

Research Article

A Hybrid Genetic Algorithm for Integrated Truck Scheduling and Product Routing on the Cross-Docking System with Multiple Receiving and Shipping Docks

Wooyeon Yu ¹, Chunghun Ha ², and SeJoon Park ³

¹Department of Industrial and Management Engineering, Myongji University, 116 Myonggi-Ro, Cheoin-Gu, Yongin-Si, Gyeonggi-Do 17058, Republic of Korea

²Dept. of Industrial Engineering, Hongik University, 94 Wausan-ro, Mapo-gu, Seoul 04066, Republic of Korea

³Division of Energy Resources Engineering and Industrial Engineering, Kangwon National University, KNU Chuncheon Campus, 1 Gangwondaehakgil, Chuncheon-Si, Gangwon-Do 24341, Republic of Korea

Correspondence should be addressed to SeJoon Park; sonmupsj@hanmail.net

Received 29 July 2020; Revised 10 November 2020; Accepted 10 March 2021; Published 18 March 2021

Academic Editor: Ricardo Aguilar-Lopez

Copyright © 2021 Wooyeon Yu 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.

In this research, a truck scheduling problem for a cross-docking system with multiple receiving and shipping docks is studied. Until recently, single-dock cross-docking problems are studied mostly. This research is focused on the multiple-dock problems. The objective of the problem is to determine the best docking sequences of inbound and outbound trucks to the receiving and shipping docks, respectively, which minimize the maximal completion time. We propose a new hybrid genetic algorithm to solve this problem. This genetic algorithm improves the solution quality through the population scheme of the nested structure and the new product routing heuristic. To avoid unnecessary infeasible solutions, a linked-chromosome representation is used to link the inbound and outbound truck sequences, and locus-pairing crossovers and mutations for this representation are proposed. As a result of the evaluation of the benchmark problems, it shows that the proposed hybrid GA provides a superior solution compared to the existing heuristics.

1. Introduction

A distribution center (or a warehouse) employing cross-docking unloads, sorts, and reorganizes the products that were delivered by the inbound trucks depending on customer orders and then immediately transfers and loads the consolidated products into the outbound trucks as shown in Figure 1. The products are not normally stored, even if they are stored temporarily for less than 24 hours. Such just-in-time scheme of the cross-docking system enables to eliminate (or reduce) storing and retrieving operations. The following are the most expensive operations among fundamental warehouse operations: receiving, sorting, storing, retrieving, and shipping. The cross-docking system also improves customer service through quick responsiveness of customer needs and high accuracy of product shipments. So,

it has become an integral part for the supply chain's competitive advantage.

Factors affecting the performance of a cross-docking system can be broadly classified into design factors and operational factors. Design factors such as location, layout, number of receiving and shipping docks, and size of temporary storage determine the capacity and inherent efficiency of the cross-docking system at the distribution center. These factors become constraints for optimization because they are already determined at the time of construction of the distribution center. On the contrary, operational factors such as processes of sorting and consolidation and schedules of inbound and outbound trucks need to be optimized according to changing conditions of the cross-docking system to increase efficiency. The problem of optimizing operational factors under design constraints to maximize the

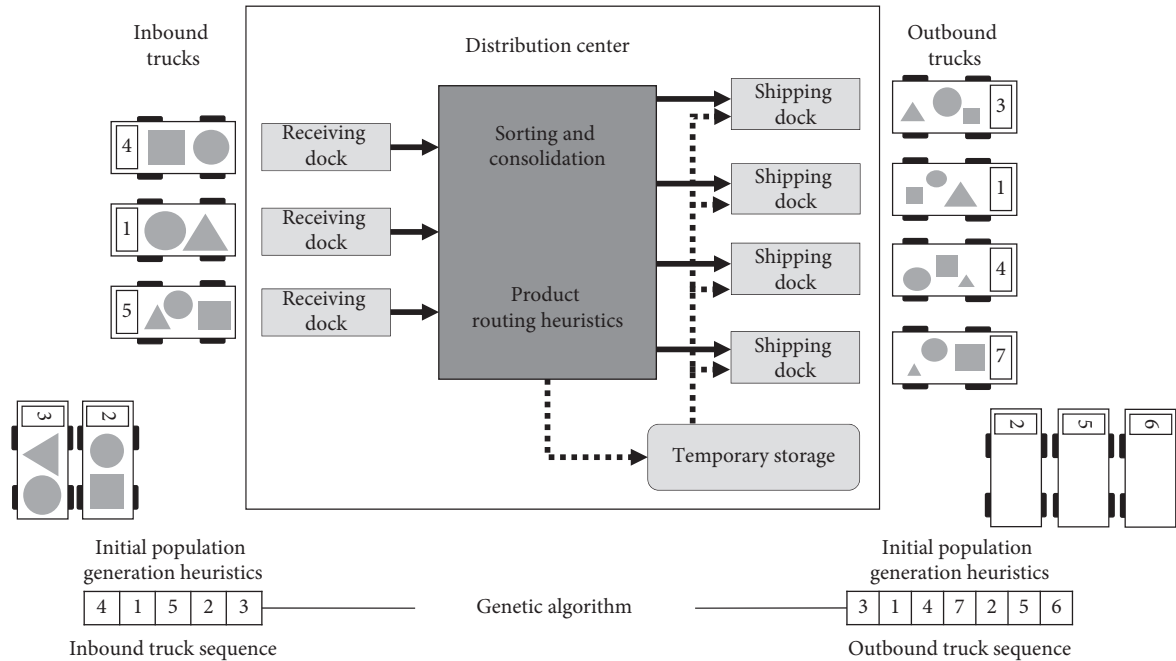


FIGURE 1: A cross-docking system with multiple receiving and shipping docks and temporary storage and the scope of our research.

performance in a cross-docking system is commonly referred to as truck scheduling problem (TRSP). In the TRSP, representative decision variables are sequences of inbound and outbound trucks and routing procedures for the products that are transferred from inbound trucks to outbound trucks. In the cross-docking system, all decision variables must be determined simultaneously because the types and quantities of the transferred products must be matched at the inbound and the outbound trucks each other. The TRSP, which has a strong NP-hard complexity, is generally modeled as mixed-integer programming (MIP) [1]. Many researchers have dealt with various TRSPs depending on the configuration of the distribution center and organizational and operational environments. However, realistic TRSPs are not easy to define exact mathematical programming models and find optimal solutions because of their high complexity. Thus, research on simplified models, which are somewhat different from actual ones, has become the mainstream [2]. Recently, studies dealing with TRSPs on cross-docking systems of practically configurable complex configurations have been increasing to overcome this problem [3–6].

The purpose of this study is to develop an improved algorithm to derive the pseudo-optimal solution of the TRSP for cross-docking systems with complex configurations. The proposed hybrid genetic algorithm (GA) uses GA for optimization of inbound and outbound truck sequences and greedy heuristic for routing of products. In general, the schedule of outbound trucks depends on customer orders (and products). Inbound trucks must have appropriate products in order to load the products required by outbound trucks just in time. Therefore, the optimal inbound truck sequence depends on the outbound truck sequence. It is difficult to solve this complex and integrated chain process

with only one approach. For this reason, we propose a hybrid GA. The proposed hybrid GA is verified by a cross-docking system with multiple receiving and shipping docks and temporary storage by Yu [3].

The single-dock cross-docking system has been researched until recently. The 20 TRSP benchmark problems for the single-dock cross-docking system are provided in [7]. Then, Golshahi-Roudbaneh et al. solved these benchmark problems with their heuristic and provided additional 15 benchmark problems [8]. These 35 benchmark problems are solved by red deer algorithm, virus colony search, water wave optimization, and social engineering optimization [9, 10]. This study contributes to improving the solution of the 20 TRSP benchmark problems presented by Yu [3].

This paper is organized as follows. After this introductory section, a brief literature review is performed in Section 2. Section 3 describes the cross-docking model studied in this research and analyzes characteristics of the model. Then, in Section 4, a thorough explanation of the proposed hybrid genetic algorithm for the cross-docking system is given. Section 5 compares the genetic algorithm performance to that of previously introduced methods. Finally, Section 6 concludes the paper and provides an outlook toward future research areas.

2. Literature Review

Depending on the purpose of configuration and optimization of the cross-docking system, mathematical programming models and optimization techniques for various TRSPs have been proposed. Most previous TRSPs dealt with have focused on a simplified cross-docking system which is composed of a single receiving dock, a single shipping dock, and temporary storage. Such a simplified configuration is

somewhat different from reality, but simple, so it is easy to solve using traditional mathematical programming approaches. Yu and Egbelu [7], Vahdani and Zandieh [11], Arabani et al. [12], Liao et al. [13], Vahdani et al. [14], and Shiguemoto et al. [15] tackled these types of TRSPs to minimize makespan that is the completion time of all operations in the cross-docking system.

Some studies are conducted on extensions of the simple TRSP. Soltani and Sadjadi [16] dealt with the cross-docking system of zero temporary storage. Choy et al. [17] considered a cross-docking system with a limited storage space. They developed a mathematical programming model and a genetic algorithm for minimizing the waiting time of inbound and outbound trucks by finding the optimal job sequence of pickup and delivery tasks in the storage zone. Sadykov [18] investigated a TRSP with the objective of minimizing the storage usage during the product transfer. The author demonstrated that the TRSP is NP-hard in the strong sense even with only two product types. If the case of the subsequences of inbound and outbound trucks is fixed, dynamic programming is proposed.

Some research studies have considered multiple objectives for the TRSP. Boysen [19] focused on minimization of flow time, processing time, and tardiness of outbound trucks. Arabani et al. [20] studied a TRSP to minimize total operation time and total lateness of the outbound trucks. To solve the problem, they proposed three multiobjective approaches based on the subpopulation concept of evolutionary algorithm. Fazel Zarandi et al. [21] developed a constraint satisfaction problem and an integer programming model for a multicriteria TRSP. They proposed a two-phase genetic algorithm approach toward solving the problem. In the first phase, tardiness and earliness costs of products were minimized. Then, in the next phase, optimization of the number of preemptions for outbound trucks was performed. Naderi et al. [22] targeted on minimizing both makespan and total tardiness. To solve the problem, the authors proposed a multiobjective iterative algorithm with several advanced features such as a modified crowding selection, a restart phase, and a local search.

Although the simplified cross-docking system is rather impractical, studying it has provided good insights into cross-docking operation and contributes to understanding more complicated cross-docking systems. Recently, some researchers have dealt with more realistic cross-docking systems considering multiple receiving and shipping docks. TRSP for multiple docks basically addresses the scheduling of only inbound trucks by assuming that the outbound truck sequence is fixed or known a priori. Since the schedule of outbound trucks depends on customer needs, this assumption is realistically valid. Konur and Golias [23] studied the TRSP for multiple docks considering uncertainty of inbound truck arrival time, in which the lower and upper bounds of them were given. They proposed a genetic algorithm to minimize the expected overall service time, which is the sum of handling and expected waiting times of the inbound trucks. Boysen et al. [24] also dealt with multiple docks. The objective was to minimize the overall value of delayed shipments. They developed a mixed-integer

programming model and solved the problem using the proposed two heuristic algorithms, including decomposition procedures and simulated annealing for solving the problem. Liao et al. [25] considered simultaneous dock assignment and sequencing of inbound trucks for a cross-docking system with multiple docks, with the objective of minimizing the overall weighted tardiness, in which a fixed departure schedule of each outbound truck is assumed. They compared the performance of six different metaheuristics: differential evolution, simulated annealing, ant colony optimization, tabu search, and two hybrid differential evolution algorithms. Then, they concluded that the ant colony optimization is superior to others.

TRSP for the cross-docking system with multiple docks can be extended to an integrated problem of solving inbound truck scheduling and outbound truck scheduling simultaneously. If the schedule of outbound trucks is adjustable, this extended TRSP allows for more efficient scheduling. Guo et al. [26] studied such an extended TRSP for multiple docks with the objective of minimizing operation cost and ensuring punctuality, in which the operation cost depends on the total distance traveled by the products from the receiving docks to the shipping docks and the temporal punctuality is measured by a combination of earliness and tardiness. To solve the problem, the authors proposed a nondominated sorting genetic algorithm with a novel greedy local search strategy. Lee et al. [27] considered the TRSP to maximize the number of products that could be shipped within a given working horizon. They derived a mixed-integer programming model and solved the problem using the proposed genetic algorithms with three different chromosome representations. Van Belle et al. [28] studied the extended TRSP with the objective of minimizing the weighted combination of overall travel time and overall tardiness. They presented a mixed-integer programming model and proposed a tabu search for solving the problem. Joo and Kim [29] studied the TRSP to minimize makespan considering three different types of trucks: trucks visiting only the receiving dock for unloading products, trucks visiting only the shipping dock for loading the products, and trucks visiting the receiving dock for unloading the products and then visiting the shipping dock for loading the products. The objective was to simultaneously determine door assignment and docking sequences for all three types of trucks to minimize makespan. To tackle it, they proposed a genetic algorithm and a self-evolution algorithm. Kuo [30] proposed a makespan calculation model based on the first-come and first-served door assignment policy of a given sequence of all inbound and outbound trucks. The makespan calculation model was integrated with a variable neighborhood search that tried to optimize the sequence of all inbound and outbound trucks. Madani-Isfahani et al. [31] considered the extended TRSP under a limited capacity, in which trucks could come in and out of the docks multiple times until all of their loads have been unloaded or loaded, and the trucks could enter any of the cross-docks. To solve the problem, the authors developed a mixed-integer programming model and solved it by simulated annealing and a firefly algorithm. Yazdani et al. [32] developed a mixed-integer programming model and

proposed a hunting search metaheuristic inspired by the foraging group behavior of animals. Yu [3] developed a mixed-integer programming model for the extended TRSP and solved it by two heuristic algorithms: a neighborhood search algorithm and a two-nested-loop search with mutation. In order to compare the performance of them, he derived the lower bound on the makespan for the problem.

In recent years, there is an increasing number of studies attempting to deal with TRSPs for more flexible cross-docking systems. Bodnar et al. [5] and Rijal et al. [4] studied the TRSP for a cross-docking system that takes into account mixed-mode doors that can be used as both receiving and shipping docks. Fard and Vahdani [6] proposed a multiobjective imperialist competitive algorithm and a multiobjective gray wolf optimizer to simultaneously optimize product holding costs and energy consumption in cross-dock systems with mixed-mode doors.

Additional detailed reviews on the studies of cross-docking systems can be found in Boysen and Flidner [33], Van Belle et al. [34], and Ladier and Alpan [2].

3. TRSP

The cross-docking system covered in this paper operates as follows. As depicted in Figure 1, a distribution center consists of multiple receiving docks, multiple shipping docks, sorting and consolidation facility, and a temporary storage area. Inbound (outbound) trucks arriving at the distribution center are assigned to empty receiving (shipping) docks sequentially according to a predetermined inbound (outbound) truck sequence. If all receiving (shipping) docks are in use, the next inbound (outbound) truck will wait for a new assignment. An inbound truck assigned to a receiving dock begins to unload all products on board and exits the dock as soon as unloading is completed. The next turn inbound truck waiting is immediately assigned to the empty receiving dock. Unloaded products are passed, split, or merged through the sorting and consolidation facility (usually a conveyor system) and eventually moved to a shipping dock where a designated outbound truck is waiting. An outbound truck that has loaded all the required quantities of products leaves the shipping dock. The next turn outbound truck in standby immediately moves to its empty receiving dock. These cross-docking procedures are terminated when all required operations are completed, that is, all outbound trucks load all required products and leave the distribution center.

In this cross-docking procedure, we apply the following assumptions. First, the type and quantity of products to unload (load) all inbound (outbound) trucks are already determined and do not change during the procedure. Also, all inbound (outbound) trucks are already waiting at the distribution center before the cross-docking procedure begins. These fundamental assumptions isolate the TRSP from other problems such as the vehicle routing problem. Second, all receiving (shipping) docks are identical. That is, an inbound (outbound) truck produces the same result on any receiving (shipping) dock. The independence between trucks and docks makes the truck sequences the only

decision variables. Third, transportation time of any product at the distribution center is a constant regardless of the type and quantity [3]. That is, the transportation time of all products from the receiving dock to the shipping dock (or temporary storage) and from temporary storage to the shipping dock is the same. Under this assumption, the arrival sequence of products at shipping docks is the same as their unloading sequence at receiving docks. If this assumption does not apply, it is almost impossible to build a mathematical programming model because the loading and unloading orders of all products must be considered as the decision variables. Fourth, unlimited size of temporary storage is introduced to increase the utilization of docks. If the unloaded quantity of a product differs from the quantity that must be loaded, the difference quantity is stored in the temporary storage area until the next required quantity. Fifth, once the truck enters the dock, the truck is allowed to leave the dock after completion of all loading or unloading activities. In a single dock, the assumption of preemption is essential to increase efficiency. However, in multiple docks, preemption is not considered as reducing the replacement time of trucks is more efficient to achieve the goal. Sixth, when unloading and loading a product, only one unit of the product can be processed at a time. Therefore, loading and unloading times are proportional to the quantity of the product.

The objective function of our TRSP is makespan, which is the completion time of the last operation at a given assignment. Operations in a TRSP start when the first product of the first scheduled inbound truck is unloaded onto a receiving dock and end when the last product of the last departing outbound truck is loaded from a shipping dock. Yu [3] developed a mixed-integer programming (MIP) model for the above cross-docking system, and we apply his MIP model.

4. Proposed Hybrid GA

Since the introduction of GA by Holland [35], GA has been proven to be one of the most successful metaheuristics to solve NP-hard or NP-complete problems. GA is an optimization technique of obtaining pseudo-optimum by probabilistic search of solution space through the evolution of a population consisting of a large number of individuals. Every individual has a chromosome, usually in the form of an array. The population evolves through the selection of parents, generation of offspring through genetic operations (crossover and mutation) between the parent's chromosomes, and reentry process of offspring into the population according to fitness.

The representation of the chromosome is an important factor that affects the performance of the GA because it determines the method of encoding, decoding, and genetic operations. In general, chromosomes consist of decision variables of the problem to be solved. Since the determinants of the TRSP are the truck sequences and product routings, the chromosome must be configured as them to apply GA to the TRSP. However, the large chromosome, composed of various types of variables, reduces the efficiency of GA by

frequently generating infeasible solutions during genetic operations and making it difficult to inherit dominant genes of parents.

We propose a new hybrid GA to overcome this drawback. The basic idea is that the TRSP will eventually be a combination of truck scheduling and product routing, so an effective approach is applied to them individually. The proposed hybrid GA combines GA and greedy heuristic. The truck sequences, which have a great effect on solution quality and are easy to express chromosomes, are optimized using GA, and the greedy heuristic is applied to product routing that is difficult to express chromosomes and consumes a lot of computation time. The detailed procedure is explained in the following sections.

4.1. Chromosome Representation and Population Structure.

In the TRSP, there are two sequences, the inbound truck sequence and the outbound truck sequence, which must be optimized simultaneously. As mentioned earlier, the two sequences are not independent because the cross-docking system seeks just-in-time product transfer between inbound and outbound trucks. The optimal outbound truck sequence is influenced by external customer requirements and the inbound truck sequence and the products on the truck. The optimal inbound truck sequence is affected by the outbound truck sequence. The two sequences are mutually dependent, but they cannot be mixed because they are separate sequences. Therefore, we propose a form of linked chromosomes as a chromosome representation for GA, as shown in Figure 2, where each gene represents a truck index and the sequence of genes represents a truck sequence.

In GA, the population is typically composed of a set of individuals, as shown in Figure 3(a). This structure, however, is not efficient for the TRSP. Because the TRSP has a very large solution space due to its high complexity, it is necessary to operate a very large population to obtain a good solution. The performance of a GA depends heavily on the balancing between exploitation and exploration. Large solution spaces make it difficult to balance them. To overcome this, we try to improve the performance of the GA by improving the structure of the population. In the TRSP, inbound truck sequences are dependent on the outbound truck sequence, so there are inbound truck sequences that result in excellent performance depending on the outbound truck sequence. Therefore, we propose the population of the nested structure as shown in Figure 3(b). The nested population structure can increase the exploitation capability of the inbound truck sequence according to the outbound truck sequence even with a population of the same size.

4.2. Crossover Operation. Crossover is a fundamental genetic operation that produces offspring from parents. An individual with high fitness is likely to be selected as a parent. Since the purpose of the TRSP under consideration is the minimization of makespan, the inverse of makespan is applied as a fitness function.

We propose a new crossover, namely, locus-pairing crossover, for the TRSP. The offspring generation procedure

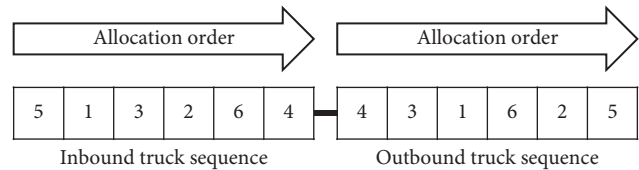


FIGURE 2: Chromosome representation of an individual in the proposed hybrid GA.

of the proposed crossover is identical to that of a typical two-point crossover. The difference is that linked chromosomes of inbound and outbound work separately but in conjunction. A typical two-point crossover consists of the following procedures; (1) select parents by applying a roulette wheel based on fitness values in the population; (2) create two random points within the range of the chromosome of the parent; (3) inherit genes between the selected two points to the same location of the chromosome of an offspring. The remaining genes in the offspring are inherited sequentially from the other parent; (4) create another offspring by performing subprocedure (3) on the loci not selected in subprocedure (2).

In the nested population structure, individual chromosomes are composed of inbound truck sequences and outbound truck sequences linked together. Since a truck sequence is a permutation, it must be crossover in only one sequence to avoid infeasibility. Therefore, to increase feasibility, crossover should be applied independently to the inbound truck sequence and outbound truck sequence. On the contrary, if the outbound truck sequence changes, the required product sequence also changes, so the inbound truck sequence for supplying it must also change. If it is close to the optimal solution, the inbound truck sequence and the outbound truck sequence will have a strong correlation with each other. In this state, if the two sequences change similarly, they are more likely to have good solutions. Based on this intuition, we propose a locus-pairing crossover that applies two-point crossover independently to the inbound truck sequence and the outbound truck sequence, but the same applies to the offspring-inherited loci. Figure 4 shows an example of creating two offspring from parents using a locus-pairing crossover. At this time, if the population size of the inbound truck sequence is n , crossover is performed between the i -th inbound truck sequence nested in each outbound truck sequence ($i = 1, 2, \dots, n$).

Locus-pairing crossover performs the crossover independently by applying the same loci to two independent sequences, which enhances the search while maintaining the correlation between the inbound truck sequence and the outbound truck sequence. Thus, even a small-sized population can efficiently pursue high solution quality.

4.3. Mutation Operation. Mutation is used to generate chromosomes that are not identical but very similar from existing chromosomes. We apply locus pairing to three typical mutations for insertion, swap, and inversion mutations. Figure 5 shows examples of three locus-pairing mutations. Pairing insertion mutation (PIM) is a mutation that

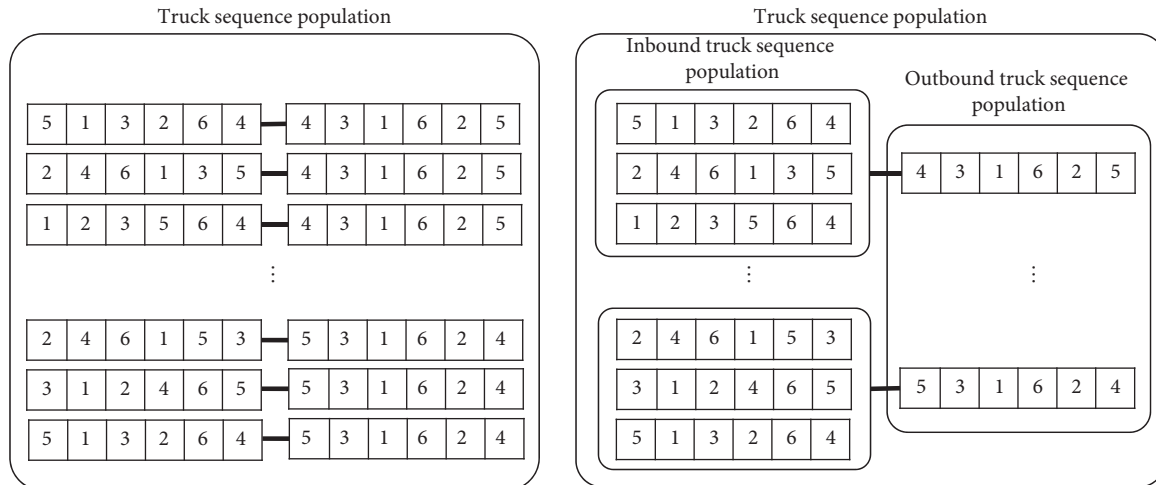


FIGURE 3: Comparison of population structures: (a) a typical structure of the population; (b) the proposed nested structure of the population for the proposed hybrid GA.

randomly selects two genes and inserts the second selected gene in front of the first selected gene. Pairing swap mutation (PSM) is a mutation that randomly selects two genes and swaps them. Finally, the pairing inversion mutation (PVM) is a mutation that randomly selects two genes and reverses their order. These locus-pairing mutations apply only to individuals in the truck sequence population. In the evolution of the inbound truck sequence population, the original mutation without locus pairing is applied.

4.4. Product Routing Heuristics. The inbound truck sequence and outbound truck sequence determine only the order in which inbound trucks and outbound trucks are assigned to receiving docks and shipping docks, respectively. Classification, partitioning, aggregation, and movement of products from inbound trucks to outbound trucks are another optimization problem. For example, suppose inbound truck I1, which contains one unit of product A and B, arrives at a receiving dock, and assume that outbound truck O1 and outbound truck O2 must load one unit of product A and B, respectively. Then, there are four ways to assign product A and product B from I1 to O1 and O2: loading all products to O1 only, loading all products to O2 only, product A to O1, and product B to O2, and vice versa. The makespan for each case will be different. The impact will be greater as the type and quantity of products increase. This optimization problem is called product routing. Since product routing also belongs to the NP-hard class, a simple and fast heuristic approach is generally applied.

Yu [3] proposed a sequence-based routing (SBR), a product routing heuristic for the TRSP that takes into account multiple docks. SBR is a greedy heuristic that sequentially assigns products required for outbound trucks from inbound trucks. At this time, the priority of outbound trucks and inbound trucks follows the outbound truck sequence and the inbound truck sequence, respectively. SBR is easy to apply and requires less computation because of its simple decision rules. On the contrary, if there is a large

discrepancy between the types and quantity of products between inbound and outbound trucks, the turnover of outbound trucks tends to decrease, and makespan increases. For this reason, we propose dock-based routing (DBR) that improves the shortcomings of SBR. The DBR preferentially assigns the products needed for the truck that can leave the docks as soon as possible among the outbound trucks in the shipping docks. DBR enables more efficient product routing than SBR by utilizing the characteristics of multiple docks, which allow multiple trucks to wait for product loading at the same time on shipping docks.

The procedure of DBR is as follows: (1) calculate the minimum required loading time (MLT) of products that have not yet been loaded for all outbound trucks that are standby in the shipping docks. (2) Select the truck with the smallest MLT. If there is a tie, the higher-rank truck is selected from the outbound truck sequence. (3) Find the inbound trucks that can send the products required by the selected outbound truck as soon as possible, and investigate the remaining quantity of each product. (4) Among them, send the required quantity of the product from the inbound truck with the smallest remaining quantity for each product to the outbound truck, and recalculate the MLT of the truck. (5) Repeat steps (1)–(4) until all products have been shipped.

A detailed step-by-step procedure is described with an example in the Appendix.

4.5. Overall Procedure of the Proposed Hybrid GA. Figure 6 shows the overall flowchart of the proposed hybrid GA proposed in this paper. Through the previous sections, we have already proposed a new population structure, crossover, mutation, and product routing heuristics to address TRSPs with multiple docks. All proposed subprocedures are combined and employed through the overall hybrid GA procedure in Figure 6. First, an initial population of outbound truck sequences is generated. At this time, chromosomes of all individuals are generated by random permutation of outbound trucks. Next, for each outbound truck sequence, a nested initial

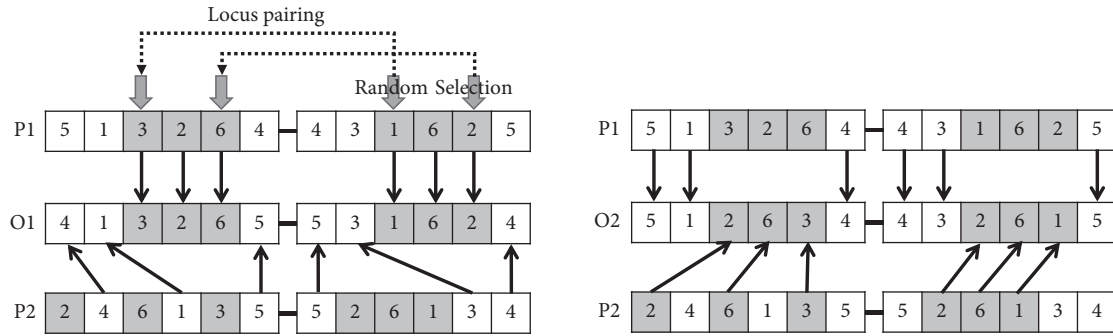


FIGURE 4: The crossover for the TRSP with multiple receiving and shipping docks.

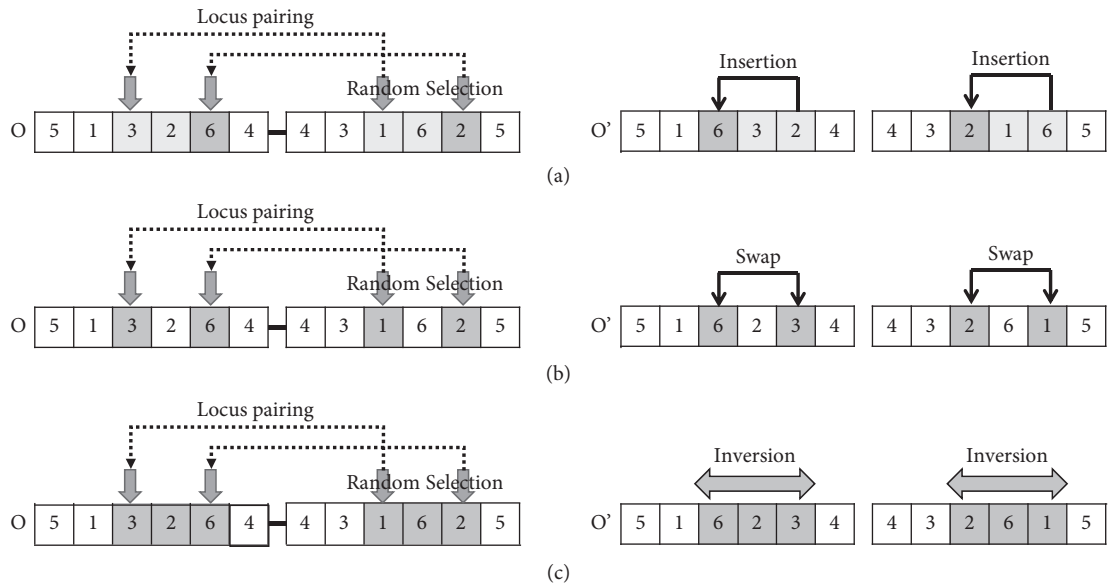


FIGURE 5: Three types of locus-pairing mutations: (a) pairing insertion mutation (PIM), (b) pairing swap mutation (PSM), and (c) pairing inversion mutation (PVM).

population of inbound truck sequences is also randomly generated. At this time, chromosomes of all individuals are also generated by random permutation of inbound trucks and linked to the corresponding outbound truck sequence. We will now refer to the linked chromosomes as individual truck sequences for convenience.

The routing of all products is determined by each individual truck sequence through the designated product routing heuristic. The makespan and fitness for each individual truck sequence are also calculated. Two offspring are generated through the locus-pairing crossover of the selected parents via the roulette wheel. Locus-pairing mutations are optionally applied to offspring. Additional independent evolutionary processes are performed through selection, crossover, and mutation for the nested population of inbound truck sequences. Due to elitism, the excellent inbound truck sequences are transferred to the next generation in a constant proportion. The termination condition of the evolution of inbound truck sequences is the number of generations. Once the evolution of the nested population is complete for all outbound truck sequences, the process returns to the evolution of the individual

truck sequence. Elitism also applies to this outer evolution. The termination condition is also the number of iterations.

5. Implementation and Computational Results

5.1. Benchmark Problems and Testing Environment. The proposed hybrid GA was verified against 20 benchmark problems [3] summarized in Table 1. The problem consists of a combination of docks, trucks, and products of various sizes. The biggest problem consists of eight receiving docks, eight shipping docks, 20 inbound trucks, 30 outbound trucks, 18 product types, and 8460 units of products. It is assumed that the changeover time (D) of the truck is 75 time units, and moving time (V) of the product is 100 time units [3]. The proposed hybrid GA was coded in Visual Basic 2017 and was run on Intel® Core™ i7-8799 CPU@ 3.20 GHz, 8 GB RAM.

5.2. Parameter Determination for the Hybrid GA. This section determines the design parameters of the proposed hybrid GA for the TRSP that affect solution quality and

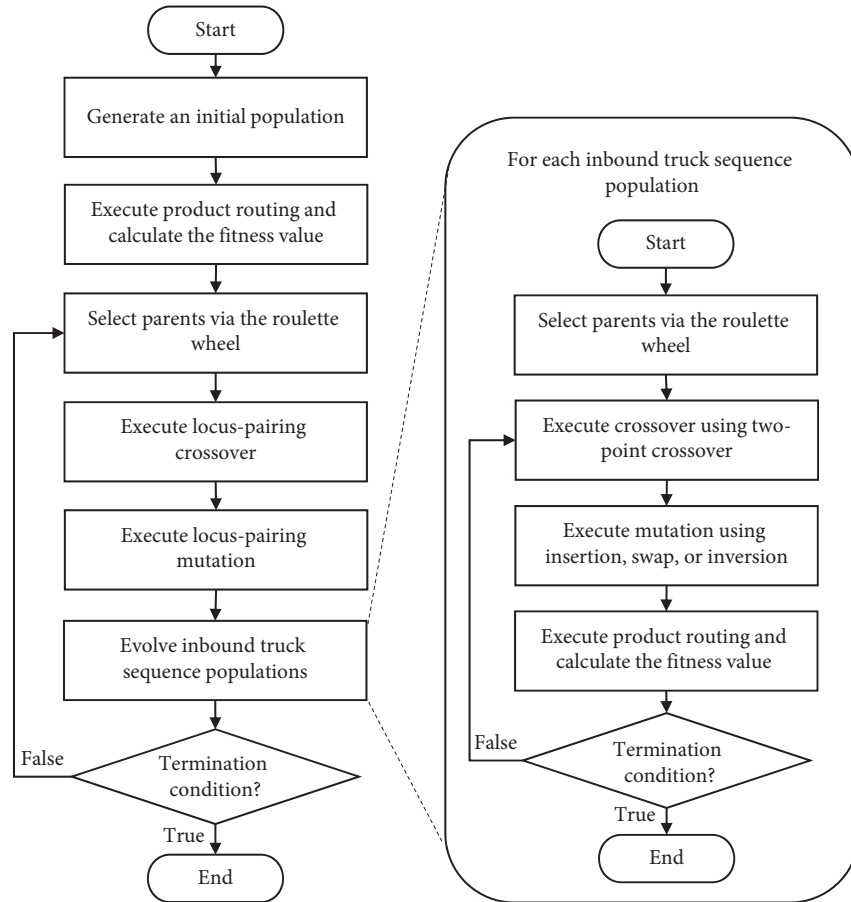


FIGURE 6: Overall procedure of the proposed hybrid GA for the TRSP.

computational performance. In addition, we evaluate the performance of the population structure, crossover, mutation, and product routing heuristic implemented in our proposed hybrid GA compared with conventional methods.

5.2.1. Default Parameter Settings for Performance Comparison. Evaluating our methods simultaneously is cumbersome, so we evaluate and determine them sequentially. First of all, the values of the basic design parameters of hybrid GA were selected for comparative evaluation of performance. The parameters considered and their default values are summarized in Table 2. Each default value is selected from the range normally applied in GA. The crossover rate and mutation rate refer to the probability that crossover and mutation operations will be performed, respectively. After generating a random number, if it is less than or equal to the rate value, each operation is executed. Elitism rate is the ratio of the superior individuals that transfer from one generation to the next. The evaluation criterion is fitness. Offspring insertion method determines the method of incorporating offspring generated in the current generation into the population of the next generation. There are two ways: competence and replacement. The competition constitutes the next generation of individuals with excellent fitness among the population of the current generation and the generated offspring. Replacement, on the

contrary, constructs all generated offspring into the next generation without contention. Product routing heuristic is selected from the SBR or DBR described in Section 4.4. The initial sequence is determined by random permutation for inbound trucks and outbound trucks, respectively. The final solution is determined to be the best solution through five repetitions.

5.2.2. Performance Verification of the Proposed Nested Structure of the Population. First, we verify the performance of the nested structure of the population. In order to compare the population of the independent conventional individual with the population of the proposed nested structure, the same-sized populations were constructed as shown in Table 3. The total number of individuals is 200. The nested population consists of 20 outbound truck sequences and 10 nested inbound truck sequences for each outbound truck sequence. The termination condition of the outer evolution is set to 200 iterations, and the inbound truck sequence evolution is set to 20 iterations. The other parameters are the same as the settings in Section 5.2.1. The performance of the algorithm was measured as the mean total makespan (MTM), which is the average of five experiments on the sum of makespans for 20 benchmark problems.

TABLE 1: Configuration of benchmark problems.

Problem	Number of receiving docks (R)	Number of shipping docks (S)	Number of inbound trucks (I)	Number of outbound trucks (O)	Number of product types (P)	Total number of products in the set
1	2	2	4	4	5	1020
2	2	2	4	5	4	990
3	2	2	5	5	8	1000
4	2	2	5	6	8	1610
5	2	2	6	6	9	2020
6	2	2	8	7	7	2300
7	2	2	7	8	8	2350
8	3	3	12	9	8	2370
9	3	3	7	10	12	3490
10	3	3	8	11	10	3780
11	3	3	10	12	9	3790
12	4	4	15	12	11	3120
13	4	4	9	13	8	4430
14	4	4	8	14	12	4260
15	5	5	15	15	10	3230
16	5	5	18	16	10	3220
17	6	6	22	20	12	6180
18	6	6	20	24	12	5740
19	8	8	18	26	15	6240
20	8	8	20	30	18	8490

MTM is 27,028.0 for the typical individual population and 26,724.0 for the nested population. The nested population provides a better solution for the TRSP in terms of MTM. Equal variances and t -test results for the two population types are shown in Table 4 and boxplot in Figure 7. In conclusion, the nested population outperforms the individual population.

5.2.3. Performance Verification of the Product Routing Heuristic. Next, we verify the performance difference of the product routing heuristics. Reflecting the results of Section 5.2.2, a comparison experiment was conducted using SBR or DBR only for product routing heuristics based on the hybrid GA using the nested population of Table 4. MTM is 26,724.0 when SBR is applied and 26,259.8 when DBR is applied. This difference is also significant for t -test as shown in Table 5 and boxplot in Figure 8. Therefore, we can conclude that DBR is better than SBR.

5.2.4. Determination of the Offspring Insertion Method. In this section, we decide how to transfer the generated offspring to the next generation. For the experiment, the hybrid GA was constructed by applying the nested population and DBR to the basic parameter set in consideration of the previous experimental results. The competition and replacement were applied alternately and then evaluated. The MTM is 26,259.8 for the competition and 26,202.4 for the replacement. As Table 6 shows, it is statistically certain that replacement is superior to competition at the significance level of 0.05 in terms of MTM. The boxplot in Figure 9 also shows the difference between competition and replacement.

5.2.5. Determination of Operational Parameters. In order to reduce the number of experiments, the experiments have been performed so far with the operating parameters fixed at

TABLE 2: Default parameters for performance comparison.

Parameter	Value
Crossover rate	0.7
Mutation rate	0.3
Elitism rate	0.1
Offspring insertion method	Competition
Product routing heuristic	SBR
Initial sequence generation method	Random permutation
Number of repetitions	5

the default values. However, operating parameters such as crossover rate, mutation rate, and elitism rate also have a significant effect on the performance of hybrid GA. In this section, the optimal operating parameters are determined for the hybrid GA with the nested population, DBR, and replacement offspring insertion policy. The levels of each parameter evaluated are as follows: crossover rate is (0.5, 0.7, 0.9), mutation rate is (0.1, 0.3, 0.5), and elitism rate is (0.1, 0.2). ANOVA was performed to determine the significance of the factors except the interaction effect of the three factors. ANOVA table is summarized in Table 7. Residual plots, main effect plots, and interaction effect plots are shown in Figures 10–12.

All main effects and all interactions except the interaction of crossover rate and elitism rate were statistically significant at the significance level of 0.05. Pooling was not performed because all other factors were significant. Residual plots in Figure 10 show that ANOVA is valid because there are no singularities in all subplots. Since interactions are significant, the optimal combination of parameters was chosen from the interaction plots in Figure 12; the crossover rate is 0.7, the mutation rate is 0.5, and the elitism rate is 0.2. This combination is the same as the optimal combination of the main effects in Figure 11.

TABLE 3: Basic parameter set for the GA and the proposed hybrid GA.

Parameter	Value	
	Individual population	Nested population
Population size of the outbound truck sequence	200	20
Population size of the inbound truck sequence	—	10
Termination condition for the individual truck sequence	200 iterations	200 iterations
Termination condition for the nested population	—	20 iterations

TABLE 4: Statistical test result for the type of population.

Test ($\alpha = 0.05$)	Hypothesis	T	p value
Equal variances (Levene's test)	H_0 : all variances are equal	—	0.782
	H_1 : at least nine variances are different		
t	H_0 : $\mu_{\text{Individual Pop.}} = \mu_{\text{nested Pop.}}$	7.01	<0.001
	H_1 : $\mu_{\text{Individual Pop.}} > \mu_{\text{nested Pop.}}$		

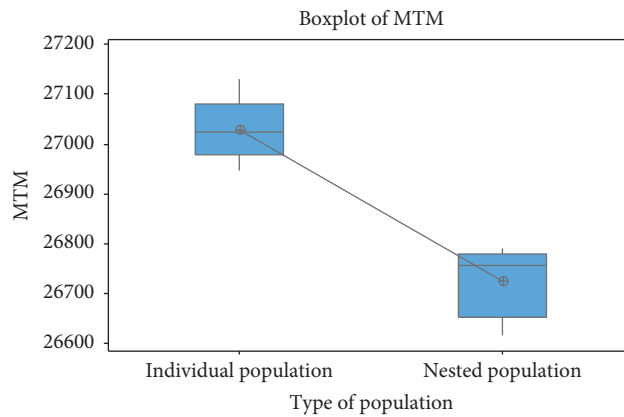


FIGURE 7: Boxplot of MTM for the type of population.

TABLE 5: Statistical test result for the product routing heuristic.

Test ($\alpha = 0.05$)	Hypothesis	T	p value
Equal variances (Levene's test)	H_0 : all variances are equal	—	0.625
	H_1 : at least one variance is different		
t	H_0 : $\mu_{\text{SBR}} = \mu_{\text{DBR}}$	12.04	<0.001
	H_1 : $\mu_{\text{SBR}} > \mu_{\text{DBR}}$		

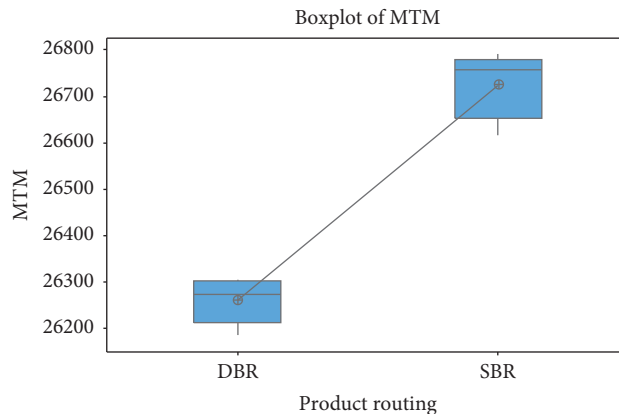


FIGURE 8: Boxplot of MTM for the product routing heuristic.

TABLE 6: Statistical test result for the offspring insertion method.

Test ($\alpha = 0.05$)	Hypothesis	T	p value
Equal variances (Levene's test)	H_0 : all variances are equal	—	0.233
	H_1 : at least nine variances are different		
t	H_0 : $\mu_{\text{competition}} = \mu_{\text{replacement}}$	2.37	0.023
	H_1 : $\mu_{\text{competition}} > \mu_{\text{replacement}}$		

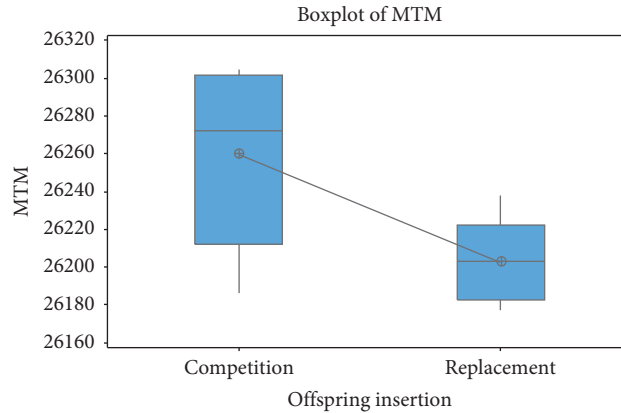


FIGURE 9: Boxplot of MTM for the offspring insertion method.

TABLE 7: ANOVA table for operational parameters.

Source of variation	Degree of freedom	Sum of squares	Mean square	F	p value
Crossover rate	2	13,284	6642	3.48	0.036
Mutation rate	2	170,600	85,300	44.71	<0.001
Elitism rate	1	12,912	12,912	6.77	0.011
Crossover rate * mutation rate	4	30,833	7708	4.04	0.005
Crossover rate * elitism rate	2	9830	4915	2.58	0.083
Mutation rate * elitism rate	2	28,630	14,315	7.50	0.001
Error	76	145,011	1908		
Total	89	411,100			

5.3. *Performance Comparison with Existing Methods.* Finally, we verify the superiority of the proposed hybrid GA. For comparison, two existing heuristics for the TRSP, NS and TNLSM by Yu [3], were considered. The parameters of the hybrid GAs applied in the comparative experiments are summarized in Table 8. The population size of the outbound truck sequence was increased to 80 for enough iterations.

The measure of performance is the percent deviation rate from the lower bound of the following equation:

$$\frac{C_{\max,l} - C_{\max,l}^{lb}}{C_{\max,l}^{lb}} \times 100, \quad (1)$$

where $C_{\max,l}$ denotes the makespan for the l th problem and $C_{\max,l}^{lb}$ is the lower bound of the l th problem. The experimental results of NS, TNLSM, and hybrid GA are summarized in Table 9. $C_{\max,l}$ of NS and TNLSM is the best of 100 repetitions, and hybrid GA is the result of 5 repetitions. The best results among the three heuristics are shown in bold. As shown in Table 9, hybrid GA provides the best solutions among 20 benchmark problems except for

problems 1, 9, 11, 13, and 15. Problems 1, 9, 11, and 13 differ only by one time unit, so hybrid GA is superior to almost all benchmark problems. The improved rate represents the degree of improvement of the hybrid GA solution compared to the best solution among the existing heuristics. The average improvement rate is 9.76%, and the maximum reaches 78.16%. The hybrid GA solves the problems in a reasonable amount of time. Usually, the hybrid GA solves the problem between 0 seconds and 10 minutes. In the worst case, when the hybrid GA solves problem 20, the average computation time is less than 60 minutes.

ANOVA was performed to statistically verify the difference in performance between heuristics, where heuristic is the fixed factor and problem is the random factor by applying randomized block design. The results are presented in the ANOVA table in Table 10 and the residual plots in Figure 13. The difference between heuristics was significant at the significance level of 0.05. The adjusted R -squared value is also 85.90%, and there is no unusual pattern in the residual plots. Based on the above results, it can be concluded that hybrid GA outperforms existing NS and TNLSM.

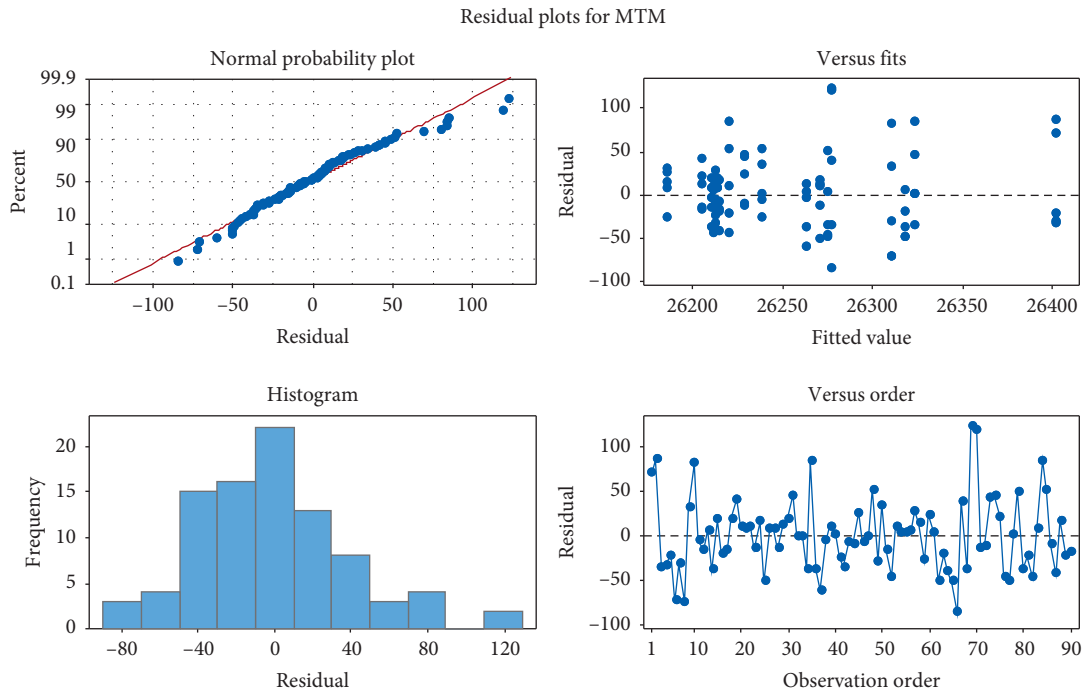


FIGURE 10: Residual plot of ANOVA for operational parameters.

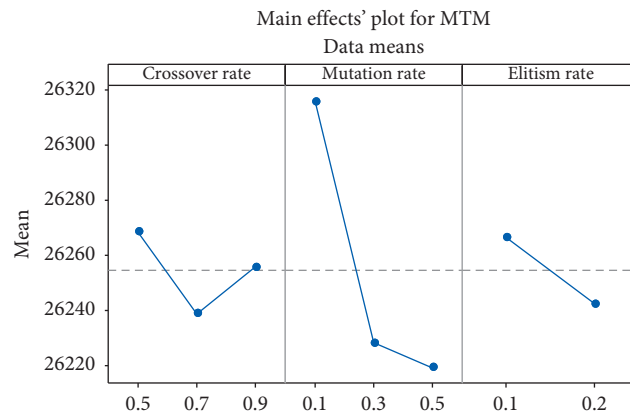


FIGURE 11: Main effect plot for operational parameters.

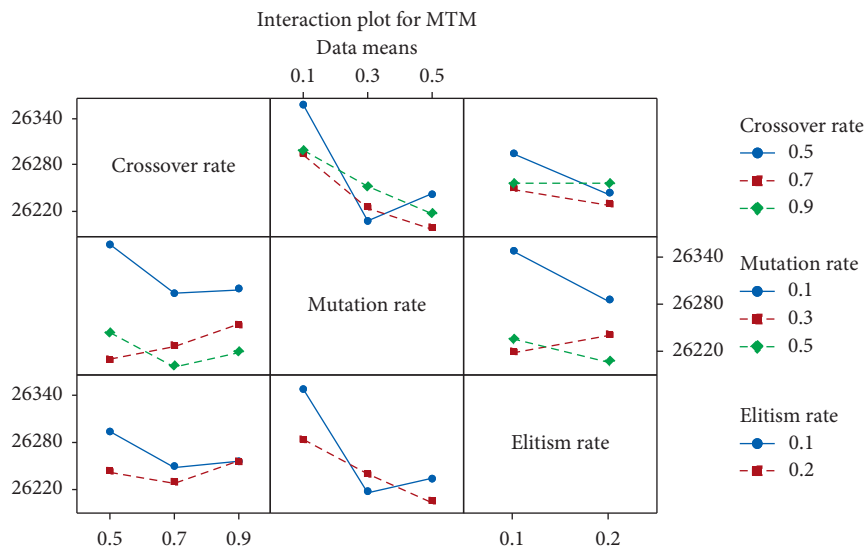


FIGURE 12: Interaction effect plot between operational parameters.

TABLE 8: Best parameter set for the proposed hybrid GA.

Parameter	Value
Population size of the outbound truck sequence	80
Population size of the inbound truck sequence	10
Termination condition for the individual truck sequence	200 iterations
Termination condition for the nested population	20 iterations
Crossover rate	0.7
Mutation rate	0.5
Elitism rate	0.2
Offspring insertion method	Replacement
Product routing heuristic	DBR
Initial sequence generation method	Random permutation
Number of repetitions	5

TABLE 9: Comparison of TRSP heuristics for 20 benchmark problems.

Problem (l)	$C_{max,l}^{lb}$	$C_{max,l}$			Percent deviation rate from the lower bound (%)					
		NS Best	TNLSM Best	Hybrid GA Best	NS Best	TNLSM Best	Hybrid GA Best	Hybrid GA Mean	Improved rate	
1	685	778	778	779	779.0	13.58	13.58	13.72	13.72	-1.08
2	708	757	759	753	753.0	6.92	7.20	6.36	6.36	8.16
3	713	829	829	818	818.0	16.27	16.27	14.73	14.73	9.48
4	1055	1182	1182	1181	1181.0	12.04	12.04	11.94	11.94	0.79
5	1260	1420	1336	1335	1335.0	12.70	6.03	5.95	5.95	1.32
6	1475	1677	1689	1586	1593.0	13.69	14.51	7.53	8.00	45.05
7	1500	1665	1665	1619	1623.6	11.00	11.00	7.93	8.24	27.88
8	1115	1211	1202	1134	1137.0	8.61	7.80	1.70	1.97	78.16
9	1439	1617	1596	1597	1598.2	12.37	10.91	10.98	11.06	-0.64
10	1560	1695	1712	1658	1664.8	8.65	9.74	6.28	6.72	27.41
11	1589	1738	1776	1739	1739.0	9.38	11.77	9.44	9.44	-0.67
12	1087	1157	1145	1123	1133.0	6.44	5.34	3.31	4.23	37.93
13	1377	1659	1659	1660	1660.0	20.48	20.48	20.55	20.55	-0.35
14	1353	1534	1512	1501	1502.0	13.38	11.75	10.94	11.01	6.92
15	896	968	972	974	981.2	8.04	8.48	8.71	9.51	-8.33
16	939	1041	1026	1016	1016.0	10.86	9.27	8.20	8.20	11.49
17	1330	1489	1533	1487	1496.0	11.95	15.26	11.80	12.48	1.26
18	1282	1417	1438	1416	1418.6	10.53	12.17	10.45	10.66	0.74
19	1049	1232	1230	1229	1229.0	17.45	17.25	17.16	17.16	0.55
20	1369	1502	1494	1486	1486.4	9.72	9.13	8.55	8.58	6.40
Sum	23781	26568	26533	26091	26144	234.05	229.98	196.24	200.51	195.22
Avg.	1189.05	1328.4	1326.7	1304.6	1307.2	11.70	11.50	9.81	10.03	9.76

TABLE 10: ANOVA table for the best percent deviation rate from the lower bound of heuristics.

Source of variation	Degree of freedom	Sum of squares	Mean square	F	p value
Heuristic (fixed factor)	2	43.10	21.55	9.49	<0.001
Problem (random factor)	19	821.53	43.24	19.03	<0.001
Error	38	86.33	2.27		
Total	59	950.96			

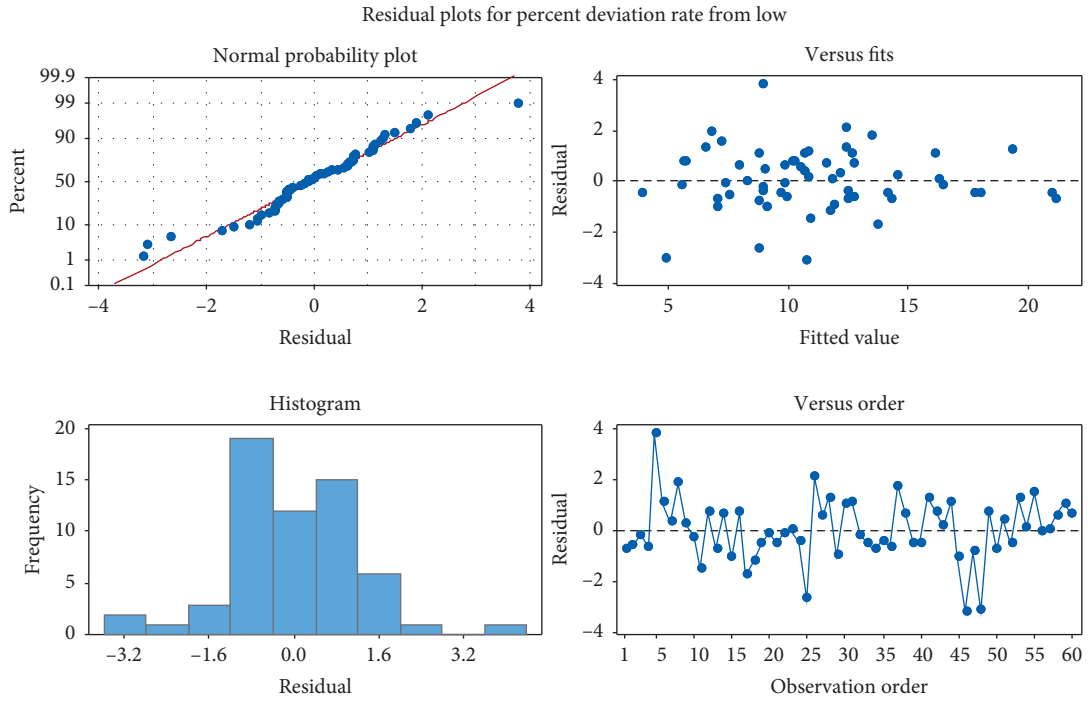


FIGURE 13: Residual plots for percent deviation rate from lower bound of heuristics.

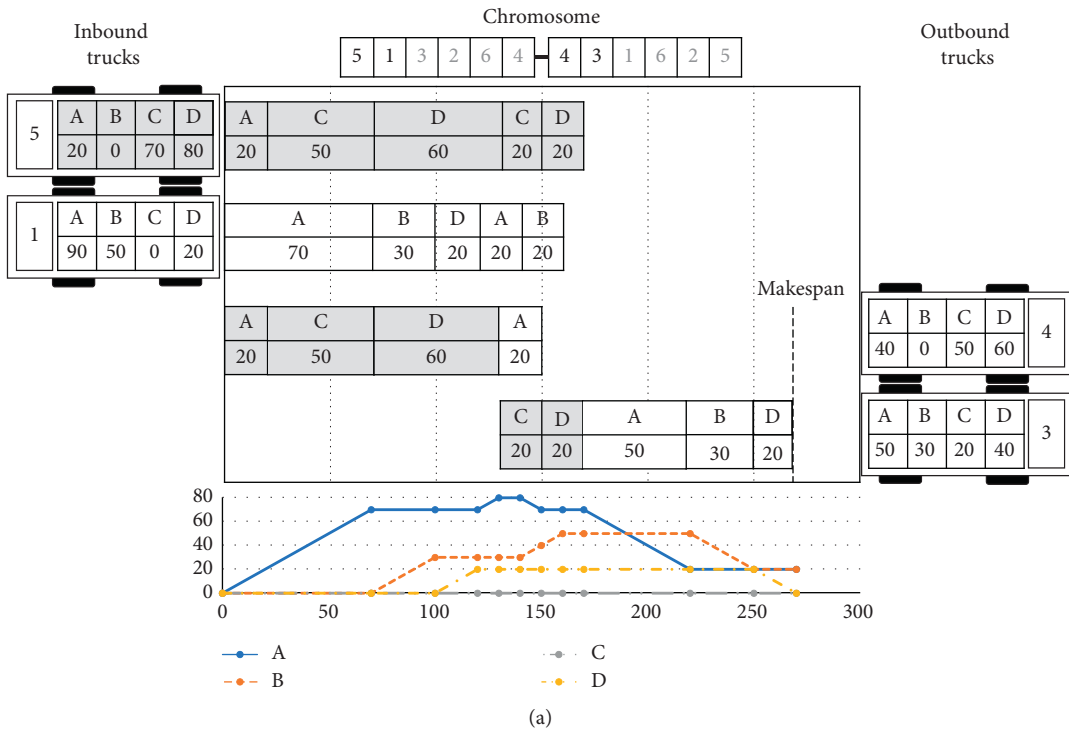


FIGURE 14: Continued.

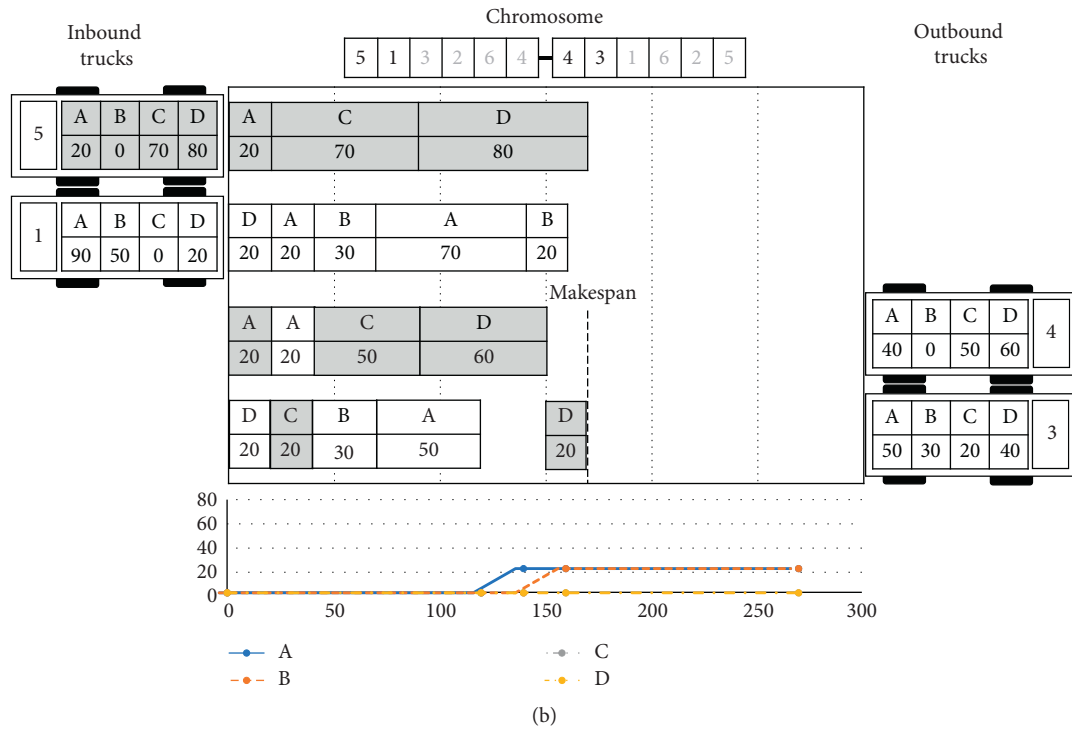


FIGURE 14: Example of product routing rules: (a) sequence-based routing and (b) docking-based routing.

6. Conclusions

TRSP is a problem that must be solved for operational efficiency of a cross-docking system. The existing TRSP is different from the realistic cross-docking system by mainly considering simple configuration. This paper addresses a more realistic problem by dealing with the TRSP for the cross-docking system with multiple receiving and shipping docks and temporary storage. As the configuration of the cross-docking system becomes more realistic, the TRSP becomes more complicated and requires a new approach. Our proposed hybrid GA can be a solution in this situation. The proposed hybrid GA is designed to utilize the properties of the cross-docking system considered as much as possible. In addition to improving the performance of the GA by applying the linked-chromosome representation, the nested population scheme, and the locus-pairing genetic operations, the performance of the product routing is also improved through the proposed dock-based product routing heuristic. In the future, we will expand our research into the TRSP for mixed-mode doors that can be switched between receiving docks and shipping docks, multiobjective TRSP capable of addressing a variety of objectives, and TRSP that simultaneously solves scheduling of inbound and outbound trucks.

Appendix

Figure 14 shows the example results of product routing by sequence-based routing [3] and docking-based routing. In this example, the inbound truck sequence is 5-1, and the

outbound truck sequence is 4-3. The two inbound trucks are marked with the type and number of products they are currently loading, and the two outbound trucks are marked with the type and number of products they have to load. The loading and unloading order of each item is represented by a Gantt chart of the truck location. The graph below the Gantt chart shows the quantity of each product stored in temporary storage. Docking-based routing is more complicated than sequence-based routing, but because routing takes into account the current dock state, it can reduce the makespan than sequence-based routing as shown in Figure 7.

In the case of sequence-based routing, the products needed for truck 4 are assigned to truck 5 first, and then the products not assigned to truck 5 are assigned to 3. Then, the products needed for truck 1 are assigned from truck 5 and then from truck 1. Therefore, the product is assigned as follows:

- (1) From truck 5 to truck 4, 20 products (product A)
- (2) From truck 5 to truck 4, 20 products (product C)
- (3) From truck 5 to truck 4, 60 products (product D)
- (4) From truck 1 to truck 4, 20 products (product A)
- (5) From truck 5 to truck 3, 20 products (product C)
- (6) From truck 5 to truck 3, 20 product (product D)
- (7) From truck 1 to truck 3, 20 products (product A)
- (8) From truck 1 to truck 3, 30 products (product B)
- (9) From truck 1 to truck 3, 20 products (product D)

Docking-based routing selects inbound trucks and their products, taking into account the minimum required

loading time (MLT) and the quantity of products. The example of docking-based routing is as follows:

- (1) Because both MLTs of trucks 3 and 4 are zero, truck 4 is selected based on outbound truck sequence 4-3.
- (2) Truck 4 can receive 20 products (product A), 50 products (product C), or 60 products (product D) from truck 5 and 40 products (product A), 50 products (product B), or 20 products (product D) from truck 1.
- (3) The smallest numbers of products to send truck 4 from truck 5 and truck 1 are the same (20 products (product A) from truck 5 and 20 products (product D) from truck 1). Therefore, 20 products (product A) of truck 5 are selected to send truck 4 based on outbound truck sequence 5-1. Then, the MLT of truck 4 becomes 20.
- (4) Truck 3 is selected by the MLTs. Then, truck 1 can send 50 products (product A), 30 products (product B), or 20 products (product D). Truck 5 cannot send its products to truck 3 because truck 5 sends its products to truck 4 until time 20.
- (5) Since the smallest number of products to send truck 3 from truck 1 is 20 (product D), truck 1 sends 20 products (product D) to truck 3. Then, the MLT of truck 3 is 20.
- (6) Because both MLTs of trucks 3 and 4 are 20, truck 4 is selected based on outbound truck sequence 4-3. Truck 4 can receive 20 products (product A) from truck 1 and 50 products (product C) or 60 products (product D) from truck 5.
- (7) Since the smallest number of products to send truck 4 is 20 products (product A) from truck 1, truck 1 sends 20 products (product A) to truck 4. Then, the MLT of truck 4 is 40.
- (8) Truck 3 (MLT of truck 3=20) can receive 20 products (product C) or 20 products (product D) from truck 5. Since the smallest number of products to send truck 3 is 20 products (product C) from truck 5, truck 5 sends 20 products (product C) to truck 3. Then, the MLT of truck 3 is 40.
- (9) Truck 4 (MLT of truck 4=40) can receive 50 products (product C) or 60 products (product D) from truck 5. Since the smallest number of products to send truck 4 is 50 products (product C) from truck 5, truck 5 sends 50 products (product C) to truck 4. Then, the MLT of truck 4 is 90.
- (10) Truck 3 (MLT of truck 3=40) can receive 50 products (product A) or 30 products (product B) from truck 1. Since the smallest number of products to send truck 3 is 30 products (product B) from truck 1, truck 1 sends 30 products (product B) to truck 3. Then, the MLT of truck 3 is 70.
- (11) Truck 3 (MLT of truck 3=70) can receive 30 products (product B) from truck 1. Truck 1 sends 30 products (product B) to truck 3. Then, the MLT of truck 3 is 120.

- (12) Truck 4 (MLT of truck 4=90) can receive 60 products (product D) from truck 5. Truck 5 sends 60 products (product D) to truck 4. Then, the departure time of truck 4 is 150.
- (13) Truck 3 (MLT of truck 3=120) can receive 20 products (product D) from truck 5. Truck 5 sends 20 products (product D) to truck 2 after time 150 since truck 5 should unload its products for truck 4 until time 150. Then, the departure time of truck 3 is 170.

Data Availability

The 20 benchmark problems, experiment results, and any other data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

References

- [1] N. Boysen, M. Flidner, and A. Scholl, "Scheduling inbound and outbound trucks at cross docking terminals," *OR Spectrum*, vol. 32, no. 1, pp. 135–161, 2010.
- [2] A.-L. Ladier and G. Alpan, "Cross-docking operations: current research versus industry practice," *Omega*, vol. 62, pp. 145–162, 2016.
- [3] 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.
- [4] A. Rijal, M. Bijvank, and R. de Koster, "Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors," *European Journal of Operational Research*, vol. 278, no. 3, pp. 752–771, 2019.
- [5] P. Bodnar, R. de Koster, and K. Azadeh, "Scheduling trucks in a cross-dock with mixed service mode dock doors," *Transportation Science*, vol. 51, no. 1, pp. 112–131, 2017.
- [6] S. S. Fard and B. Vahdani, "Assignment and scheduling trucks in cross-docking system with energy consumption consideration and trucks queuing," *Journal of Cleaner Production*, vol. 213, pp. 21–41, 2019.
- [7] 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.
- [8] A. Golshahi-Roudbaneh, M. Hajiaghahi-Keshteli, and M. M. Paydar, "Developing a lower bound and strong heuristics for a truck scheduling problem in a cross-docking center," *Knowledge-Based Systems*, vol. 129, pp. 17–38, 2017.
- [9] H. Mohammadzadeh, N. Sahebjamnia, A. M. Fathollahi-Fard, and M. Hajiaghahi-Keshteli, "New approaches in metaheuristics to solve the truck scheduling problem in a cross-docking center," *International Journal of Engineering, Transaction B: Applications*, vol. 31, pp. 1258–1266, 2018.
- [10] A. M. Fathollahi-Fard, M. Ranjbar-Bourani, N. Cheikhrouhou, and M. Hajiaghahi-Keshteli, "Novel modifications of social engineering optimizer to solve a truck scheduling problem in a

- cross-docking system,” *Computers & Industrial Engineering*, vol. 137, Article ID 106103, 2019.
- [11] B. Vahdani and M. Zandieh, “Scheduling trucks in cross-docking systems: robust meta-heuristics,” *Computers & Industrial Engineering*, vol. 58, no. 1, pp. 12–24, 2010.
- [12] A. B. Arabani, S. F. 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, pp. 1964–1979, 2011.
- [13] T. W. Liao, P. J. Egbelu, and P. C. Chang, “Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations,” *Applied Soft Computing*, vol. 12, no. 11, pp. 3683–3697, 2012.
- [14] B. Vahdani, R. Tavakkoli-Moghaddam, and S. M. Mousavi, “Scheduling of trucks in cross-docking systems: a hybrid meta-heuristic algorithm,” *Lecture Notes on Management Science*, vol. 5, pp. 125–132, 2013.
- [15] A. L. Shigumoto, U. S. Cavalcante Netto, and G. H. S. Bauab, “An efficient hybrid meta-heuristic for a cross-docking system with temporary storage,” *International Journal of Production Research*, vol. 52, no. 4, pp. 1231–1239, 2014.
- [16] R. Soltani and S. J. Sadjadi, “Scheduling trucks in cross-docking systems: a robust meta-heuristics approach,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 46, no. 5, pp. 650–666, 2010.
- [17] K. L. Choy, H. K. H. Chow, T. C. Poon, and G. T. S. Ho, “Cross-dock job assignment problem in space-constrained industrial logistics distribution hubs with a single docking zone,” *International Journal of Production Research*, vol. 50, no. 9, pp. 2439–2450, 2012.
- [18] R. Sadykov, “Scheduling incoming and outgoing trucks at cross docking terminals to minimize the storage cost,” *Annals of Operations Research*, vol. 201, no. 1, pp. 423–440, 2012.
- [19] N. Boysen, “Truck scheduling at zero-inventory cross docking terminals,” *Computers & Operations Research*, vol. 37, no. 1, pp. 32–41, 2010.
- [20] A. B. Arabani, M. Zandieh, and S. F. Ghomi, “A cross-docking scheduling problem with sub-population multi-objective algorithms,” *The International Journal of Advanced Manufacturing Technology*, vol. 58, pp. 741–761, 2012.
- [21] M. H. Fazel Zarandi, H. Khorshidian, and M. Akbarpour Shirazi, “A constraint programming model for the scheduling of JIT cross-docking systems with preemption,” *Journal of Intelligent Manufacturing*, vol. 27, no. 2, pp. 297–313, 2016.
- [22] B. Naderi, S. Rahmani, and S. Rahmani, “A multiobjective iterated greedy algorithm for truck scheduling in cross-dock problems,” *Journal of Industrial Engineering*, vol. 2014, Article ID 128542, 12 pages, 2014.
- [23] D. Konur and M. M. Goliias, “Analysis of different approaches to cross-dock truck scheduling with truck arrival time uncertainty,” *Computers & Industrial Engineering*, vol. 65, no. 4, pp. 663–672, 2013.
- [24] 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.
- [25] T. W. Liao, P. J. Egbelu, and P. C. Chang, “Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations,” *International Journal of Production Economics*, vol. 141, no. 1, pp. 212–229, 2013.
- [26] Y. Guo, Z. R. Chen, Y. L. Ruan, and J. Zhang, “Application of NSGA-II with local search to multi-dock cross-docking scheduling problem,” in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics, 2012*, pp. 779–784, Seoul, Korea, October 2012.
- [27] 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.
- [28] J. Van Belle, P. Valckenaers, G. Vanden 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.
- [29] C. M. Joo and B. S. Kim, “Scheduling compound trucks in multi-door cross-docking terminals,” *The International Journal of Advanced Manufacturing Technology*, vol. 64, no. 5–8, pp. 977–988, 2013.
- [30] Y. Kuo, “Optimizing truck sequencing and truck dock assignment in a cross docking system,” *Expert Systems with Applications*, vol. 40, no. 14, pp. 5532–5541, 2013.
- [31] 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.
- [32] M. Yazdani, B. Naderi, and M. Mousakhani, “A model and metaheuristic for truck scheduling in multi-door cross-dock problems,” *Intelligent Automation & Soft Computing*, vol. 21, no. 4, pp. 633–644, 2015.
- [33] N. Boysen and M. Flidner, “Cross dock scheduling: classification, literature review and research agenda,” *Omega*, vol. 38, no. 6, pp. 413–422, 2010.
- [34] J. Van Belle, P. Valckenaers, and D. Cattrysse, “Cross-docking: state of the art,” *Omega*, vol. 40, no. 6, pp. 827–846, 2012.
- [35] J. H. Holland, *Adaptation in Natural and Artificial System: An Introduction with Application to Biology, Control and Artificial Intelligence*, MIT Press, Cambridge, MA, USA, 1975.