TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 34/67 pp211-216
Volume 13, Number S1, October 2008

A Decision Support Method for Truck Scheduling and Storage Allocation Problem at Container

CAO Jinxin (曹瑾鑫)**, SHI Qixin (史其信), Der-Horng Lee†

Department of Civil Engineering, Tsinghua University, Beijing 100084, China;
† Department of Civil Engineering, National University of Singapore, Singapore 117576, Singapore

Abstract: Truck scheduling and storage allocation, as two separate subproblems in port operations, have been deeply studied in past decades. However, from the operational point of view, they are highly interdependent. Storage allocation for import containers has to balance the travel time and queuing time of each container in yard. This paper proposed an integer programming model handling these two problems as a whole. The objective of this model is to reduce congestion and waiting time of container trucks in the terminal so as to decrease the makespan of discharging containers. Due to the inherent complexity of the problem, a genetic algorithm and a greedy heuristic algorithm are designed to attain near optimal solutions. It shows that the heuristic algorithm can achieve the optimal solution for small-scale problems. The solutions of small- and large-scale problems obtained from the heuristic algorithm are better than those from the genetic algorithm.

Key words: container terminal; vehicle scheduling; genetic algorithms; heuristic algorithm

Introduction

Seaport container terminals play an important role in modern logistics as a hub of maritime transportation. The competitiveness of a terminal is primarily reflected by its efficiency of transshipment because of that the charges paid by a ship depend on the turnaround time and the number of containers loaded and unloaded at a terminal. The number of containers corresponding to the terminal is specified by the ship company and is fixed, thus the turnaround time becomes a critical factor affecting the costs of ship companies for transshipments. As can be seen in Fig. 1, container terminal is generally operated with three types of equipments, namely quay cranes, yard trucks as well as yard cranes.

Received: 2008-05-30
** To whom correspondence should be addressed.
E-mail: cjx@mails.tsinghua.edu.cn; Tel: 86-10-62792260

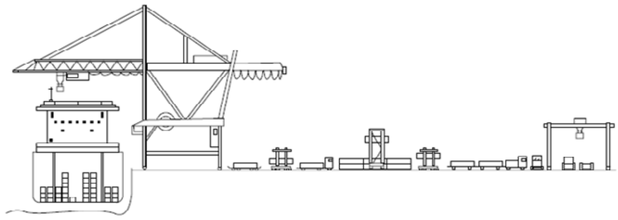

Fig. 1 Container Terminal System[1]

Operational problems in container terminals have been divided into several subproblems, such as berth allocation[2], stowage planning[3], quay crane scheduling[4], yard truck scheduling, yard crane scheduling[5], and storage allocation[6]. Stahlbock and Voß[7] updated recent work published after the comprehensive literature review presented by Steenken et al.[1].

Bish[8-10] focused on the truck scheduling and storage allocation problem but did not consider the travel time and waiting time of trucks explicitly. In practice, during peak hours, trucks need to wait for an idle yard crane to unload the containers from them, after which they can proceed to new jobs. Note that the resulting waiting time can be much longer than the truck travel time.

Murty et al.[11] considered scheduling container trucks and allocating storage space to arriving containers. The number of waiting trucks at yard blocks is monitored and storage location for discharging containers is allocated to storage places which have fewer waiting trucks.

Ng et al.[12] addressed the problem of scheduling a fleet of trucks to perform a set of transportation jobs with sequence-dependent processing time and different ready time. However, the waiting time at yard blocks were not taken into account.

In this paper, an integer programming model is built for container truck scheduling and storage allocation of discharging containers. The integer programming model is proposed to formulate in the next section. The waiting time at quay side is considered through different ready time of containers transported by trucks while the waiting time at yard side are formulated explicitly in the model. The objective of this paper is to minimize the makespan of discharging import containers. Due to the NP-hardness of the problem, a genetic algorithm and a greedy heuristic are proposed in respective sections. Small- and large-scale instances are adopted to examine the performance of the proposed solution algorithms.

1 Model Formulation

In general, once a vessel has anchored at the assigned berth, outgoing containers will be discharged before incoming containers can be loaded. A number of quay cranes will need to be assigned for such unloading and loading. Meanwhile, a fleet of trucks will be engaged to transport containers between the vessel and storage yard blocks. Yard blocks can be partitioned on the basis of vessel to temporarily store the discharged containers. In this research, only discharging process is considered while the loading process can be modeled in a similar way.

Let n denote the number of containers to be discharged from the ship, m be the number of yard blocks partitioned for this ship as storage space, and s be the number of trucks assigned to finish the discharging jobs for this ship. Discharging plan for each quay crane is predetermined by the terminal operator, so that each container i ($i = 1, 2, \ldots, n$) has their own ready time, r_i, for transportation which is defined as the moment at which the container is ready to be picked up by a truck. The travel time from the ship to yard block l ($l = 1, 2, \ldots, m$) is λ_l. And the processing time for unloading a container from the truck at all yard blocks is identical and equal to δ. Container i ($i = 1, 2, \ldots, n$) may have to wait for the assigned truck for its shipment, so the actual transportation begins at t_i: if the assigned truck is idle at r_i, $t_i = r_i$; otherwise, the container needs to wait until the assigned truck becomes idle. Every container is assigned to a storage yard block, and let d_i denote the travel time of container i ($i = 1, 2, \ldots, n$). A truck picks up container i at t_i and arrived at its storage yard block at $(t_i + d_i)$. Let $\tilde{t}_i$ be the moment at which container i is unloaded from the truck: if at $(t_i + d_i)$ one or more yard cranes at this yard block are idle, $\tilde{t}_i = t_i + d_i$; otherwise, the truck with container needs to wait until there is an idle yard crane. If the container is unloaded at $\tilde{t}_i$, the truck transporting it will return to the ship and be ready for picking up a new container at $(\tilde{t}_i + d_i)$, and the yard block will have one or more idle yard cranes after time $(\tilde{t}_i + \delta)$. Let C_i be the completion time of discharging container i, i.e., $C_i = \tilde{t}_i + \delta$.

The following notations are also used in the model formulation.

U_{ik} = 1, if truck k ($k = 1, 2, \ldots, s$) is assigned to transport container i ($i = 1, 2, \ldots, n$);
= 0, otherwise.

V_{il} = 1, if container i ($i = 1, 2, \ldots, n$) is allocated to yard block l ($l = 1, 2, \ldots, m$);
= 0, otherwise.

X_{ijk} = 1, if truck k ($k = 1, 2, \ldots, s$) processes container j after container i ($i, j = 1, 2, \ldots, n$, $i \neq j$);
= 0, otherwise.

Y_{ijl} = 1, if container i ($i = 1, 2, \ldots, n$) is stacked in yard block l ($l = 1, 2, \ldots, m$) immediately before container j ($i, j = 1, 2, \ldots, n$, $i \neq j$);
= 0, otherwise.

K = A large positive number.

The truck scheduling and storage allocation problem for discharging containers can be formulated as follows:

$$\text{Min} \quad \max_{i=1,2,\ldots,n} C_i \tag{1}$$

$$\text{s.t.} \quad C_i = \tilde{t}_i + \delta, \ i = 1, 2, \ldots, n \tag{2}$$

$$d_i = \sum_{l=1}^{m} V_{il}\lambda_l, \ i = 1, 2, \ldots, n \tag{3}$$

$$t_i \geqslant r_i, \ i = 1, 2, \ldots, n \tag{4}$$

$$\tilde{t}_i \geqslant t_i + d_i, \ i = 1, 2, \ldots, n \tag{5}$$

$$t_i + d_i \leqslant K \cdot (1 - Y_{ijl}) + t_j + d_j, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ l = 1, 2, \ldots, m \tag{6}$$

$$\tilde{t}_i + \delta \leqslant K \cdot (1 - Y_{ijl}) + \tilde{t}_j, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ l = 1, 2, \ldots, m \tag{7}$$

$$\tilde{t}_i - d_i \leqslant K \cdot (1 - X_{ijk}) + t_j, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ k = 1, 2, \ldots, s \tag{8}$$

$$\sum_{k=1}^{s} U_{ik} = 1, \ i = 1, 2, \ldots, n \tag{9}$$

$$\sum_{l=1}^{m} V_{il} = 1, \ i = 1, 2, \ldots, n \tag{10}$$

$$X_{ijk} \leqslant U_{ik}, \ i, j = 1, 2, \ldots, n, \ i \neq j, \quad k = 1, 2, \ldots, s \tag{11}$$

$$X_{ijk} \leqslant U_{jk}, \ i, j = 1, 2, \ldots, n, \ i \neq j, \quad k = 1, 2, \ldots, s \tag{12}$$

$$X_{ijk} + X_{jik} \leqslant 1, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ k = 1, 2, \ldots, s \tag{13}$$

$$X_{ijk} + X_{jik} \geqslant U_{ik} + U_{jk} - 1, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ k = 1, 2, \ldots, s \tag{14}$$

$$Y_{ijl} \leqslant V_{il}, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ l = 1, 2, \ldots, m \tag{15}$$

$$Y_{ijl} \leqslant V_{jl}, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ l = 1, 2, \ldots, m \tag{16}$$

$$Y_{ijl} + Y_{jil} \leqslant 1, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ l = 1, 2, \ldots, m \tag{17}$$

$$Y_{ijl} + Y_{jil} \geqslant V_{il} + V_{jl} - 1, \quad i, j = 1, 2, \ldots, n, \ i \neq j, \ l = 1, 2, \ldots, m \tag{18}$$

$$U_{ik}, \ V_{il}, \ X_{ijk}, \ Y_{ijl} \in \{0, 1\}, \quad i, j = 1, 2, \ldots, n, \ l = 1, 2, \ldots, m, \quad k = 1, 2, \ldots, s \tag{19}$$

$$C_i, \ d_i, \ t_i, \ \tilde{t}_i \in \mathbf{Z}^+, \ i = 1, 2, \ldots, n \tag{20}$$

Equation (1) represents the objective function of the truck scheduling and storage allocation problem that is to minimize the makespan of the n discharging containers. Constraints (2) represent the calculation of completion time of each job. Constraints (3) represent the computation of travel time of each container. The travel time of a container is equal to the travel time from the ship to the allocated storage yard block of the container. Constraints (4) ensure that containers can be transported only after their ready time, and constraints (5) ensure that containers can be unloaded by yard cranes only after they arrive at yard blocks. If container i arrives at yard block l before container j, constraints (6) force $Y_{ijl} = 1$, and constraints (7) give the first-come-first-serve rule at yard blocks. Constraints (8) state that a truck can transport a new container until it finishes its preceding containers and returns to the ship. Constraints (9) and (10) ensure that each container is allocated to only one yard block and transported by only one truck. Constraints (11) give that $U_{ik} = 1$ if truck k processes a container after container i, and constraints (12) give that $U_{jk} = 1$ if truck k processes a container before container i. Constraints (13) and (14) ensure that truck k processes either container i before container j or container j before container i, if both job i and job j are assigned to truck k, i.e., $U_{ik} + U_{jk} = 2$. Constraints (15) give that $V_{il} = 1$ if yard block l processes a container after container i, and constraints (16) give that $V_{jl=1}$ if yard block l processes a container before container i. Constraints (17) and (18) ensure that yard block l processes either container i before container j or container j before container i, if both container i and container j are allocated to yard block l, i.e., $V_{il} + V_{jl} = 2$. Constraints (19) and (20) give the range of all decision variables.

It is not likely to get the optimal solution for real-scale problems in real life with the spending of reasonable computational time and cost; therefore, a heuristic algorithm is designed in the next section for problem solution.

2 Genetic Algorithm

The genetic algorithm (GA), developed by Holland[13], is one of the nature-inspired meta-heuristics used for handling combinatorial problems. The usual form of GA has been reported by Goldberg[14]. GA is a stochastic search technique based on the mechanism of natural selection and natural genetics. GA starts with a set of random solutions called population. Each individual, named chromosome is represented by a string. The chromosomes evolve through successive iterations, called generations. When generating offspring genetic operations, crossover, and mutation, are adopted on random selected chromosomes. And new generation is selected based on Darwinian evolution by evaluating the fitness. Individuals with better performances will have more probability to be chosen. GA has been refined by numerous researchers, and it is one of the popular meta-heuristic algorithms for solving facility scheduling and operation problems in container terminals[12].

2.1 Representation

In this paper, a chromosome includes two parts: the first $(n+s-1)$ holds represent the truck schedule and the following $(n+m-1)$ holds represent the storage allocation. The positive holds represent the containers, and without losing of generality, the index of containers are ranked by the ready time of containers, i.e., $r_i \leqslant r_{i+1}$ ($i=1,2,\ldots,n-1$). The negative holds in the chromosome represent $(s-1)$ trucks and $(m-1)$ yard blocks. Containers in the chromosome before a negative hold are assigned to the corresponding truck/yard block. Other containers without any negative hold followed are assigned to truck s or yard block m.

2.2 Initialization

The genetic algorithm starts with a group of initial individuals as the first generation. To simplify the generation of initial individuals, a set of randomly generated schedule and allocation plan is used as the first generation. In the initialization, two parts of a chromosome are generated independently. In the first part, positive number $1,2,\ldots,n$, and negative number -1, $-2,\ldots,-(s-1)$ are randomly permuted while in the second part, positive number $1,2,\ldots,n$, and negative number $-1,-2,\ldots,-(m-1)$ are randomly permuted.

2.3 Fitness evaluation and selection

After given a feasible solution represented by a chromosome, the completion time of each job can be calculated, and the corresponding makespan for this solution can be determined as the maximum completion time of all containers.

By ranking the containers assigned for each truck in increasing order, the schedule of each truck can be achieved. The second part of the chromosome gives the storage allocation of each container, and the travel time of each container can be calculated. The minimum makespan can then be calculated. The evaluation of this chromosome is given by

$$\text{eval}=\frac{1}{\max\limits_{i=1,2,\ldots,n} C_i} \tag{21}$$

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[15].

2.4 Crossover

Generally, the above mentioned chromosome representation will yield illegal offspring by one-point, two-point or multi-point crossover in the sense of that some holds may be missed while some holds may be duplicated in the offspring. Therefore, this paper adopts 'order crossover'[15] for one of the two parts of the chromosome, in which repairing procedure is embedded to resolve the illegitimacy of offspring, as is shown in Fig. 2.

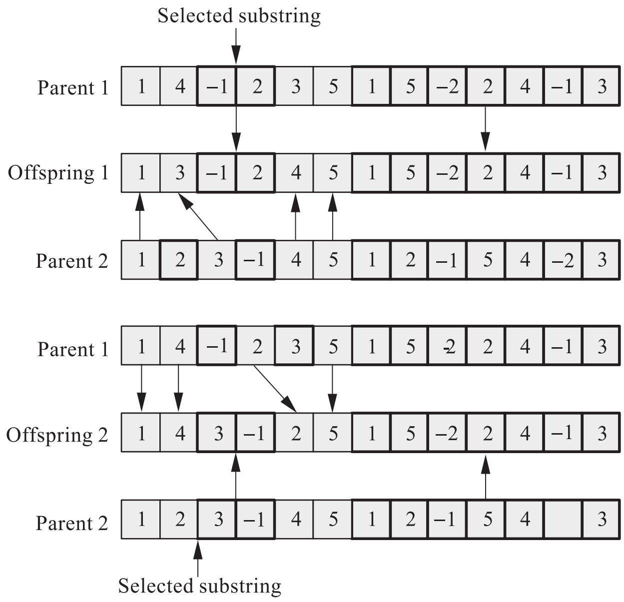

Fig. 2 An illustration of the ordered crossover

2.5 Mutation

Mutation forces the GA searching new areas. It also helps the GA avoid premature convergence and find the global optimal solution. In general, in the mutation all individuals in the population are checked bit by bit and the bit values are randomly reversed according to a pre-specified rate. However, in this paper the mutation selects chromosomes randomly in terms of the probability of mutation and chooses two positions in the same part of the selected chromosome at random then swaps the holds on these positions.

3 Proposed Heuristic Algorithm

GA is a meta-heuristic method in dealing with general combinatorial optimization problems. However, the performance of GA is dependent on the problem structure, choosing of population size, chromosome coding, and choosing of genetic operators. Heuristic algorithm is problem specialized, which can be used to solve specific problems with some simple rules. In this section, a greedy heuristic algorithm is designed for solving the truck scheduling and storage allocation problem.

The basic concept of this greedy heuristic algorithm is to assign a container to the truck that can transport it earliest and allocate storage yard block to the container which can unloaded the container earliest. Containers are assigned to trucks and allocated to yard blocks in the order of ready time. This algorithm can be approach the makespan of n containers by the following steps in n iterations:

Step 1: Set the idle time of trucks $\mathrm{t\_idle}_k = 0$ $(k = 1, 2, \ldots, s)$, the idle time of yard blocks $\mathrm{y\_idle}_l = 0$ $(l = 1, 2, \ldots, m)$, and $i = 1$;

Step 2: Compare the idle times of all trucks, and assign the one with minimum idle time, denoted $k^{\min}$, to transport container i. Update $t_i = \max\{\mathrm{t\_idle}_{k^{\min}}, r_i\}$;

Step 3: Compare the $\max\{t_i + \lambda_l, \mathrm{y\_idle}_l\}$ for all yard blocks, and denote $l^{\min}$ as the yard block with the minimum value. Allocate container i to yard block $l^{\min}$. Update $\tilde{t}_i = \max\{t_i + \lambda_{l^{\min}}, \mathrm{y\_idle}_{l^{\min}}\}$, $\mathrm{t\_idle}_{k^{\min}} = \tilde{t}_i + \lambda_{l^{\min}}$, $\mathrm{y\_idle}_{l^{\min}} = \tilde{t}_i + \delta$, and $C_i = \tilde{t}_i + \delta$;

Step 4: Go to **Step 5**, if $i = n$; otherwise let $i = i + 1$, and go to **Step 2**;

Step 5: The makespan of n containers is equal to $\max\limits_{i=1,2,\ldots,n} C_i$.

4 Computational Experiments

In this section, small-scale examples are given to compare the performances of GA and greedy heuristic algorithm to the optimal solution computed by CPLEX. Because of the difficulty of CPLEX in handling large-scale problems, the comparisons for large-scale instances are only undertaken between GA and the proposed heuristic algorithm.

4.1 Small-scale examples

Ten small-scale randomly generated examples with 5 containers transported by 2 trucks to 3 yard blocks are adopted in the computational experiments. The ready time of containers is uniformly distributed in [0, 500], and the travel time of yard blocks are uniformly distributed in [120, 600]. The processing time at yard block is set at 99.

From Fig. 3, it is clear that for small-scale examples heuristic algorithm achieves better solution compared with GA. The average improvement of heuristic compared with GA is about 12%.

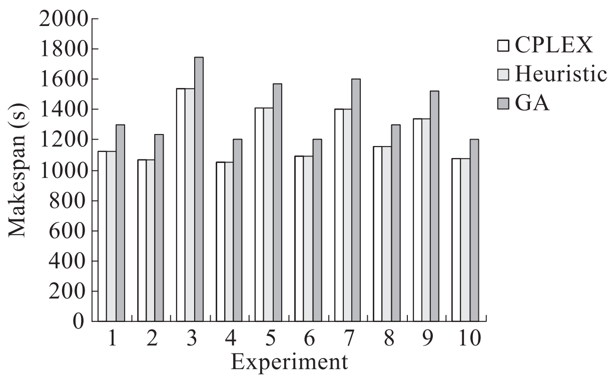

Fig. 3 Results of small-scale instances

4.2 Large-scale examples

Solutions of 10 large-scale examples obtained by GA and heuristic are compared.

Figure 4 depicts that for large-scale examples the performance of the proposed heuristic is also better than GA, and the average improvement of solution is about 7%.

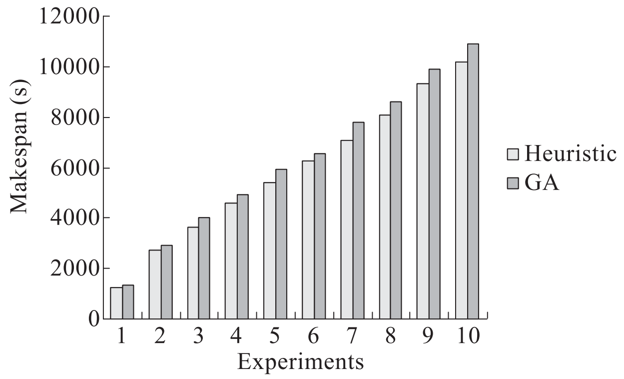

Fig. 4 Results of large-scale examples

5 Conclusions

In this paper, an integer programming model for truck scheduling and storage allocation problem is formulated. In this model, a fleet of trucks are assigned to transport discharging containers from the ship to one of the storage yard blocks. The objective of this problem is to achieve the minimum make span for all discharging containers. The processing time of trucks discharging containers consists of travel time on the network and waiting time at quayside and yard side. In this paper, quayside waiting time due to the technical performance of quay cranes and unloading sequence is considered in the different ready time of containers. The waiting time and travel time are explicitly considered in the model. Compared with previous research, the model formulated in this paper is more practical and comprehensive by considering different items of processing time of discharging containers. By balancing the travel time and waiting time, minimum makespan is achieved.

Due to the NP-hardness of the truck scheduling and storage allocation problem, which means that it is impossible to find the optimal solution, a genetic algorithm and a greedy heuristic algorithm are designed to handle this problem. The greedy heuristic algorithm can obtain the same objective value for small-scale instance as those from CPLEX. Results of both small- and large-scale examples show that the proposed heuristic algorithm does perform better than the genetic algorithms for solving this problem.

References

[1] Steenken D, Voß S, Stahlbock R. Container terminal operation and operations research — A classification and literature review. *OR Spectrum*, 2004, **26**: 282-292.

[2] Imai A, Nishimura E, Papadimitriou S. The dynamic berth allocation problem for a container port. *Transportation Research Part B*, 2001, **35**(4): 401-417.

[3] Imai A, Sasaki K, Nishimura E, Papadimitriou S. Multi-objective simultaneous stowage and load planning for a container ship with container rehandle in yard stacks. *European Journal of Operational Research*, 2006, **171**(2): 373-389.

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

[5] Lee D H, Cao Z, Meng Q. Scheduling of two-transtainer systems for loading outbound containers in port container terminals with simulated annealing algorithm. *International Journal of Production Economics*, 2007, **107**(1): 115-124.

[6] Lee L H, Chew E P, Tan K C, Han Y. An optimization model for storage yard management in transshipment hubs. *OR Spectrum*, 2006, **28**(4): 539-561.

[7] Stahlbock R, Voß S. Operations research at container terminals: A literature update. *OR Spectrum*, 2008, **30**(1): 1-52.

[8] Bish E K, Leong T Y, Li C L, Ng J W C, Simchi-Levi D. Analysis of a new vehicle scheduling and location problem. *Naval Research Logistics*, 2001, **48**(5): 363-385.

[9] Bish E K. A multiple-crane-constrained scheduling problem in a container terminal. *European Journal of Operational Research*, 2003, **144**(1): 83-107.

[10] Bish E K, Chen F Y, Leong Y T, Nelson B L, Ng J W C, Simchi-Levi D. Dispatching vehicles in a mega container terminal. *OR Spectrum*, 2005, **27**(4): 491-506.

[11] Murty K G, Wan Y, Liu J, Tseng M M, Leung E, Lai K K, Chiu H W C. Hongkong international terminals gains elastic capacity using a data-intensive decision-support system. *Interfaces*, 2005, **35**(1): 61-75.

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

[13] Holland J H. Adaptation in Natural and Artificial Systems. Ann Arbor, USA: University of Michigan Press, 1975.

[14] Goldberg D. Genetic Algorithms in Search, Optimization and Machine Leaning. Cambridge, MA: Addison-Wesley, 1989.

[15] Gen M, Cheng R. Genetic Algorithms and Engineering Design. Singapore: John Wiley, 1996.