J. Ind. Eng. Int., 6 (11), 39-50, Spring 2010
ISSN: 1735-5702

Yard crane scheduling in port container terminals using genetic algorithm

H. Javanshir [1]*; S. R. Seyedalizadeh Ganji [2]

[1]*Assistant Professor, Dep. of Industrial Engineering, Islamic Azad University, South Tehran Branch, Tehran, Iran*

[2]*M. Sc., Dep. of Transportation Engineering, Islamic Azad University, Science and Research Branch, Tehran, Iran*

Received: 12 June 2008; Revised: 27 August 2008; Accepted: 1 December 2008

Abstract: Yard crane is an important resource in container terminals. Efficient utilization of the yard crane significantly improves the productivity and the profitability of the container terminal. This paper presents a mixed integer programming model for the yard crane scheduling problem with non- interference constraint that is NPHARD in nature. In other words, one of the most important constraints in this model which we can mention to yard crane non- interference constraint is that they usually move on the same rails in the yard block. Optimization methods, like branch and bound algorithm, has no sufficient efficiency to solve this model and become perfectly useless when the problem size increases. In this situation, using an advanced search method like genetic algorithm (GA) may be suitable. In this paper, a GA is proposed to obtain near optimal solutions. The GA is run by MATLAB 7.0 and the researchers used LINGO software which benefits from the Branch and Bound algorithm for comparing outputs of GA and the exact solution. We should consider the abilities of the LINGO software which is not capable of solving the problems larger than 5 slots to 3 yard cranes. The computational results show that the proposed GA is effective and efficient in solving the considered yard crane scheduling problem.

Keywords: *Container transportation; Genetic Algorithm (GA); Yard crane scheduling; Mathematical programming*

1. Introduction

Container-related transportation activities have grown remarkably over the last 10 years and the trend does not show any sign of slowing down as illustrated by the annual world container traffic figures, in millions of TEUs (20 feet equivalent container units), displayed in Table 1 (Crainic and Kim, 2005). In a port container terminal, a number of vessels are often berthed alongside, and each vessel is served by multiple quay cranes which are supported by a large number of yard cranes in the yard. Fig. 1, depicts the typical flow of containers in a container terminal.

Given the multi-criterion nature, the complexity of operations, and the size of the entire operations management problem, it is impossible to make the optimal decisions that will achieve the overall objectives. Logically, the hierarchical approach is adopted to break the whole problem into smaller sequential problems. The input to a problem is actually the output of its immediate predecessor, and is treated as a known quantity after the preceding problem is solved. Fig. 2 gives a typical hierarchical structure of operational decisions in a container terminal.

Container handling and storage operations include the management and handling of containers while they are in the yard storage and thus occur between the receiving and delivery operations and the ship operations. Container-handling equipment performs the placement of containers into yard storage and their retrieval when needed. Yard cranes move along blocks of containers to yard bays to perform these operations. Planning these operations is part of the equipment-assignment process, which allocates tasks to container-handling equipment. Based on the quay-crane schedule, one or two yard cranes are assigned to each quay crane for loading and unloading. The remaining yard cranes are allocated to receiving and delivery operations. Terminal operators aim to assign and operate yard cranes in such a way that inefficient moves and interferences among yard cranes are minimized. In this paper, the researchers' purpose is to consider a yard crane model with non-interference constraint by using Genetic Algorithm (GA). For this sake, the paper is organized as follows:

In Section 2, the researchers describe a literature review of yard crane problems. Section 3 presents the theoretical and mathematical description of Yard Crane Scheduling with Non-Interference

*Corresponding Author Email: hjavanshir@yahoo.com
Tel.: +98 912 33 307 26

constraints Problem (YCSNIP). The approximate solution procedure based on the GA is presented in Section 4. Section 5 provides us some computational examples and comparisons between Genetic and B&B algorithms. A large size problem also is considered in this section. Finally, the last section contains the most important findings of the paper.

2. Literature review

Recently, few studies have been conducted for yard cranes. The objectives considered in most cases were the total waiting time and the total delays of trucks. Lai and Lam (1994) and Lai and Leung (1996) proposed various dispatching rules for yard cranes and tested them by simulation. Zhang *et al.* (2002) and Cheung *et al.* (2002) solved static versions of the crane deployment problem when the total workload at each storage area is known in advance. Zhang *et al.* (2002) proposed a mixed-integer programming model and addressed it by a method based on Lagrangean relaxation.

Cheung *et al.* (2002) addressed a similar problem but removed the restriction that crane movements must be completed within a single period. This allows using a shorter period length resulting in a more accurate model. A successive piecewise-linear approximation method was proposed. Kim *et al.* (2003) addressed the problem of sequencing transfer tasks of a yard crane for outside trucks in dynamic situations where new trucks arrive continuously.

A dynamic programming model was suggested and decision rules derived by a reinforcement learning technique were proposed.

Table 1: World container traffic (Crainic and Kim, 2005).

Year	Container Year traffic (Millions TEU)	Growth rate(%)
1993	113.2	12.5
1995	137.2	9.8
1997	153.5	4.2
1999	203.2	10.0
2000	225.3	10.9
2001	231.6	2.8
2002	240.6	3.9
2003	254.6	5.8
2004	280.0	10.6
2005	304.0	8.6

Murty (2007) summarized some of problems in operating policies and the design of the terminal layout, and report on newer operating policies and design which can help improve performance.

He used the quay crane rate (QCR) as the measure to performance to maximize, and discussed the most important of the various factors influencing the QCR, such as: congestion on the road system inside the yard, yard crane overloading frequency and crane flashing frequency.

Finally, some of planning policies based on treating all yard crane's operating in a zone as a pool of yard crane's serving and alternative layout for export storage yard was presented.

Note that there are other sources in container ports which can be optimized. It is worth mentioning that some researches are studied GA in other container port’s field such as scheduling of trailers and trucks (Ng *et al.*, 2007).

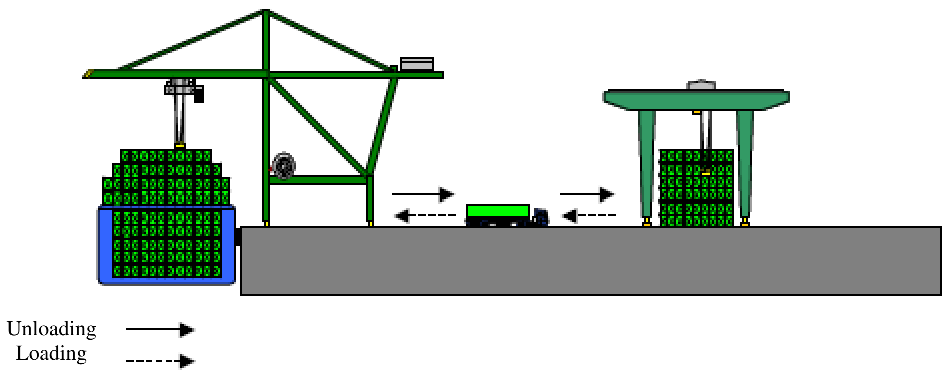

Figure 1: Typical flow of containers in terminal operations (Ng and Mak, 2005).

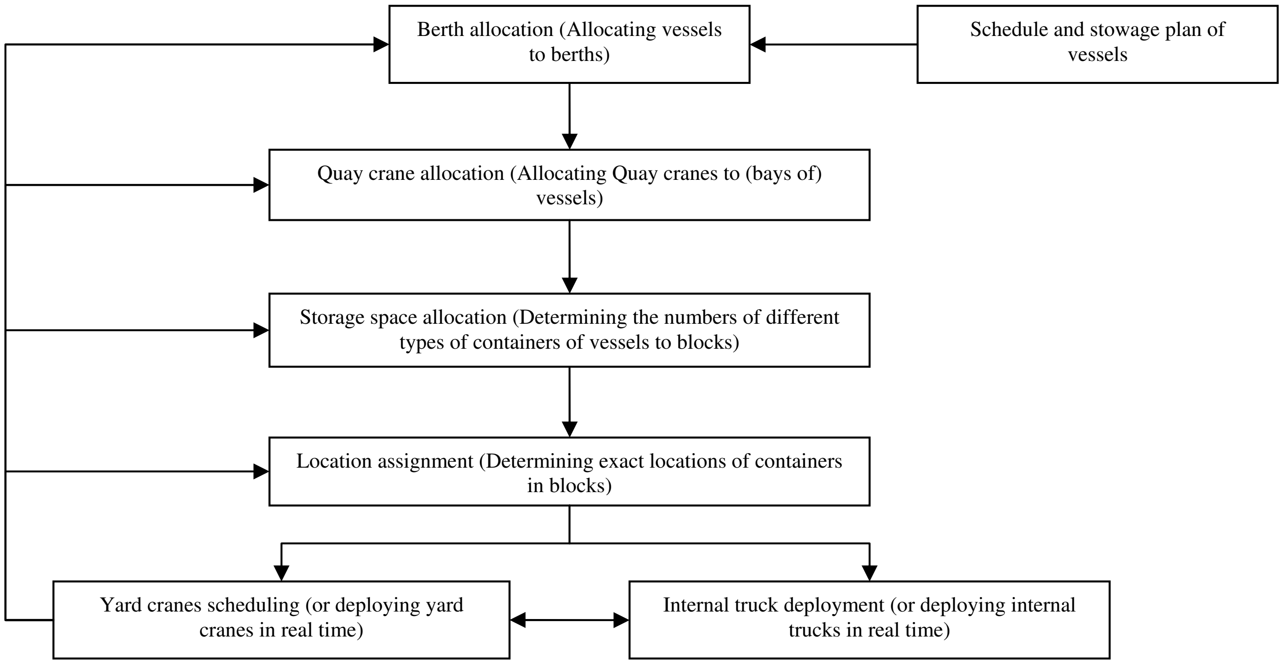

Figure 2: Hierarchical structure of operational decisions in a container terminal (Zhang , 2003).

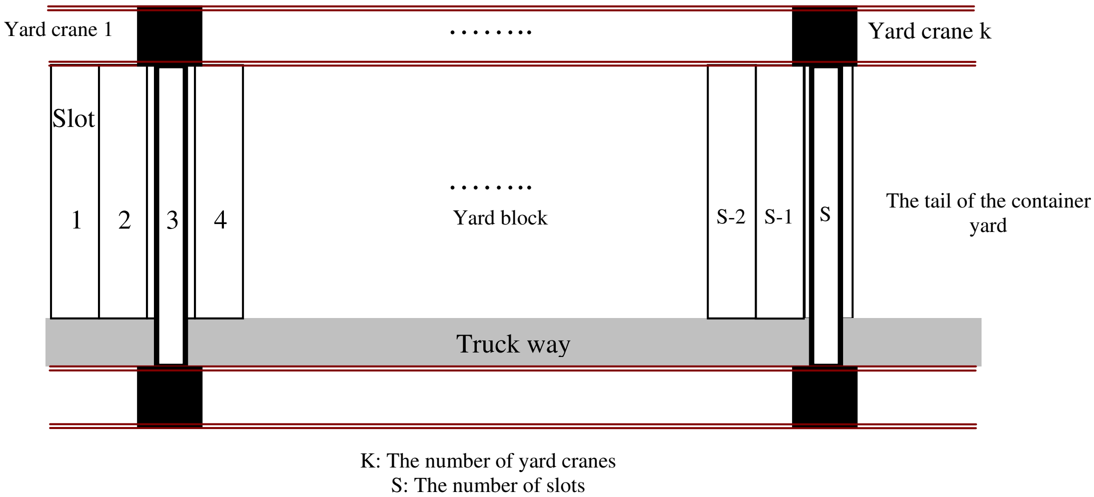

Figure 3: The illustration of the YCSNIP.

This paper focuses on the Yard Crane Scheduling with Non-Interference constraints Problem (YCSNIP) for single blocks. So, the rest of the paper is organized as follow:

Section 3 provides a mathematical modeling based on the Lee *et al*. (2008) presented model with different application for yard crane scheduling (not quay crane scheduling). A Solution procedure using the genetic algorithm for solving the mathematical formulation is introduced Section 4. The results of computational experiments in Section 5 show the proposed genetic algorithm is effective and efficient in solving the YCSNIP.

3. Problem description

According to Fig. 3, one single block divided into slots indicates the YCSNIP and shows both yard cranes and slots are arranged in an increasing order from the front the container block to the tail of it.

3.1. Problem assumption

The following assumptions are imposed in formulating the YCSNIP:

1. Yard cranes are on the same track and thus cannot cross over each other.

2. Only one yard crane can work on a slot at a time until it completes the slot.

3. Compared with processing time of a slot by a yard crane, travel time of a yard crane between two slots is small and hence it is ignored.

3.2. Mathematical modeling

In order to formulate the YCSNIP, the following parameters and decision variables are introduced:

3.3. Parameters

K The number of yard cranes,

S The number of slots,

P_s The processing time of slot s by a yard crane, $(1 \le s \le S)$,

M A sufficiently large positive constant number,

K The number of yard cranes.

3.4. Decision variables

$X_{s.k}$ 1, if slot s is handled by yard crane *K*; *0*, otherwise, $(1 \le s \le S, 1 \le k \le K)$,

Y_{s,s^1} 1, if slot s finishes no later than slot S^1 starts; 0, otherwise, $(1 \le s, s' \le S)$,

C_s The completion time of slot *S*, $(1 \le s \le S)$.

The YCSNIP can be formulated as follows:

$$\text{Minimize } Z = \max_s C_s \tag{1}$$

Subject to:

$$C_s - P_s \ge 0 \qquad \forall 1 \le s \le S \tag{2}$$

$$\sum_{k=1}^{K} X_{s,k} = 1 \qquad \forall 1 \le s \le S \tag{3}$$

$$C_s - (C_{s'} - P_{s'}) + Y_{s,s'} M > 0 \quad \forall 1 \le s, s' \le S \tag{4}$$

$$C_s - (C_{s'} - P_{s'}) - (1 - Y_{s,s'}) M \le 0 \qquad \forall 1 \le s, s' \le S \tag{5}$$

$$M\,(Y_{s,s'} + Y_{s',s}) \ge \sum_{k=1}^{K} k\, X_{s,k} - \sum_{l=1}^{K} l\, X_{s',l} + 1 \qquad \forall 1 \le s, s' \le S \tag{6}$$

$$X_{s,k}, Y_{s,s'} = 0 \text{ or } 1, \qquad \forall 1 \le s, s' \le S, \quad \forall 1 \le k \le K \tag{7}$$

The objective function (1) minimizes the make span of handling containers in the yard, which is the latest completion time among all slots. Constraints (2) define the property of the decision variable C_s. Constraints (3) ensure that every slot must be performed only by one yard crane. Constraints (4) and (5) define the properties of decision variables $Y_{s,s'}$: Constraints (4) indicate that $Y_{s,s'} = 1$ if $C_s \le C_{s'} - p_{s'}$, which means $Y_{s,s'} = 1$ when slot *s* finishes no later than slot s' starts; Constraints (5) indicate that $Y_{s,s'} = 0$ if $C_s > C_{s'} - p_{s'}$, which means $Y_{s,s'} = 0$ when slot *S* finishes after slot s' starts.

Finally, the interference between yard cranes can be avoided by imposing Constraints (6). Suppose that slots s and S^1 are performed simultaneously and $S<S^1$, then this means that $Y_{s,s'} + Y_{s's} = 0$. Note that both yard cranes and slots are arranged in an increasing order from the front to the tail of the yard. Thus, if yard crane *K* handles slot s and yard crane l handles slot s', then $k + 1 \le l$. Some applied examples about proposed decision variables $Y_{s,s'}$ (Table 5) and $X_{s,k}$ (Table 6), are given in Section 5.

4. Solution procedure using the genetic algorithm

To facilitate the solution procedure, this paper employs a Genetic Algorithm (GA) to obtain near optimal solution, which is widely used in solving difficult problems. GA is a search algorithm based on the mechanics of natural selection and natural genetics.

In general, there are three common genetic operators in a GA: selection, crossover, and mutation. The procedure of the proposed GA is illustrated in Fig. 4.

Note that the circumstance of chromosome representation is the most important part of GA that is explained in following parts.

4.1. Chromosome representation

For heuristic algorithms such as GA creating population of chromosomes is very important, because the correct chromosome representation results in increasing of solution speed. In this paper a chromosome of the GA represents a sequence of slots. Fig. 5 provides a sample chromosome, in which a gene is a slot number. Based on the sequence of slots represented by the chromosome, a yard crane schedule can be constructed using the procedure in Fig.6.

4.1.1. Fitness evaluation and selection

The problem (YCSNIP) is a minimization problem; thus, the smaller the objective function value is, the higher the fitness value must be. For this, the fitness function could be defined by the reciprocal of the objective function (Kim and Kim, 1996). However, the fitness value is set to be the reciprocal of its objective function value, as shown in Eq. (8), otherwise, the fitness value is zero.

$$Finttness = \frac{1}{Z} \qquad (8)$$

In this paper, a roulette wheel approach is adopted as the selection procedure. It belongs to the fitness-proportional selection and can select a new population with respect to the probability distribution based on fitness values (Gen and Cheng ,1996).

4.2. Crossover operation

The crossover scheme is widely acknowledged as critical to the success of GA. The crossover scheme should be capable of producing a new feasible solution (or child) by combining good characteristics of both parents.

Preferably, the child should be considerably different from each parent. This paper adopts two point crossovers (Gen and Cheng ,1996; Lee *et al.*, 2008), in which repairing procedure is embedded to resolve the illegitimacy of offspring. The probability of crossover operator's usage in any iteration of GA is considered 0.60 of the present generation. The details of the proposed crossover operation are elaborated as follows:

Step 1. Select a substring from one parent randomly.

Step 2. Produce a child by copying the substring into its corresponding positions.

Step 3. Delete the slots which are already in the substring from the second parent. The resulted sequence of slots contains the slots that the child needs.

Step 4. Place the slots into the unfixed positions of the child from left to right according to the order of the sequence to produce an offspring.

The procedure is illustrated in Fig. 7. It gives an example of making two offspring from the same parents.

4.3. Mutation operation

Mutation operator is another important part in GA that forces the algorithm to search new areas, and help the GA avoid premature convergence and find global optimal solution. This operator introduces random changes to the chromosomes by altering the value of a gene with a user-specified probability called mutation rate. In our application, if mutation is to occur in a gene, we generate two random numbers between 1 and the string length, which define positions within the chromosome. The value of a gene at these two positions is interchanged, thereby changing the order of handling (slots) sequence, to create a new chromosome. Based on our preliminary experiments, the mutation rate was set to 0.09. Fig. 8 presented the application of mutation operator in this procedure.

5. Computational examples

The GA algorithm was coded by MATLAB 7.0 using a Pentium IV-2.8 GHz PC with 512 MB RAM. As a comparison, LINGO software (Version 8.0) was used to exactly solve random instances with small sizes. It is worth mentioning LINGO solves the programming problems applying B&B method. In addition, Lingo as commercial software is not able to solve the problem larger than 5 slots and 3 yard cranes. Problems with dimensions of 3×2, 4×2, 4×3, 5×2, and 5×3 have been used to compare the obtained results from two methods. The overall procedure of B & B method is shown in Fig. 9.

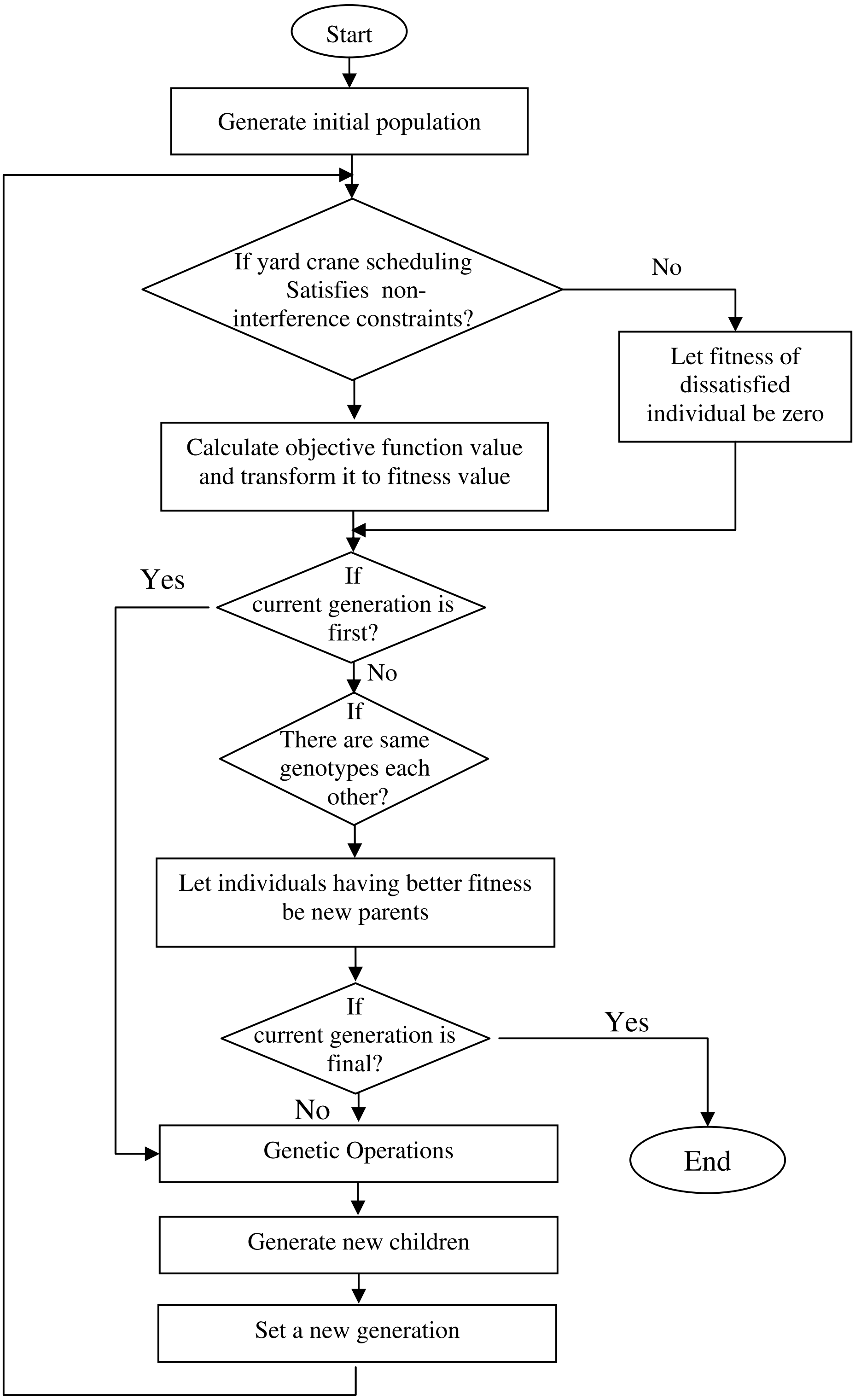

Figure 4: The flow chart of the proposed GA.

Slot number	*1*	*5*	*9*	*2*	*3*	*8*	*6*	*4*	*7*
Gene number	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*

Figure 5: An illustration of the chromosome representation.

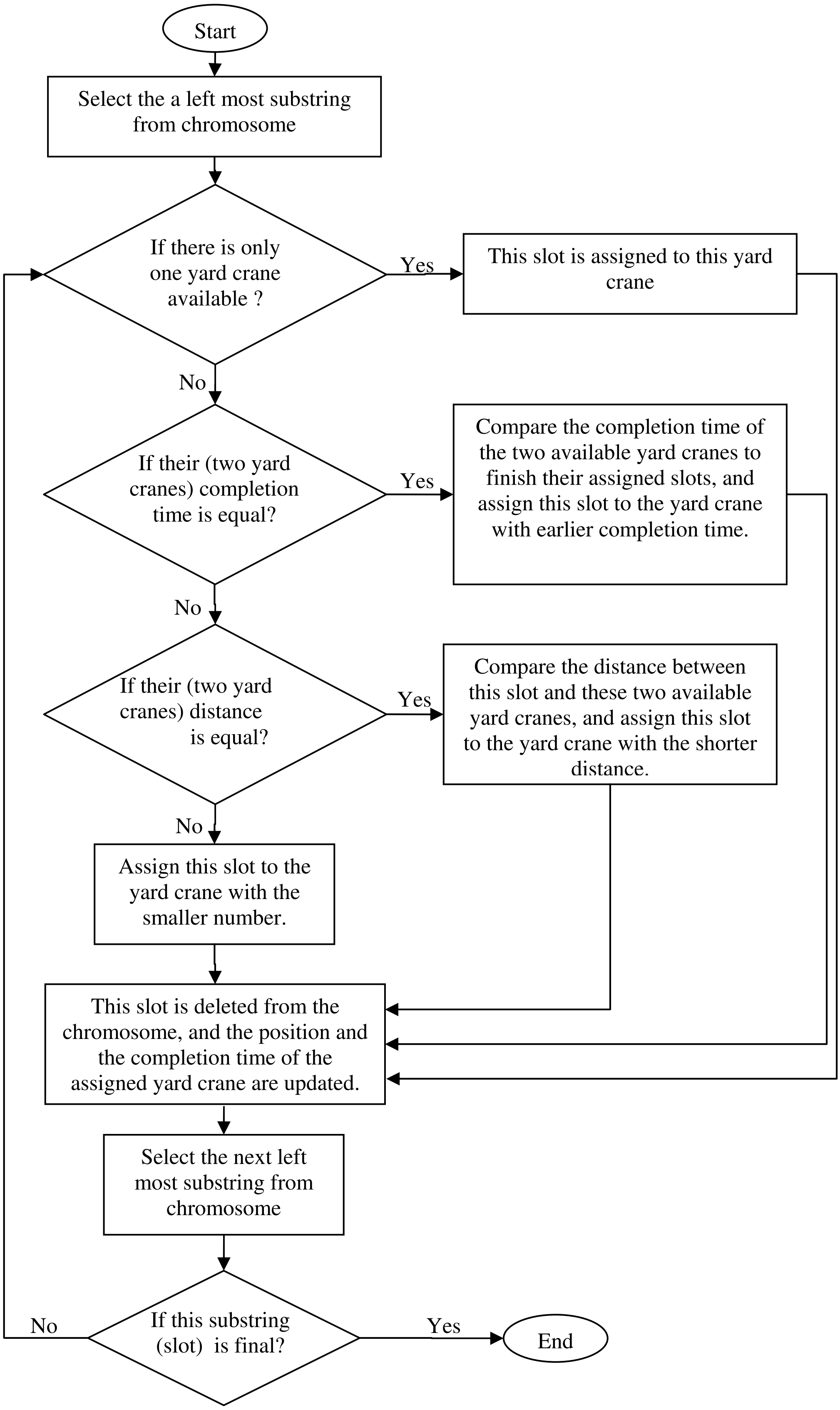

Figure 6: The flow chart of the proposed the procedure for chromosome analysis.

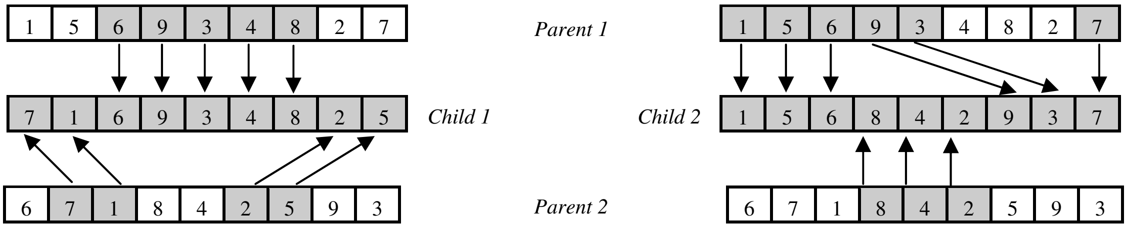

Figure 7: An illustration of the presented crossover operator.

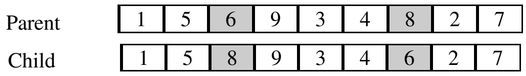

Figure 8: An illustration of the mutation.

Table 2: The processing time of slot *S* by a yard crane (min).

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
P_s	30	50	20	62	44	50	38	26	28	70	36	22	46	60	30	40	28	72	64	32

Problems used in the experiments were generated randomly, but systematically. We developed thirteen random instances with small and medium sizes. The processing time of slots is shown in Table 2. As shown in Table 3, in these problems, the number of yard cranes ranges from 2 to 3 and the number of slots ranges from 3 to 9. Among these thirteen problems, only five problems are solved by the B & B method that was equal with GA solution method. So, GA can be known as a suitable algorithm for solving the above-mention scheduling model.

The noticeable point is about the completion time of slots that may be the same for different problems. This situation can not be a reason for equal service of cranes to slots. This means that in one problem this time may be belonged to one crane and in another belonged to 2 or 3 cranes. For this reason, the completion times of cranes and objective function values are presented in Table 3.

Table 4 illustrates the similar pattern for the large size examples with 10, 15 and 20 slots serving by 2 and 3 yard cranes.

Output of an instance presented in Table 4, with 20 slots processed by 3 cranes, can be observed in Tables 5 and 6, in detail and in the form of decision variables, $Y_{s,s'}$ and $X_{s,k}$.

We can see the applicable results of the model in Table 7 which comes from Tables 5 and 6. For example, according to available data in Table 5, slot 5 earlier than slot 4, 10, 14 and 16 is processed that can be seen in Table 7.

In addition, the completion time of slots is 210, 286, 312 and 256, respectively. According to related outputs to decision variables $X_{s,k}$, we can also observe how to allocate the cranes to available slots in yard, as shown in Table 7.

In optimal situation, slots 1, 9, 7, 5, 4, 3, 2 by crane 3 and slots 6, 8, 10, 12, 15, 18 by crane 2 and also the rest of slots by crane 3 are processed. Finally, the maximum completion time of the mentioned problem was obtained in 312 minutes.

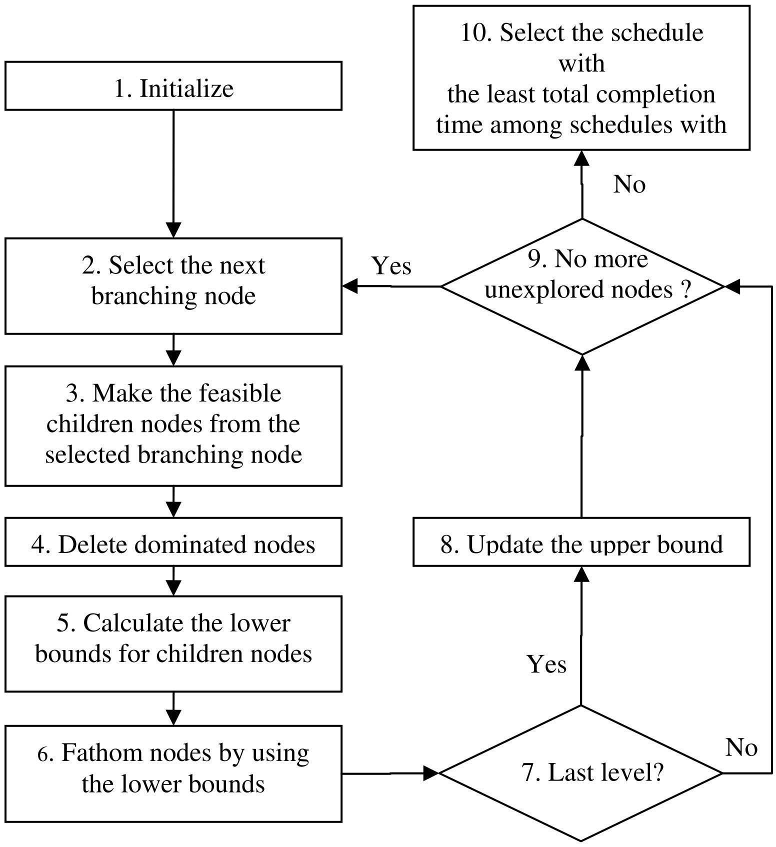

Figure 9: The overall procedure of the B & B method (Lai and Lam, 1994).

Table 3: Solving small size problems using GA and lingo.

Problem size	Value (Objective function) min		Completion time for slots using GA min								
$K \times S$	Lingo (B & B)	GA	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9
3×2	70	70	30	70	20						
4×2	82	82	30	80	20	82					
4×3	62	62	30	70	20	62					
5×2	106	106	50	100	20	106	44				
5×3	82	82	30	80	20	82	44				
6×2	Not Available	142	30	80	20	142	64	114			
6×3	Not Available	94	30	80	20	82	44	94			
7×2	Not Available	162	50	100	20	162	82	132	38		
7×3	Not Available	106	50	100	20	106	44	88	38		
8×2	Not Available	162	50	100	20	162	108	158	104	26	
8×3	Not Available	126	30	80	20	126	64	114	64	26	
9×2	Not Available	180	30	118	20	180	118	168	68	46	74
9×3	Not Available	132	50	100	20	132	70	116	66	26	28

Table 4: Solving large size problems using GA.

Problem size	10×2	10×3	15×2	15×3	20×2	20×3
Value (Objective function) min	212	142	308	214	442	312
Slot number	C_s	C_s	C_s	C_s	C_s	C_s
1	50	30	98	30	112	50
2	144	80	192	150	442	166
3	20	46	42	50	20	20
4	206	142	304	212	392	272
5	94	140	142	144	410	210
6	142	96	242	100	200	216
7	92	66	178	64	150	88
8	26	26	68	26	46	50
9	54	28	58	50	110	116
10	212	136	248	214	330	286
11			94	100	82	64
12			22	22	22	22
13			140	126	158	212
14			308	186	260	312
15			30	80	80	94
16					198	252
17					50	28
18					366	166
19					294	128
20					230	160

Table 5: GA results for $Y_{s,s'}$ s (Decision variables).

s \ s'	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1	0	1	1	1	1	0	1	1	0	0	1	1	1	1	0	1	1	1
2	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0
3	1	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0
6	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
7	0	1	0	1	1	1	0	0	1	1	0	0	1	1	0	1	0	1	0	1
8	0	1	0	1	1	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1
9	0	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	1
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	0	1	1	1	0	0	1	1	0	0	1	1	1	1	0	1	1	1
12	0	1	0	1	1	1	1	1	1	1	1	0	1	1	1	1	0	1	1	1
13	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	1	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	1	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	1	0	1
16	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
17	0	1	0	1	1	1	1	0	1	1	1	0	1	1	1	1	0	1	1	1
18	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0
19	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	1
20	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0	1	0	0	0	0

Table 6: GA results for $X_{s,k}$ s (Decision variables).

s	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
K																				
1	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	1	1	0	1	1
2	0	0	0	0	0	1	0	1	0	1	0	1	0	0	1	0	0	1	0	0
3	1	1	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0

Table 7: Final crane allocation in the yard slots for problem size 20 × 3.

Value (Objective function) = 312				
Number of slots	Crane 1	Crane 2	Crane 3	the completion time of slot C_s (min)
1			*	50
2			*	166
3			*	20
4			*	272
5			*	210
6		*		216
7			*	88
8		*		50
9			*	116
10		*		286
11	*			64
12		*		22
13	*			212
14	*			312
15		*		94
16	*			252
17	*			28
18		*		166
19	*			128
20	*			160

6. Conclusion

It has been widely documented that container terminals play pivotal role in the world. The efficiency of container terminals depends on the resource allocation in the terminal. Yard cranes are the most expensive equipment used in the storage yard. This paper described a mathematical formulation for the yard crane scheduling problem as an important problem in the operation of port container terminals. In this study, the problem of scheduling multiple yard cranes to handle containers within a yard has been studied. A mixed integer program with non-interference constrain between cranes that usually move on the same rail in the yard has been proposed. Since, it has been shown that the scheduling problem is an NP-HARD problem; a heuristic method based on GA has been proposed to find optimal schedules for the problem. Numerical comparisons have shown the accuracy and efficiency of proposed GA in solving the YCSNIP.

In this paper, the factors such as the travel time of a yard crane between two slots, has not been considered. The incorporation of these factors into the YCSNIP can be a topic for future research.

References

Cheung, R. K.; Li, C. L.; Lin, W., (2002), Inter block crane deployment in container terminals. *Transportation Science*, 36(1), 79-93.

Crainic, T. G, Kim, K. H., (2005), Intermodal transportation.

Gen, M.; Cheng, R., (1996), Genetic *algorithms and engineering design*. John Wiley & Sons, New York.

Kim, J. U.; Kim, Y. D., (1996), Simulated annealing and genetic algorithms for scheduling products with multi-level product structure. *Computers and Operations Research,* 23(9), 857-868.

Kim, K. H.; Lee, K. M.; Hwang, H., (2003), Sequencing delivery and receiving operations for yard cranes in port container terminals. *International Journal of Production Economics,* 84(3), 283-292.

Kim, K. H.; Park, Y. M., (2004), A crane scheduling method for port container terminals. *European Journal of Operation Research*, 156(3), 752-768.

Lai, K. K.; Lam, K., (1994), A study of container yard equipment allocation strategy in Hong Kong. *International Journal of Modeling & Simulation*, 14(3), 134-138.

Lai, K. K.; Leung, J. W., (1996), *Analysis of yard crane deployment strategies in a container terminal.* Proceedings of ICC & IE Conference, 1187-1190.

Lee, D. H.; Wang, H. Q.; Miao, L., (2008), Quay crane scheduling with non-interference constraints in port container terminals. *Transportation Research Part E*, 44(1),124-135.

Murty, K. G., (2007), Yard crane pools and optimum layouts for storage yards of container terminals. *Journal of Industrial and Systems Engineering*, 3(--), 190-199

Ng, W. C.; Mak, K. L., (2005), Yard crane scheduling in port container terminals. *Applied Mathematical Modelling*, 29(3), 263-276.

Ng, W. C.; Mak, K. L.; Zhang, Y. X., (2007), Scheduling trucks in container terminals using a genetic algorithm. *Engineering Optimization*, 39(1), 33-47.

Zhang, C.; Liu, J.; Wan, Y. W.; Murty, K. G.; Linn, R. J., (2003), Storage space allocation in container terminals. *Transportation Research Part B,* 37(10), 883-903.

Zhang, C.; Wan, Y.W.; Liu, J.; Linn, R. J., (2002), Dynamic crane deployment in container storage yards. *Transportation Research Part B: Methodology*, 36B(6), 537-555.