. A numerical experiment is carried out to evaluate the performance of the GA.

Other objective function than the travel time of straddle carrier (SC) such as the stability of the vessel should be studied additionally in order to cover all aspects of the load sequencing problem. Note that each container in the yard is encoded into a independent digit in the string. Thus, it is a strong point of the genetic algorithm in this paper that we can easily incorporate additional constraints into or objective functions GA for determining the sequence of individual containers.

Further researches are needed to develop more efficient and sophisticated genetic algorithms for the loading sequencing problem, although the results in this paper gave us a promising prospect for GA applications.

References

[1] Bazaraa, M. S. and Jarvis, J. J., *Linear Programming and Network*, John Wiley & Sons, pp574-585, 1992.

[2] Bertsimas, D. J., "A Vehicle Routing Problem with Stochastic

Demand," *Operations Research*. Vol.40, No.3, pp574-585, 1992.

[3] Chan, K. C., and Tansri, H, "A Study of Genetic Crossover Operations on the Facilities Layout Problem," *Computers ind. Engng.*, Vol.26, No.3, pp537-550, 1994.

[4] Cho, D. W., *Development of a Methodology for Containership Load Planning*. Ph.D. Thesis, Oregon State University, 1982.

[5] Chung, Y. G., Randhawa, S. U. and McDowell, E. D., "A Simulation Analysis for a Transtainer-based Container Handling Facility," *Computers ind. Engng*. Vol. 14, No.2, pp113-125, 1988.

[6] Croce, F. D., Tadei, R., and Volta, G., "A genetic Algorithm for the Job Shop Problem," *Computers Ops Res.* Vol.22, No.1, pp15-24, 1995.

[7] Gifford, L., *A Containership Load Planning Heuristic for a Transtainer based Container Port*. M.Sc. Thesis, Oregon State University, 1981.

[8] Goetschalckx, M. and Ratliff, H. D., "Order Picking in an Aisle," *IIE Transactions*. Vol.20, No.1, pp53-62, 1988.

[9] Hall, R. W., "Distance Approximations for Routing Manual Pickers in a Warehouse," *IIE Transactions*. Vol. 25, No.4, pp76-87, 1993.

[10] Li, C. L., David, S. L. and Martin, D., "On the Distance constrained Vehicle Routing Problem," *Operations Research*. Vol.40, No.4, pp790-799, 1992.

[11] Ong, H. L. and Huang, H. C., "Asymptotic Expected Performance of some TSP Heuristics: An Empirical Evaluation," *European Journal of Operational Research*. Vol. 43, pp231-238, 1989.

[12] Randolph, W. H., "Distance Approximations for Routing Manual Pickers in a Warehouse," *IIE Transactions*. Vol.25, No. 4, pp76-87, 1993.

[13] Rao, M. R., "A Note on the Multiple Traveling Salesman Problem," *Operations Research*. Vol.28, No.3, pp629-632, 1980.

[14] Tate, D. M., and Smith, A. E., "A Genetic Approach to the Quadratic Assignment Problem". Computers *Ops. Res*. Vol.22, No.1, pp73-83, 1995

97년 2월 최초 접수, 97년 10월 최종 수정