Research Article

Improved Harmony Search Algorithm for Truck Scheduling Problem in Multiple-Door Cross-Docking Systems

Zhanzhong Wang , Yue Lu , Liying Zhao , and Ningbo Cao

College of Transportation, Jilin University, No. 5988 Renmin Street, Changchun, Jilin, China

Correspondence should be addressed to Liying Zhao; 791950198@qq.com

Received 8 December 2017; Revised 22 February 2018; Accepted 7 March 2018; Published 24 April 2018

Academic Editor: Youssef N. Raffoul

Copyright © 2018 Zhanzhong Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The key of realizing the cross docking is to design the joint of inbound trucks and outbound trucks, so a proper sequence of trucks will make the cross-docking system much more efficient and need less makespan. A cross-docking system is proposed with multiple receiving and shipping dock doors. The objective is to find the best door assignments and the sequences of trucks in the principle of products distribution to minimize the total makespan of cross docking. To solve the problem that is regarded as a mixed integer linear programming (MILP) model, three metaheuristics, namely, harmony search (HS), improved harmony search (IHS), and genetic algorithm (GA), are proposed. Furthermore, the fixed parameters are optimized by Taguchi experiments to improve the accuracy of solutions further. Finally, several numerical examples are put forward to evaluate the performances of proposed algorithms.

1. Introduction

Cross docking is a process of logistics, of which the products are unloaded from the inbound trucks and then sorted and loaded into outbound trucks without warehousing. In a traditional distribution center, there are four major functions of warehousing that can be distinguished: receiving, storage, order picking, and shipping. Since the major costs of warehousing are operations of storage and order picking, and the cross docking can successfully eliminate them to considerably decrease distribution costs, cross docking is nowadays used by many companies, such as Wal Mart, Goodyear GB Ltd., Toyota, Eastman Kodak Co., and Dots and LLC [1].

There are several problems that researchers are passionate about, such as layout design, location, network, and vehicle routing problems. For example, Bartholdi III and Gue [2] introduced an efficient design of the cross-docking layout, whose objective was to minimize the transfer time, material handling, and congestion. Dondo and Cerdá [3] presented a new monolithic MILP formulation which integrated the pickup/delivery vehicle routing and scheduling with the assignment of dock doors. Nevertheless, two of the most

important factors that affect the efficiency of cross docking are door assignment and truck scheduling. A proper sequence of trucks can significantly decrease the makespan and improve the efficiency. The problems of door assignment and truck scheduling are always considered together to optimize the procedure of the cross docking.

The layout of the cross-docking terminal is in many forms, such as the number of dock doors and whether there is a temporary storage. Yu and Egbelub [4] proposed a truck scheduling model with one inbound door and one outbound door. Boloori Arabani et al. [5] put forward metaheuristic algorithms to solve the same problem. Forouharfard and Zandieh [6] proposed an integer programming (IP) model to solve the problem of single receiving and shipping door. Besides, a genetic algorithm was applied to solve the model. In addition, Yu [7] dealt with truck scheduling problem with multiple dock doors to find the best sequence for trucks as well as assignments for trucks to doors and at the same time determine the unloading sequences, product routing, and routing sequences. Madani-Isfahani et al. [8] considered a truck scheduling model with multiple doors and then designed two metaheuristics, namely, simulated

annealing and firefly algorithm to solve the offered model. Assadi and Bagheri [9] considered a cross-docking terminal with multiple doors and temporary storage, assuming the ready time of inbound trucks were uncertain. Assadi and Bagheri [10] proposed two algorithms, namely, differential evolution and population-based simulated annealing to solve the multiple-door cross-docking system.

The objective functions of truck scheduling are about time, number of products transferred to temporary storage, customer satisfaction, and so on. The makespan of cross docking as well as the total earliness and total tardiness of trucks is discussed more often. Wisittipanich and Hengmeechai [11] and Boloori Arabani et al. [12] proposed a truck scheduling model with objective of minimizing the total earliness and tardiness simultaneously. Lee et al. [13] established a mathematical model to maximize the number of products shipped in a given working horizon. Ladier and Alpan [14] proposed an objective of minimizing both total number of pallets put in storage and the dissatisfaction of transportation providers to evaluate the quality of cross-docking system.

The operation forms of cross docking may be different, the times of a truck moves in and out can be once or more. Vahdani et al. [15] presented a truck scheduling model with no temporary storage in the cross-docking terminal. Both inbound and outbound trucks can move in and out of the docks during their tasks repetitively. Two known metaheuristics named genetic algorithm and electromagnetism-like algorithm were applied to minimize the total flow time. Mohtashami [16] dealt with a model that the outbound trucks can repetitively move; however the inbound trucks can move in only once.

Truck scheduling problems could be divided into two categories, which are the scheduling of both inbound and outbound trucks as well as the scheduling of inbound trucks only. Boysen et al. [17] dealt with scheduling of inbound trucks only, assuming outbound trucks left at fixed sequence and then proposed decomposition procedures and simulated annealing to minimize total lost profit. Rahmanzadeh Tootkaleh et al. [18] proposed an inbound truck scheduling model based on fixed outbound trucks' departure times.

This paper proposes a truck scheduling model at a cross-docking system with multiple inbound and outbound doors to minimize the total makespan. The inbound and outbound trucks are all available at time zero, so it is necessary to schedule all the trucks and make proper door assignments as well as truck sequences. Furthermore, it is assumed that outbound trucks load products from temporary storages first and then from inbound trucks, in case that the temporary storages get filled up. An inbound truck enters the inbound platform when the predecessor leaves and leaves the inbound platform when none of the outbound trucks at the outbound platforms need its products. It is worth noting that its leaving time is no less than the start time plus the unloading time due to the waiting time existing. In addition, three metaheuristics are proposed, namely, HS, IHS, and GA, to solve the model in order to find the best sequence.

The remainder of this paper is organized as follows: The next section describes the assumptions, mixed integer

programming model for the truck scheduling problem, and the process of the cross docking. Section 3 presents three metaheuristics: harmony search, improved harmony search, and genetic algorithm. In Section 4, the performances of the metaheuristic algorithms are evaluated over the tests about truck scheduling. Finally, conclusions are drawn and future work is shortly expressed.

2. Problem Description

The studied problem in this paper is a mathematical model based on mixed integer programming working on trucks scheduling problems in a cross-docking system with multiple doors. Figure 1 shows the layout of the cross-docking terminal; in the terminal, temporary storages are designed besides the inbound doors, which are extremely large but limited. The outbound trucks load products from temporary storages first and then from inbound trucks in the inbound doors to avoid filling up the temporary storages. The objective function is minimizing the total makespan in order to reduce the waiting time and improve the efficiency of cross-docking system.

2.1. Assumptions

- (1) There are multiple receiving doors and shipping doors in cross-docking system.
- (2) The shipping time is related to the distance.
- (3) Every outbound truck leaves until satisfying its needs.
- (4) Every inbound truck leaves until unloading its products.
- (5) Every truck is available at time zero.
- (6) The quantities and categories the inbound trucks load are equal to the quantities and categories the outbound trucks need.
- (7) There is no preemption of any truck.
- (8) The capacity of each truck is different, each inbound truck loads specific types of products, and outbound truck needs specific types of products.
- (9) There is a temporary storage next to each receiving door which is with a large volume but limited.
- (10) The changeover time of trucks is equal.
- (11) The time of loading and unloading is in direct proportion to the quantity of products.
- (12) The processes of sorting and labeling are not considered.
- (13) The outbound truck loads products from inbound trucks, when there are no more products that the outbound truck needs in the temporary storages.
- (14) The order of loading and unloading of products is not considered.
- (15) The products can be exchanged; that is, the products on a certain inbound truck can be loaded into any outbound trucks, which need them.
- (16) There is only one chance for each inbound and outbound truck to move in. The inbound truck unloads the rest of the products to the temporary storage and leaves when the inbound truck on the inbound platform cannot fill any outbound trucks on the outbound platforms. An outbound truck leaves when it fills all the products needed.

FIGURE 1: The layout of cross-docking terminal.

2.2. Notations

R : number of inbound trucks ($i = 1, 2, \dots, I$).
 S : number of outbound trucks ($j = 1, 2, \dots, O$).
 P : number of product types.
 k : index for product type ($k = 1, 2, \dots, P$).
 I : number of doors at receiving door ($m = 1, 2, \dots, R$).
 O : number of doors at shipping door ($n = 1, 2, \dots, S$).
 DT : changeover time of trucks.
 M : a positive big number.
 T : loading or unloading time per product unit.
 r_{ik} : number of units of product type k that is initially loaded in inbound truck i .
 s_{jk} : number of units of product type k that is initially needed for outbound truck j .
 t_{mn} : transfer time per unit of product type k from inbound door m to outbound door n .
 c_i : time at which inbound truck i enters the receiving door.
 C_i : time at which inbound truck i leaves the receiving door.
 l_j : time at which outbound truck j enters the shipping door.
 L_j : time at which outbound truck j leaves the shipping door.
 $L_{previous,j}$: time at which the predecessor of outbound truck j leaves the outbound door.
 x_{ijk} : number of units of product type k transferred from inbound truck i to outbound truck j .
 LX_{ijk} : number of units of product type k transferred from temporary storage to outbound truck j , which

the products have been transferred from inbound truck i to temporary storage.

C_{ij} : time at which outbound truck j starts to load products from inbound truck i .

$C_{ij_{last}}$: time at which the last outbound truck j starts to load products from inbound truck i .

D_j : time at which outbound truck j starts to load products from temporary storage.

v_{ij} : if any products are transferred from inbound truck i to outbound truck j directly: 1, otherwise: 0.

y_{im} : if inbound truck i is assigned to receiving door m : 1, otherwise: 0.

z_{jn} : if outbound truck j is assigned to shipping door n : 1, otherwise: 0.

p_{ij} : if inbound trucks i and j are assigned to the same receiving door and i is a predecessor of truck j : 1, otherwise: 0.

q_{ij} : if outbound trucks i and j are assigned to the same shipping door and i is a predecessor of truck j : 1, otherwise: 0.

2.3. The Mathematical Model. The mathematical model for the problem is given as follows:

$$\min T \quad (1)$$

$$\text{S.t. } T \geq L_j, \quad j = 1, 2, \dots, S \quad (2)$$

$$\sum_{m=1}^I y_{im} = 1, \quad i = 1, 2, \dots, R \quad (3)$$

$$\sum_{n=1}^O z_{jn} = 1, \quad j = 1, 2, \dots, S \quad (4)$$

$$\sum_{j=1}^S x_{ijk} = r_{ik}, \quad i = 1, 2, \dots, R, \quad k = 1, 2, \dots, P \quad (5)$$

$$\sum_{i=1}^R x_{ijk} = s_{jk}, \quad i = 1, 2, \dots, S, \quad k = 1, 2, \dots, P \quad (6)$$

$$p_{ij} + p_{ji} \leq 1, \quad i \neq j, \quad i, j = 1, 2, \dots, R \quad (7)$$

$$q_{ij} + q_{ji} \leq 1, \quad i \neq j, \quad i, j = 1, 2, \dots, S \quad (8)$$

$$y_{im} + y_{jm} - 1 \leq p_{ij} + p_{ji}, \quad i, j = 1, 2, \dots, R, \quad i \neq j, \quad m = 1, 2, \dots, I \quad (9)$$

$$z_{in} + z_{jn} - 1 \leq q_{ij} + q_{ji}, \quad i, j = 1, 2, \dots, S, \quad i \neq j, \quad n = 1, 2, \dots, O \quad (10)$$

$$C_i + DT - M * (1 - p_{ij}) \leq c_j, \quad i, j = 1, 2, \dots, R \quad (11)$$

$$C_i \geq C_{ij_{\text{last}}} + \sum_{k=1}^P x_{ijk} - M * \sum_{j=1}^S v_{ij}, \quad (12)$$

$$i = 1, 2, \dots, R, \quad j = 1, 2, \dots, S, \quad k = 1, 2, \dots, P$$

$$C_i \geq C_l + DT + \sum_{k=1}^P r_{ik} - M * (1 - p_{li}) - M * \sum_{j=1}^S v_{ij}, \quad (13)$$

$$i, l = 1, 2, \dots, R, \quad k = 1, 2, \dots, P, \quad j = 1, 2, \dots, S$$

$$C_{ij} \geq \max \left(c_i, \left(l_j + T \sum_{i=1}^R \sum_{k=1}^P Lx_{ijk} \right) \right) - M * (1 - V_{ij}), \quad (14)$$

$$D_j \geq L_i + DT - M * (1 - q_{ij}), \quad i, j = 1, 2, \dots, S, \quad i \neq j \quad (15)$$

$$L_i + DT - M * (1 - q_{ij}) \leq l_j, \quad i, j = 1, 2, \dots, S, \quad i \neq j \quad (16)$$

All variables ≥ 0 .

Constraint (3) guarantees that the makespan is greater than or equal to the time the last outbound truck leaves the

outbound door. Constraints (4)-(5) ensure every inbound truck or outbound truck is allowed to be assigned to one of the inbound doors or outbound doors. Constraint (6) ensures the total number of products of type k from inbound truck i to all outbound trucks is equal to the number of products of type k that unloaded from the inbound truck i . Constraint (7) ensures the total number of products of type k from all inbound trucks to outbound truck j is equal to the number of products of type k that loaded to the outbound truck j . Constraints (8)-(9) guarantee the inbound truck or outbound truck occurs only once. Constraint (10) confirms the relationship between y_{im} and p_{ij} ; similarly, the relationship between z_{jn} and q_{ij} is ensured by constraint (11). Constraint (12) ensures that the start time of the next inbound truck j is equal to the finished time of the former inbound truck i plus the delay time of truck change. Constraint (13) is the condition of the finished time of inbound truck, in the case that there is any outbound truck loading products directly from the inbound truck i . Constraint (14) is the condition of the finished time of inbound truck i , in the case that there is no outbound truck loading products directly from the inbound truck i . Constraint (15) assures the time at which outbound truck j starts to load products from inbound truck i . Constraint (16) assures the time at which outbound truck j starts to load products from temporary storages. Constraint (17) ensures that the start time of the next outbound truck j is equal to the finished time of the former outbound truck i plus the delay time of truck change.

2.4. The Process of Cross Docking. The temporary storage is extremely large but not without a limit, so the outbound trucks on the outbound docks load products from temporary storages first and then load products from inbound trucks on the inbound docks, if the outbound trucks are still not full. The inbound trucks will stay at the inbound docks if there are any outbound trucks full and leaving the outbound dock; if not, leave the docks and unload the rest of the products to the temporary storages. The process of cross docking will be finished when all outbound trucks are full, and no truck remains. The specific flow diagram of the process of cross docking is shown in Figure 2.

There are multiple inbound doors and outbound doors in cross-docking system; more than two trucks operate at the same time at the inbound and outbound docks. Formulate the unloading and loading principles in order to assure that the cross-docking system moves smoothly and efficiently. Here are two principles as follows:

(1) Which outbound truck at outbound docks is chosen by the inbound truck to transfer products to is decided by the distance between the docks. The products from one inbound truck are loaded to the outbound truck that are the nearest from the inbound truck. For the outbound truck, if the nearest inbound truck cannot satisfy it, the far inbound trucks at other inbound docks will be considered.

(2) If there are two inbound trucks constituting the products that the outbound truck needs and the two inbound trucks are equally far from the outbound truck, the outbound truck will choose one inbound truck first randomly and then the other.

FIGURE 2: The process of cross docking.

3	4	1	2	5	4	2	6	1	5	3
1	2	1	1	2	1	3	2	2	3	1

FIGURE 3: Scheme of a solution.

FIGURE 4: Sequences of trucks.

3. Solution Methods

This paper proposes three algorithms, namely, harmony search, improved harmony search, and genetic algorithm. It is necessary to design the representation scheme of the solution, which will be used in further work. The solution consists of two layers of arrays, the first layer expresses the codes of the inbound and outbound trucks, and the second layer represents the codes of the inbound and outbound docks. The length of a solution is decided by the total number

of inbound and outbound trucks. As shown in Figure 3, it is a cross-docking terminal with two inbound doors and three outbound doors; besides there are five inbound trucks and six outbound trucks in it. The left five columns represent the inbound trucks as well as the dock doors they belong to; therefore, the right six columns represent the outbound trucks. Figure 4 demonstrates unloading and loading sequences of inbound and outbound trucks. For example, inbound truck 3 is the first truck to unload in inbound door 1.

3.1. Improved Harmony Search

3.1.1. Harmony Search Algorithm. Harmony search was first proposed by Geem et al. in 2001 [19]; it is a novel intelligent algorithm, which simulates the principle of the music play. In the music performance, the musicians adjust the tones of their instruments repeatedly through the memory to achieve a beautiful harmony.

3.1.2. Initialize the Problem and Parameters. The objective function of this paper is to minimize the total makespan of the cross-docking system. The parameters in harmony search are HMS, HMCR, PAR, and BW. HMS represents the number of solution vectors in the harmony memory. HMCR expresses the harmony memory considering rate.

PAR indicates the pitch adjusting rate. BW means bandwidth for each generation. In addition, the harmony memory (HM) is an array that contains all the solution vectors.

3.1.3. Initialize the Solution Set. Generate HMS sets of solutions randomly in HM matrix; the format of the solution is the same as in Figure 3. A solution consists of two levels, the first level are the codes of trucks and the second level are the codes of the doors. The solution set HM consists of HMS sets of solutions, and the number of rows in the array is $2 * \text{HMS}$. The sum of the inbound trucks and outbound trucks is equal to the number of columns in the array. The left R columns of HM represent the codes of inbound trucks, and the right S columns of HM represent the codes of outbound trucks.

$$\text{HM} = \begin{bmatrix} T_1^1 & T_2^1 & \cdots & T_R^1 & T_{R+1}^1 & T_{R+2}^1 & \cdots & T_{R+S}^1 \\ D_1^1 & D_2^1 & \cdots & D_R^1 & D_{R+1}^1 & D_{R+2}^1 & \cdots & D_{R+S}^1 \\ T_1^2 & T_2^2 & \cdots & T_R^2 & T_{R+1}^2 & T_{R+2}^2 & \cdots & T_{R+S}^2 \\ D_1^2 & D_2^2 & \cdots & D_R^2 & D_{R+1}^2 & D_{R+2}^2 & \cdots & D_{R+S}^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ T_1^{\text{HMS}-1} & T_2^{\text{HMS}-1} & \cdots & T_R^{\text{HMS}-1} & T_{R+1}^{\text{HMS}-1} & T_{R+2}^{\text{HMS}-1} & \cdots & T_{R+S}^{\text{HMS}-1} \\ D_1^{\text{HMS}-1} & D_2^{\text{HMS}-1} & \cdots & D_R^{\text{HMS}-1} & D_{R+1}^{\text{HMS}-1} & D_{R+2}^{\text{HMS}-1} & \cdots & D_{R+S}^{\text{HMS}-1} \\ T_1^{\text{HMS}} & T_2^{\text{HMS}} & \cdots & T_R^{\text{HMS}} & T_{R+1}^{\text{HMS}} & T_{R+2}^{\text{HMS}} & \cdots & T_{R+S}^{\text{HMS}} \\ D_1^{\text{HMS}} & D_2^{\text{HMS}} & \cdots & D_R^{\text{HMS}} & D_{R+1}^{\text{HMS}} & D_{R+2}^{\text{HMS}} & \cdots & D_{R+S}^{\text{HMS}} \end{bmatrix}. \quad (17)$$

3.1.4. Generate a New Solution Randomly. A new solution is generated based on three rules: (1) memory consideration, (2) pitch adjustment, and (3) random selection. Each value of the variable of the solution is selected from the historical values stored in the HM with possibility HMCR. That is to say, in the memory consideration, the value of the first variable (T'_1) in the first level of the solution is selected randomly from values in the specified HM range ($T_1^1 - T_1^{\text{HMS}}$), and the value of the first variable (D'_1) in the second level of the solution is selected randomly from values in the HM range ($D_1^1 - D_1^{\text{HMS}}$) with possibility HMCR. Otherwise, select a value from the possible range of values with possibility $(1 - \text{HMCR})$. Values of other variables in the solution are selected in the same manner.

$$\begin{aligned} T'_i &= \begin{cases} T'_i \in \{T_1^1, T_1^2, \dots, T_1^{\text{HMS}}\} & \text{with probability HMCR} \\ T'_i \in T & \text{with probability } (1 - \text{HMCR}) \end{cases} \\ D'_i &= \begin{cases} D'_i \in \{D_1^1, D_1^2, \dots, D_1^{\text{HMS}}\} & \text{with probability HMCR} \\ D'_i \in D & \text{with probability } (1 - \text{HMCR}). \end{cases} \end{aligned} \quad (18)$$

T and D in formulas (18) represent all possible values of T'_i and D'_i , respectively, which are discrete variables determined by the number of inbound and outbound trucks and the number of inbound and outbound doors.

If the value of a variable is obtained by memory consideration, it should be determined whether to be pitch-adjusted with probability PAR.

$$\begin{aligned} &\text{Pitch adjusting decision for } T'_i \\ &= \begin{cases} \text{Yes} & \text{with probability PAR,} \\ \text{No} & \text{with probability } (1 - \text{PAR}). \end{cases} \end{aligned} \quad (19)$$

$$\begin{aligned} &\text{Pitch adjusting decision for } D'_i \\ &= \begin{cases} \text{Yes} & \text{with probability PAR,} \\ \text{No} & \text{with probability } (1 - \text{PAR}). \end{cases} \end{aligned}$$

The value will be adjusted if the pitch adjusting decision is yes, and the adjusting formulas are as follows:

$$\begin{aligned} T'_i &= T'_i \pm \text{rand}() * \text{BW} \\ D'_i &= D'_i \pm \text{rand}() * \text{BW}. \end{aligned} \quad (20)$$

3	3	1	5	5	4	6	6	3	5	3
1	2	1	1	2	1	3	2	2	3	1

FIGURE 5: A new solution of HS.

3	2	1	5	4	4	6	2	3	5	1
1	2	1	1	2	1	3	2	2	3	1

FIGURE 6: A modified solution of HS.

Here, $\text{rand}()$ is a random number between 0 and 1.

The new solution may not meet the model requirements after the generating rules, so some corrections are needed. As shown in Figure 5, there are repeated 3 and repeated 5 in the inbound truck codes at the first level of the solution. It could not be real for the reasonable solution. The position of the second 3 should be replaced by the missing number 2 or 4 randomly, and the position of the second 5 should be replaced by the rest missing number. As for the outbound truck codes, the same approach should be applied to. Figure 6 represents a modified solution that may occur.

3.1.5. Update the Solution Set. Considering the objective function value, replace the worst solution by the new solution, if the new solution is better than the worst in the HM; otherwise, remain the worst and generate a new solution again.

3.1.6. Check Stopping Criterion. The maximum cycle time is defined; the loop will continue until it meets the stopping criterion.

3.1.7. Improved Harmony Search Algorithm. The parameters of basic harmony search are fixed, but parameters perform differently in the every iteration. The algorithm that possesses small parameter of PAR with large parameter of BW performs worse and increases the iteration needed to find the optimum solution. Furthermore, large PAR and small BW will cause great improvement in final generations with algorithm converged to optimal solution vector. Considering the parameter of BW separately, small BW in final generations increase the fine-tuning of solution vectors, but large BW performs better in the early iterations and increases the diversity of solution vectors to search for the best solution.

To eliminate the drawbacks of HS algorithm with fixed values of PAR and BW and improve the performances of them, dynamic values of parameters will be applied to the basic HS algorithm. The formulas of dynamic parameters are defined as follows [20, 21]:

$$\begin{aligned} \text{PAR}(\text{gn}) &= \text{PAR}_{\min} + \frac{(\text{PAR}_{\max} - \text{PAR}_{\min})}{NI} \times \text{gn} \\ \text{BW}(\text{gn}) &= \text{BW}_{\max} \exp(c \times \text{gn}) \\ c &= \frac{\ln(\text{BW}_{\min}/\text{BW}_{\max})}{NI}. \end{aligned} \quad (21)$$

PAR(gn) and BW(gn) are the values of PAR and BW in each generation, $\text{PAR}_{\max}$ and $\text{PAR}_{\min}$ are the maximum and minimum pitch adjusting rate, $\text{BW}_{\max}$ and $\text{BW}_{\min}$ are the maximum and minimum bandwidth, gn represents the generation number, and NI represents the solution vector generations.

3.2. Genetic Algorithm. There are several steps in genetic algorithm, such as population initialization, selection, crossover, and mutation operation. The population initialization process is similar to the initialization of solution sets in harmony search, and the selection method is roulette. So here are some explanations for crossover and mutation operation.

3.2.1. Crossover Design of Chromosomes. The chromosomes of this paper consist of two parts that are inbound trucks codes and outbound trucks codes; the crossover operations need to be done in two parts. As shown in Figure 7, two cross points are generated randomly as 2 and 3, which means the cross point of inbound trucks sequence is between the locations two and three; likewise, cross point of outbound trucks sequence is between the locations three and four. A solution is generated but does not meet the requirements. Therefore, some modified operations like red digits are applied following what HS does.

3.2.2. Mutation Design of Chromosomes. In order to perform the mutation operations for inbound and outbound trucks, swap mutation is applied. As shown in Figure 8, for inbound trucks sequence, the value at mutation point 1 and mutation point 2 will be changed, and the outbound trucks sequence mutates likewise.

4. Parameters Setting

There are several parameters in each algorithm. Meanwhile, each factor varies from different levels. However, the value of each parameter could affect the performances of the algorithms. Hence, several methods should be carried out to find out the best combination of parameters in each algorithm.

Taguchi experiment is a kind of method to design experiments, which decreases the number of experiments but provides an efficient combination of variables. Each

Crossing point ↓					Crossing point ↓					
3	4	1	2	5	4	2	6	1	5	3
1	2	1	1	2	1	3	2	2	3	1

1	3	2	5	4	6	5	2	1	4	3
2	2	1	2	2	1	3	1	2	1	1

3	4	2	5	4 ₁	4	2	6	1	4 ₅	3
1	2	1	2	2	1	3	2	2	1	1

1	3	1 ₄	2	5	6	5	2	1	5 ₄	3
2	2	1	1	2	1	3	1	2	3	1

FIGURE 7: The process of crossover.

Mutation point 1		Mutation point 2			Mutation point 3			Mutation point 4		
↓		↓			↓			↓		
3	4	1	5	4	4	2	6	1	4	3
1	2	2	1	2	1	3	2	2	1	1
1	4	3	5	4	4	4	6	1	2	3
2	2	1	2	2	1	1	2	2	3	1

FIGURE 8: The process of mutation.

TABLE 1: Levels of parameters for HS.

Parameters	Levels			
	L1	L2	L3	L4
HMS	10	20	30	40
HMCR	0.7	0.9	0.95	0.99
PAR	0.2	0.4	0.6	0.7
BW	0.01	0.35	0.7	1

factor can be compared independently with any other factors, which ensures a balanced comparison among all levels of any factors. For the above reasons, Taguchi experiments are widely used in engineering design to determine optimal experimental conditions. The performance of each algorithm is evaluated by the signal to noise (S/N) to search combination of factors with a small amount of variation.

In the Taguchi method, the measured values of searching for the optimal combination of the parameters will be transformed into S/N ratio. The objective of the function in

this paper is the small-the-better type, and the formula is as follows:

$$S/N \text{ ratio} = -10 \log \frac{1}{n} \left[\sum_{i=1}^n \text{objective function}_i^2 \right]. \quad (22)$$

There are four parameters in HS, which are HMS, HMCR, PAR, and BW, each of whom has four levels. Table 1 displays different levels of the parameters, and the best combination is that the values of HMS, HMCR, PAR, and BW are 30, 0.95, 0.6, and 0.7, respectively, as shown in Figure 9.

FIGURE 9: Main effects plot for S/N ratios of HS factors.

FIGURE 10: Main effects plot for S/N ratios of IHS factors.

TABLE 2: Levels of parameters for IHS.

Parameters	Levels		
	L1	L2	L3
HMS	10	20	30
HMCR	0.7	0.85	0.99
PAR _{min}	0.05	0.25	0.4
PAR _{max}	0.6	0.8	0.9
BW _{min}	0.05	0.25	0.4
BW _{max}	0.8	0.9	1

There are six parameters in IHS, which are HMS, HMCR, PAR_{min}, PAR_{max}, BW_{min}, and BW_{max}, each of whom has three levels. Table 2 displays different levels of the parameters, and the best combination is that the values of HMS, HMCR, PAR_{min}, PAR_{max}, BW_{min}, and BW_{max} are 30, 0.85, 0.05, 0.6, 0.05, and 0.9, respectively, as shown in Figure 10.

There are four parameters in GA, which are P_c , P_m , P_s , and POP, representing crossover probability, mutation probability, selection probability, and population size, respectively, each of whom has four levels. Table 3 suggests different levels of the parameters, and the best combination of values for P_c , P_m , P_s , and POP are 0.8, 0.2, 0.5, and 20, respectively, as shown in Figure 11.

5. Computational Results

5.1. Illustrative Examples. The number of inbound doors and outbound doors in the docking system are 2 and 3, respectively. In order to illustrate the performance of each proposed algorithm, several cases have been proposed to be experimented. The problems are divided into nine cases, considering the number of trucks, products, and their types. The subproblems 1–3 are divided due to the number of the products and types, of which subproblem 1 represents the small size, subproblem 2 is on behalf of the medium size, and subproblem 3 represents the large size. In addition, each subproblem is divided further, due to the number of

FIGURE 11: Main effects plot for S/N of GA factors.

TABLE 3: Levels of parameters for GA.

Parameters	Levels			
	$L1$	$L2$	$L3$	$L4$
P_c	0.6	0.7	0.8	0.9
P_m	0.05	0.1	0.15	0.2
P_s	0.2	0.3	0.4	0.5
POP	10	20	30	40

inbound and outbound trucks. Table 4 displays the detailed information of the nine cases.

It should be mentioned that the moving time of products from the inbound door to the outbound door is related to the distance, which means the moving time is 30, if doors are far apart, otherwise it is 15. And the changeover time of trucks is considered equal to 50 units; furthermore, the loading time

is 1 unit, which is the equal to the unloading time; all data is processed by MATLAB 2016.

5.2. Comparison of Solution Procedures. For the experiments, the percentage deviation of the solutions in each method from the optimal solution based on makespan and the computational time are used to evaluate the performances. Percentage deviation of a solution is calculated as follows:

$$\text{Percentage deviation (\%)} = \frac{(\text{Evaluative factor for compared solution}) - (\text{Evaluative factor for best solution})}{(\text{Evaluative factor for best solution})} \times 100\%. \quad (23)$$

In order to evaluate performances of the three proposed methods in terms of the optimal value, worst value, average value, and the computational time, each case is tested 20 times and terminates at the iteration of 120. Table 5 shows makespan and computational time of nine tests by the three methods. As can be seen in Table 5, in the aspect of the optimal solution, there are 6 sets of optimal solutions searched by GA, 4 sets by IHS, and, nevertheless, 1 set by HS. In the aspect of the worst solution, IHS performs better, which gets 6 sets of best solutions among the worst solutions. As for average solution, the average values of nine cases calculated by GA are all showing best among the three methods. In terms of computational time, IHS performs best, while GA need much more time than the other methods.

As shown in Table 6, the percentage deviations of the optimal makespan searched by GA from the optimal makespan searched by three methods range from 0% to 6.18% for nine cases and range from 0% to 5.72% for IHS. The percentage deviations of the worst makespan searched by HS from the optimal makespan among the worst solutions searched by three methods range from 0% to 6.18% for nine cases and range from 0% to 11.1% for GA. GA performs best in average makespan; however, the percentage deviations of the average value calculated by HS from the best among the average values calculated by three methods range from 0.18% to 8.8% for nine cases. The percentage deviations of the computational time of GA from the shortest computational time among the three methods range from 188.39% to 511.59% for nine cases,

TABLE 4: Nine sets of numerical experiments.

Subproblem 1 (small)			Subproblem 2 (medium)			Subproblem 3 (large)		
Case 1			Case 2			Case 3		
Product	357	Product	370	Product	399	Product	370	Product
Type	5	Type	5	Type	6	Type	5	Type
Inbound truck	6	Inbound truck	15	Inbound truck	27	Inbound truck	15	Inbound truck
Outbound truck	5	Outbound truck	15	Outbound truck	33	Outbound truck	15	Outbound truck
Case 4			Case 5			Case 6		
Product			Product			Product		
Type			Type			Type		
Inbound truck			Inbound truck			Inbound truck		
Outbound truck			Outbound truck			Outbound truck		
Case 7			Case 8			Case 9		
Product			Product			Product		
Type			Type			Type		
Inbound truck			Inbound truck			Inbound truck		
Outbound truck			Outbound truck			Outbound truck		

TABLE 5: Makespan and computational time obtained by three methods for the nine test problems.

Problem set	HS				IHS				GA			
	Optimal	Worst	Average	Computational time	Optimal	Worst	Average	Computational time	Optimal	Worst	Average	Computational time
1	198	254	233.2	17.301	187	238	220.4	16.734	196	248	209.9	48.260
2	455	498	477.05	29.280	433	474	456.9	27.611	419	489	441.35	90.881
3	873	964	920.8	29.442	866	937	903.9	26.109	820	1012	869.85	159.679
4	429	467	445.6	23.842	429	471	445.5	24.480	429	465	444.7	97.429
5	687	798	741.65	40.347	647	750	721.35	39.253	687	777	689	141.686
6	922	1010	975.4	36.304	905	984	933.25	38.102	856	983	918.6	176.780
7	791	1011	926.3	31.501	784	979	928.25	32.548	791	922	853.2	109.181
8	1044	1193	1112.1	51.448	963	1093	1044.9	49.160	948	1095	1011	176.188
9	1166	1337	1231.3	50.592	1153	1243	1195.7	49.094	1110	1381	1186.4	268.185

TABLE 6: Percentage deviation for makespan and computational time between optimal solutions and other solutions.

Problem set	Optimal solution makespan	Optimal worst solution makespan	Optimal average solution makespan	Optimal computational time	RPD for optimal solutions			RPD for worst solutions			RPD for average solutions			RPD for computational time		
					HS	IHS	GA	HS	IHS	GA	HS	IHS	GA	HS	IHS	GA
1	187	238	209.9	16.734	5.88%	0.00%	4.81%	6.72%	0.00%	4.20%	11.10%	5.00%	0.00%	3.39%	0.00%	188.39%
2	419	474	441.35	27.611	8.59%	3.34%	0.00%	5.06%	0.00%	3.16%	8.09%	3.52%	0.00%	6.04%	0.00%	229.15%
3	820	937	869.85	26.109	6.46%	5.61%	0.00%	2.88%	0.00%	8.00%	5.86%	3.91%	0.00%	12.77%	0.00%	511.59%
4	429	465	444.7	23.842	0.00%	0.00%	0.00%	0.43%	1.29%	0.00%	0.20%	0.18%	0.00%	0.00%	2.68%	308.64%
5	647	750	689	39.253	6.18%	0.00%	6.18%	6.40%	0.00%	3.60%	7.64%	4.70%	0.00%	2.79%	0.00%	260.96%
6	856	983	918.6	36.304	7.71%	5.72%	0.00%	2.75%	0.10%	0.00%	6.18%	1.59%	0.00%	0.00%	4.95%	386.94%
7	784	922	853.2	31.501	0.89%	0.00%	0.89%	9.65%	6.18%	0.00%	8.57%	8.80%	0.00%	0.00%	3.32%	246.6%
8	948	1093	1011	49.160	10.13%	1.58%	0.00%	9.15%	0.00%	0.18%	10.00%	3.35%	0.00%	4.65%	0.00%	258.4%
9	1110	1243	1186.4	49.094	5.05%	3.87%	0.00%	7.56%	0.00%	11.10%	3.78%	0.78%	0.00%	3.05%	0.00%	446.27%

TABLE 7: Makespan and percentage deviation under the operation times 20 s and 40 s.

Problem set	Operation time 20 s						Operation time 40 s					
	IHS		HS		GA		IHS		HS		GA	
	Makespan	PD	Makespan	PD	Makespan	PD	Makespan	PD	Makespan	PD	Makespan	PD
1	235.6	0.00%	240	1.87%	293.6	24.62%	220.8	0.00%	222	0.54%	264.7	19.88%
2	473.7	0.00%	487.3	2.87%	675.9	42.69%	473.8	0.00%	481.4	1.60%	612.2	29.21%
3	924.9	0.00%	945.2	2.19%	1266.1	36.89%	913.4	0.00%	937.1	2.59%	1215.8	33.11%
4	464.4	0.00%	465.3	0.19%	539	16.06%	437.4	0.00%	451.6	3.25%	482.1	10.22%
5	765.6	0.00%	776	1.36%	968.1	26.45%	759.5	0.00%	760.5	0.13%	942.4	24.08%
6	959.1	0.00%	997.8	4.04%	1283.8	33.85%	966.5	0.00%	973.1	0.68%	1198.1	23.96%
7	977.2	3.48%	944.3	0.00%	1077.3	14.08%	937.4	1.32%	925.2	0.00%	1024.5	10.73%
8	1142.5	0.00%	1155.3	1.12%	1393.4	21.96%	1083.7	0.00%	1104.1	1.88%	1351.9	24.75%
9	1298.7	0.00%	1308.6	0.76%	1386.5	6.76%	1235.5	0.00%	1272.1	2.96%	1316.9	6.59%

which show GA performs worst in computational time and needs much more time than other methods.

To evaluate the performances of three algorithms under the same operation time, nine sets of experiments will be terminated at 20 s and 40 s using IHS, HS, and GA, respectively. To improve the accuracy of the results, each experiment will be run 10 times so as to get the average value. As shown in Table 7, for each method, the makespan get shorter, when the operation time is longer. IHS performs best among them for the same running time; however, GA performs worst. The percentage deviations (PD) of the values searched by GA from the best values get smaller when operation time is longer. However, the percentage deviations get larger for HS when operation time is longer.

6. Conclusions

This study has described a truck scheduling model in cross-docking system with multiple doors; furthermore, three algorithms are proposed, namely, HS, IHS, and GA to solve the problem. To evaluate the performances of the methods, these algorithms were applied for 9 sets of different problems divided by the number of products, categories, and trucks to minimize the total makespan of cross-docking procedure. During the analysis, it is proved that IHS performs better than HS in searching for optimal solutions, when dynamic values of parameters are applied. In addition, for the same iteration, both IHS and GA provide satisfactory results. However, GA performs better in providing optimal solutions, while IHS is far superior to GA in computational time. For the same operation time, HS performs best; however GA performs worst. For future studies, new hybrid metaheuristic algorithms will be applied to cross-docking system, instead of the basic algorithms, and the truck scheduling model could be perfect in uncertainty or fuzzy arrival time of trucks.

Conflicts of Interest

There are no conflicts of interest regarding the publication of this paper.

References

- [1] J. Van Belle, P. Valckenaers, G. V. Berghe, and D. Cattrysse, "A tabu search approach to the truck scheduling problem with multiple docks and time windows," *Computers & Industrial Engineering*, vol. 66, no. 4, pp. 818–826, 2013.
- [2] J. J. Bartholdi III and K. R. Gue, "Reducing labor costs in an LTL crossdocking terminal," *Operations Research*, vol. 48, no. 6, pp. 823–832, 2000.
- [3] R. Dondo and J. Cerdá, "A monolithic approach to vehicle routing and operations scheduling of a cross-dock system with multiple dock doors," *Computers & Chemical Engineering*, vol. 63, no. 11, pp. 184–205, 2014.
- [4] W. Yu and P. J. Egbelu, "Scheduling of inbound and outbound trucks in cross docking systems with temporary storage," *European Journal of Operational Research*, vol. 184, no. 1, pp. 377–396, 2008.
- [5] A. R. Boloori Arabani, S. M. T. Fatemi Ghomi, and M. Zandieh, "Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage," *Expert Systems with Applications*, vol. 38, no. 3, pp. 1964–1979, 2011.
- [6] S. Forouharfard and M. Zandieh, "An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems," *The International Journal of Advanced Manufacturing Technology*, vol. 51, no. 9–12, pp. 1179–1193, 2010.
- [7] W. Yu, "Truck scheduling for cross docking systems with multiple receiving and shipping docks," *International Journal of Shipping and Transport Logistics*, vol. 7, no. 2, pp. 174–196, 2015.
- [8] M. Madani-Isfahani, R. Tavakkoli-Moghaddam, and B. Naderi, "Multiple cross-docks scheduling using two meta-heuristic algorithms," *Computers & Industrial Engineering*, vol. 74, pp. 129–138, 2014.
- [9] M. T. Assadi and M. Bagheri, "Scheduling trucks in a multiple-door cross docking system with unequal ready times," *European Journal of Industrial Engineering*, vol. 10, no. 1, pp. 103–125, 2016.
- [10] M. T. Assadi and M. Bagheri, "Differential evolution and Population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems," *Computers & Industrial Engineering*, vol. 96, no. 20, pp. 149–161, 2016.
- [11] W. Wisittipanich and P. Hengmeechai, "A multi-objective differential evolution for Just-In-Time door assignment and truck scheduling in multi-door Cross docking problems," *Industrial*

- Engineering & Management Systems*, vol. 14, no. 3, pp. 299–311, 2015.
- [12] A. R. Boloori Arabani, S. M. T. Fatemi Ghomi, and M. Zandieh, “A multi-criteria cross-docking scheduling with just-in-time approach,” *The International Journal of Advanced Manufacturing Technology*, vol. 49, no. 5-8, pp. 741–756, 2010.
 - [13] K. Lee, B. S. Kim, and C. M. Joo, “Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance,” *Expert Systems with Applications*, vol. 39, no. 17, pp. 12975–12983, 2012.
 - [14] A.-L. Ladier and G. Alpan, “Crossdock truck scheduling with time windows: earliness, tardiness and storage policies,” *Journal of Intelligent Manufacturing*, 2014.
 - [15] B. Vahdani, R. Soltani, and M. Zandieh, “Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics,” *The International Journal of Advanced Manufacturing Technology*, vol. 46, no. 5-8, pp. 769–783, 2010.
 - [16] A. Mohtashami, “Scheduling trucks in cross docking systems with temporary storage and repetitive pattern for shipping trucks,” *Applied Soft Computing*, vol. 36, pp. 468–486, 2015.
 - [17] N. Boysen, D. Briskorn, and M. Tschöke, “Truck scheduling in cross-docking terminals with fixed outbound departures,” *OR Spectrum*, vol. 35, no. 2, pp. 479–504, 2013.
 - [18] S. Rahmanzadeh Tootkaleh, S. M. T. Fatemi Ghomi, and M. Sheikh Sajadieh, “Cross dock scheduling with fixed outbound trucks departure times under substitution condition,” *Computers & Industrial Engineering*, vol. 92, pp. 50–56, 2016.
 - [19] Z. W. Geem, J. H. Kim, and G. V. Loganathan, “A new heuristic optimization algorithm: harmony search,” *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
 - [20] M. Mahdavi, M. Fesanghary, and E. Damangir, “An improved harmony search algorithm for solving optimization problems,” *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.
 - [21] M. R. Valaei and J. Behnamian, “Allocation and sequencing in 1-out-of-N heterogeneous cold-standby systems: Multi-objective harmony search with dynamic parameters tuning,” *Reliability Engineering & System Safety*, vol. 157, pp. 78–86, 2017.