

A Multi-Objective Differential Evolution for Just-In-Time Door Assignment and Truck Scheduling in Multi-door Cross Docking Problems

Warisa Wisittipanich*, Piya Hengmeechai

Excellence Centre in Logistics and Supply Chain Management Chiang Mai University, Chiang Mai, Thailand

(Received: July 21, 2015 / Accepted: September 7, 2015)

ABSTRACT

Nowadays, the distribution centres aim to reduce costs by reducing inventory and timely shipment. Cross docking is a logistics strategy in which products delivered to a distribution centre by inbound trucks are directly unloaded and transferred to outbound trucks with minimum warehouse storage. Moreover, on-time delivery in a distribution network becomes very crucial especially when several distribution centres and customers are involved. Therefore, an efficient truck scheduling is needed to synchronize the delivery throughout the network in order to satisfy all stakeholders. This paper presents a mathematical model of a mixed integer programming for door assignment and truck scheduling in a multiple inbound and outbound doors cross docking problem according to Just-In-Time concept. The objective is to find the schedule of transshipment operations to simultaneously minimize the total earliness and total tardiness of trucks. Then, a multi-objective differential evolution (MODE) is proposed with an encoding scheme and four decoding strategies, called ITSH, ITDD, OTSH and OTDD, to find a Pareto frontier for the multi-door cross docking problems. The performances of MODE are evaluated using 15 generated instances. The numerical experiments demonstrate that the proposed algorithm is capable of finding a set of diverse and high quality non-dominated solutions.

Keywords: Cross Docking, Door Assignment, Truck Scheduling, Just-in-time, Differential Evolution

* Corresponding Author, E-mail: warisa.o@gmail.com

1. INTRODUCTION

As competitive market environment has aimed to rapidly supply products from suppliers to customers, the logistics companies must continuously develop new strategies to reduce cost and also satisfy customer requirements. Typically, operations of a distribution centre consist of five basic functions: receiving, sorting, storing, retrieving, and shipping. Among these functions, the most time consuming activities are storing and retrieving. Cross docking is one approach of logistics strategies that is currently adopted by many companies to eliminate these two expensive functions; storing and retrieving. The basic

idea behind cross docking is to transfer incoming products directly to outgoing vehicles without storing them in between (Van Belle *et al.*, 2012).

Cross docking is initially designed for the situation of customers ordering small volume of products at the same time and demanding a more accurate and timely delivery. In a cross docking system, products delivered by incoming (inbound) vehicles are immediately unloaded, sorted, routed, and loaded to outgoing (outbound) vehicles for on time delivery to customers. Thus, a little or no inventory remains at the centre. If the shipments are temporarily stored, it must be hold for a short time. Thus, cross docking coincides with the goals of lean supply

chain managements. The cross docking has a several advantages such as the consolidation of shipments, a shorter delivery lead time, the faster inventory turnover, the reduction of operation costs, and the increase of service levels. Since, the focus of cross dock is on transshipping, not holding inventory, an efficient cross docking system requires a correct synchronization of inbound and outbound vehicles. A synchronization of vehicle flows in cross docking is very complicated and challenging. Several important elements must be carefully taken into considerations; for example, products should be delivered with a short lead time, the arrival and departure schedules of trucks should be specified in advance, and a system of both software and hardware are well prepared.

Currently, Just-in-Time (JIT) concept becomes more and more important in several applications. In cross docking system, one approach to harmonize the flow of transshipments is to determine the arrival time and departure time of trucks in advance. Then, an efficient truck scheduling plan is required to synchronize the distributions throughout the network and guarantee that all distributions can be made at predetermined delivery times as much as possible. This on-time delivery becomes very important especially in a distribution network consisting of several cross docking centers and customers.

This paper studies the problem of door assignment and truck scheduling in cross docking terminals with multiple inbound doors and outbound doors according to the JIT concept. Since both door assignment and truck scheduling problem are NP-hard, its combination is even more difficult to solve. In practical, to deal with the problem complexities, this study presents an implementation of a metaheuristic, named multi-objective differential evolution (MODE), with an encoding and decoding schemes in order to find a Pareto frontier for the problems. The objective is to find the schedule of transshipment operations in order to simultaneously minimize the total earliness and the total tardiness of trucks.

The paper is organized as follows. Section 2 summarizes some of literatures on door assignment and truck scheduling problems in cross docking system. The theory of Differential Evolution (DE) algorithm is also provided in this section. Section 3 provides problem descriptions with a mathematical model used in this study. Section 4 describes the proposed MODE algorithm with its application to the cross docking problem. Experimental results are reported in Section 5. Lastly, in section 6, conclusions and future works are shortly discussed.

2. LITERATURE REVIEW

One of the comprehensive reviews on cross docking problems was given by Van Belle *et al.* in 2012. In this study, an overview of cross docking concept and the guidelines for successful implementation of cross docking were discussed. Typically, there are several decisions to be made in cross docking operations. The first

decision is in the planning process in which strategic decisions about cross dock locations and cross dock layout need to be made. Second, the tactical decisions are about the cross docking networks and how well the products are distributed through the network in order to optimize certain objectives. Finally, the operational decisions; for example, the vehicle routing, the assignment of trucks or truck scheduling, and where the products will be temporarily stored, have to be considered. Moreover, other decisions not directly concerned in cross docking operations such as a shape of cross dock, the number of doors and the arrival pattern of trucks should be also taken into account to make an efficient cross docking system.

Door assignment and truck scheduling are one of crucial decisions at the cross docking terminals. Simultaneously scheduling of inbound and outbound trucks enables harmonization of incoming flows and outbound flows of the products so that the Just-in-Time (JIT) supply of products can be attained. The door assignment and truck scheduling problems with multiple inbound and outbound doors problem are proven as NP-hard problem which can take extremely long computing time to find the optimal solution by the traditional exact methods. Therefore, for practical purpose, heuristics and metaheuristics algorithms are more preferable as they do provide high-quality or near-optimal solutions within acceptable computing times.

Most of research works on cross docking operations discuss on door assignment and truck scheduling problems separately. For the door assignment problem, earlier studies focused on the exact solution approaches. Tsui and Chang (1990) applied a bi-linear program to determine the allocation of trucks to inbound door and outbound door in order to minimize the total material handling effort. Later, Tsui and Chang (1992) solved the quadratic door assignment problem by branch and bound method. Bartholdi and Gue (2000) proposed a non-linear model to minimize the total labor cost, subject to an additional door pressure constraint. Due to the fact that the door assignment problem is NP-hard, more recent researches have been devoted to the development of heuristics and metaheuristics. Cohen and Keren (2009) extended the approach of Tsui and Chang (1990) in which the mathematical model was adapted in which the capacity of the outbound trucks was taken into consideration. Then, they proposed a heuristic algorithm to solve the problem, and the results showed that the heuristic provided good performances.

For the truck scheduling problems, Yu and Egbelu (2008) considered the truck scheduling problem with a single receiving door and a single shipping door. They presented the mixed integer programming (MIP) model and proposed the heuristic algorithm to minimize the total operation time (makespan). The experimental results indicated that solutions obtained from the heuristics are very close to optimal solutions. Vahdani and Zandieh (2010) further extended the work of Yu and

Egbelu (2008). They used the solution obtained from Yu and Egbelu's heuristic method as an initial solution and applied five metaheuristics; a genetic algorithm (GA), a tabu search (TS), a simulated annealing (SA), an electromagnetism-like algorithm and a variable neighborhood search (VSN) to improve solution quality in the evolution process. According to the numerical results, all five approaches were able to enhance solution quality with slightly higher computing times; however, VNS is recommended to use in a truck scheduling problem. Chen and Lee (2009), Boysen *et al.* (2010) and Liao *et al.* (2012) also presented several metaheuristic algorithms to solve the same problems.

There are much less research works considering the door assignment and truck scheduling problem simultaneously. Li *et al.* (2009) considered a multiple door cross docking system where all door could be used either as inbound or outbound doors. The problem was formulated as a parallel machine scheduling problem and assumed that there is no differentiation between inbound and outbound operations. Alpan *et al.* (2010) attempted to solve the multiple inbound and outbound doors cross docking configuration by a bounded dynamic programming. Liao *et al.* (2013) studies the dock assignment and sequencing of inbound trucks simultaneously with an objective to minimize total weighted tardiness under fixed outbound truck scheduling. The problem was solved by six metaheuristic algorithms: SA, TS, ant colony optimization (ACO), differential evolution (DE) and two hybrid differential evolution algorithms. The results demonstrated that ACO provide the best solution among other algorithms. Van Belle *et al.* (2013) presented a truck scheduling problem in a cross dock system consisting of multiple dock doors using the mixed integer programming. The objective of the study was to minimize the total travel time and the total tardiness. They showed that the MIP was able to solve the problems to optimal but with high expense of computational times. They, then proposed the TS approach to solve the problems, and the experimental result showed that TS is able to find a good result in the short period of time. Lee *et al.* (2012) proposed a genetic algorithm using simple chromosomes with dispatching rule (GA_DR) to solve the scheduling problem of inbound trucks and outbound trucks at multiple dock doors. The objective was to maximize the number of products transferred and shipped within a given working horizontal time.

As mentioned earlier, on-time delivery in a distribution network becomes very crucial especially when several distribution centres and customers are involved. A synchronized truck scheduling plan is needed to be harmonized, so that the distribution throughout the network can be optimized. Nevertheless, there are very few researcher works related to JIT content. Li *et al.* (2004) considered the cross dock operations with the JIT approach. The problem was modelled as a machine scheduling problem, and a genetic algorithm with two local search approaches were presented. The numerical ex-

periments showed that the first algorithms (IPGA) provided higher quality solutions but slower than the second algorithms (SWOGA). Another study concerned with JIT scheduling in cross dock terminals is the work of Arabani *et al.* (2010). Three metaheuristics; a genetic algorithm (GA), particle swarm optimization (PSO) and differential evolution (DE) were used to solve a single inbound and outbound door in a cross dock in order to find a schedule of trucks with minimization of earliness and tardiness.

Due to the importance of on-time delivery in a distribution network and only few studies focuses on the multi-objective scheduling in a cross docking system with JIT philosophy, this paper applies a differential evolution (DE) algorithm, one of the most efficient evolutionary algorithms (EAs), to minimize total earliness and total tardiness of trucks in the multi-door cross docking terminals.

3. PROBLEM DESCRIPTION

This study focuses on the operational activities at cross docking terminals with multiple inbound and outbound doors. At these terminals, several types of products from incoming trucks are unloaded, transferred and then loaded to any shipping dock by product need of each outbound truck in order to deliver to the demand points in a distribution network. One of the most important issues in cross docking system is to establish coordination between the performance of inbound and outbound trucks. An efficient cross docking system functions as a purely transshipment center in which trucks are assigned to doors and sequenced properly with minimum waiting time. Moreover, products must be transferred quickly, so that product storage can be minimized.

This study extends the mathematical models, developed by Yu and Egbelu (2008) and Van Belle *et al.* (2013). While Yu Egbelu (2008) considered the transshipment of multiple products in a simplified cross docking system with a single receiving door and a single shipping door, Van Belle *et al.* (2013) considered a cross docking system with multiple receiving and shipping doors to handle a single product type. Thus, this study integrated these two previous models and presents the transshipment problem of multiple product types in a multi-door cross docking system. The mathematical model is presented as a mixed integer programming (MIP). The basic assumptions for door assigning and truck sequencing problem used in this paper are listed as follows:

- Other cross dock operations such as sorting, labeling, packing, and unpacking docking are not considered.
- Products in inbound and outbound trucks have to be completely unloaded or loaded before leaving the dock door.
- Track changeover time is constant for all inbound and outbound trucks.

- Packaging size is predetermined as the same, thus the time for load and unload one unit of product is constant.
- The sequences of unloading products from the truck or loading products to the truck are not considered.
- The capacity of the temporary storage is infinite.

The door assignment and truck scheduling problem can be formulated as a mixed integer programming model. The following notions and parameters are used.

- I : number of inbound trucks ($i = 1, 2, \dots, I$)
 O : number of outbound trucks ($j = 1, 2, \dots, O$)
 P : number of product types ($k = 1, 2, \dots, P$)
 R : number of doors at receiving door ($m = 1, 2, \dots, R$)
 S : number of doors at shipping door ($n = 1, 2, \dots, S$)
 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 units of product type k from receiving door m to shipping door n
 di_i : departure time of inbound truck i
 do_j : departure time of outbound truck j
 L : loading or unloading time per product unit
 D : truck changeover time
 M : a positive big number

The decision variables are as follows:

- e_i : time at which inbound truck i enters the receiving door
 F_i : time at which inbound truck i leaves the receiving door
 d_j : time at which outbound truck j enters the shipping door
 L_j : time at which outbound truck j leaves the shipping door
 b_{ijk} : number of units of product type k transferred from inbound truck i to outbound truck j
- $$v_{ij} = \begin{cases} 1, & \text{if any products transfer from inbound truck } i \text{ to outbound truck } j \\ 0, & \text{otherwise} \end{cases}$$
- $$p_{i'i'} = \begin{cases} 1, & \text{if inbound trucks } i \text{ and } i' \text{ are assigned to the same door and truck } i \text{ is a predecessor of truck } i' \\ 0, & \text{otherwise} \end{cases}$$
- $$q_{j'j'} = \begin{cases} 1, & \text{if outbound trucks } j \text{ and } j' \text{ are assigned to the same door and truck } j \text{ is a predecessor of truck } j' \\ 0, & \text{otherwise} \end{cases}$$
- $$x_{im} = \begin{cases} 1, & \text{if inbound trucks } i \text{ is assigned to receiving door } m \\ 0, & \text{otherwise} \end{cases}$$
- $$y_{jn} = \begin{cases} 1, & \text{if outbound trucks } j \text{ is assigned to shipping door } n \\ 0, & \text{otherwise} \end{cases}$$

$$z_{ijmn} = \begin{cases} 1, & \text{if inbound trucks } i \text{ is assigned to receiving door } m, \text{ outbound trucks } j \text{ is assigned to shipping door } n \text{ and } v_{ij} = 1 \\ 0, & \text{otherwise} \end{cases}$$

The mathematical model of the problem is formulated as follows:

Minimize the Total Earliness:

$$\sum_{i=1}^I \max(0, di_i - F_i) + \sum_{j=1}^O \max(0, do_j - L_j) \quad (1)$$

Minimize the Total Tardiness:

$$\sum_{i=1}^I \max(0, F_i - di_i) + \sum_{j=1}^O \max(0, L_j - do_j) \quad (2)$$

Subjected to constraints:

$$\sum_{j=1}^O b_{ijk} = r_{ik} \quad (\forall i = 1 \dots I, \forall k = 1 \dots P) \quad (3)$$

$$\sum_{i=1}^I b_{ijk} = s_{jk} \quad (\forall j = 1 \dots O, \forall k = 1 \dots P) \quad (4)$$

$$\sum_{k=1}^P b_{ijk} \leq Mv_{ij} \quad (\forall i = 1 \dots I, \forall j = 1 \dots O) \quad (5)$$

$$\sum_{m=1}^R x_{im} = 1 \quad (\forall i = 1 \dots I) \quad (6)$$

$$\sum_{n=1}^S y_{jn} = 1 \quad (\forall j = 1 \dots O) \quad (7)$$

$$\sum_{m=1}^R \sum_{n=1}^S z_{ijmn} = v_{ij} \quad (\forall i = 1 \dots I, \forall j = 1 \dots O) \quad (8)$$

$$z_{ijmn} \leq x_{im} \quad (\forall i = 1 \dots I, \forall j = 1 \dots O, \forall m = 1 \dots R, \forall n = 1 \dots S) \quad (9)$$

$$z_{ijmn} \leq y_{jn} \quad (\forall i = 1 \dots I, \forall j = 1 \dots O, \forall m = 1 \dots R, \forall n = 1 \dots S) \quad (10)$$

$$x_{im} + x_{i'm} - 1 \leq p_{i'i'} + p_{i'i} \quad (\forall i, i' = 1 \dots I, i \neq i', \forall m = 1 \dots R) \quad (11)$$

$$p_{i'i'} + p_{i'i} \leq 1 \quad (\forall i, i' = 1 \dots I) \quad (12)$$

$$y_{jn} + y_{j'n} - 1 \leq q_{j'j} + q_{jj'} \quad (\forall j, j' = 1 \dots O, j \neq j', \forall n = 1 \dots S) \quad (13)$$

$$q_{j'j} + q_{jj'} \leq 1 \quad (\forall j, j' = 1 \dots O) \quad (14)$$

$$e_i \geq F_i + D - M(1 - p_{i'i'}) \quad (\forall i, i' = 1 \dots I) \quad (15)$$

$$F_i \geq e_i + L \sum_{k=1}^P r_{ik} \quad (\forall i = 1 \dots I) \quad (16)$$

$$d_j \geq L_j + D - M(1 - q_{j'j'}) \quad (\forall j, j' = 1 \dots O) \quad (17)$$

$$L_j \geq d_j + L \sum_{k=1}^P s_{jk} \quad (\forall j = 1 \dots O) \quad (18)$$

$$L_j \geq F_i + \sum_{m=1}^R \sum_{n=1}^S T_{mn} z_{ijmn} + L \sum_{k=1}^P s_{jk} - M(1 - v_{ij}) \quad (\forall i = 1 \dots I, \forall j = 1 \dots O) \quad (19)$$

$$\text{all variables} \geq 0 \quad (20)$$

The objective of the model is to minimize the total earliness and the total tardiness of inbound and outbound trucks as shown in equation (1) and (2) respectively. Constraint (3) ensures that the total number of product type k transferred from inbound truck i to all outbound trucks is equal to the number of product type k unloaded from inbound truck i . Similarly, Constraint (4) ensures that the total number of product type k transferred from all inbound truck to outbound truck j is equal to the number of product type k needed for outbound truck j . Constraint (5) enforces the exact relationship between the variable b_{ijk} and v_{ij} . Constraints (6) and (7) ensure that every inbound truck is assigned to a receiving door and every outbound truck is assigned to a shipping door, respectively. Constraints (8)-(10) enforce the correct relationship between the variables x_{im} , $y_{jn} - 1$, z_{ijmn} . Constraints (11)-(12) enforce the correct relationship between the x_{im} variables and the p_{ij} variables. Constraints (13)-(14) enforce the correct relationship between the variable y_{jn} and q_{ij} . Constraint (15) sets the entering time of each inbound truck to be equal to its predecessor plus truck changeover time. Constraint (16) sets the leaving time of each inbound truck to be equal to its entering time plus the time required to unloading all products. Constraint (17) states that the leaving time of each outbound truck must be greater than its predecessor leaving time plus the truck changeover time. Constraint (18) guarantees that the leaving time of each outbound truck must be greater than its entering time plus the time for loading all products. Constraint (19) ensures that the leaving time of each outbound truck must be greater than the leaving time of inbound truck plus the time to transfer and unload all products.

4. PROPOSED ALGORITHM

Storn and Price (1995) proposed Differential Evolution (DE) for global optimization over continuous space. DE is a population-based random search method like other Evolutionary Algorithms (EAs); however, its advantages are fewer control variables and performing well in search ability and convergence with less effort of computational times. The evolution of DE population continues through repeated cycles of three main operations; mutation, crossover, and selection until stopping criterion are met. DE has been widely applied and proved to be efficient in many application areas.

4.1 Multi-Objective Differential Evolution (MODE)

The real world problems typically contain several conflicting objectives. As a consequence, a research trend toward multi-objective optimization has become very attractive to practitioners. This paper adopts the framework of the Multi-Objective Differential Evolution (MODE) with a Pareto concept, proposed by Wisittipanich and Kachitvichyanukul (2014), in order to search for non-

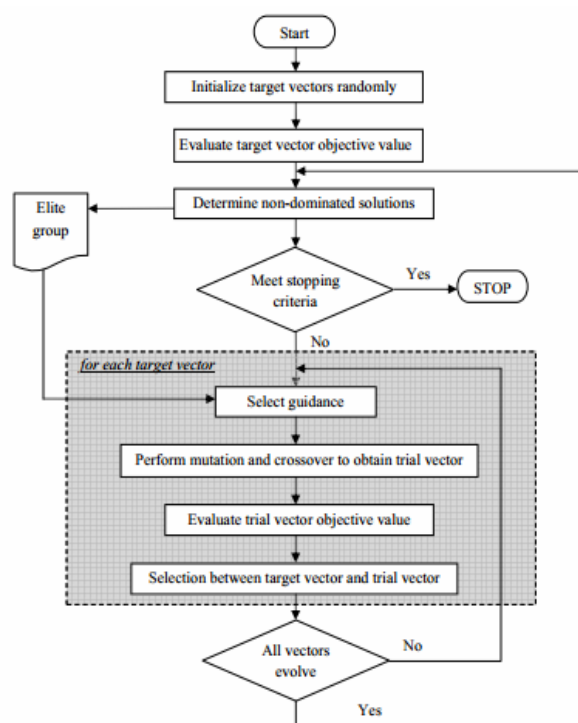


Figure 1. Framework of MODE (Wisittipanich and Kachitvichyanukul, 2014).

dominated solutions in the JIT door assignment and truck scheduling in a multi-door cross docking problem. The framework of MODE is illustrated in Figure 1.

In MODE, similarly to the Elitist structure in NSGA-II (Deb *et al.*, 2002), the population experiences are stored in an external archive, called Elite group, as a set of non-dominated solutions. Then, the MODE framework applies a sorting procedure to solutions in Elite group. More specifically, this sorting procedure only performs on the set of newly generated trial vectors after all vectors completed one move in order to identify the group of new non-dominated solutions. Then, Elite group screens its solutions to eliminate inferior solutions. As a result, Elite group in the archive contains only the best non-dominated solutions found so far in the searching process of the MODE population.

It is important to note that, in multi-objective problems, the presence of multiple candidates in the Elite group offers a large number of possibilities on the vector movements, and the quality of the final solutions are strongly influenced by the movement behavior adopted by the population. As a consequence, one of important decisions in multi-objective problems is how to select the candidates in the Elite group as guidance toward the Pareto frontier. This paper adopts three mutation strategies proposed in MODE as the movement guidance to utilize the information provided by the Elite group. Each mutation strategy has a distinct search behavior that directs vectors in DE population with the purpose of

attaining the high-quality Pareto optimal front. Three strategies used in this study are explained as following.

MODE-ms1: Search around solutions located in the less crowded areas

MODE-ms2: Pull the current front toward the true front

MODE-ms3: Fill the gaps of non-dominated solution on a front

Mutation strategies MODE-ms1 aims to discover new non-dominated solutions around solutions on the Pareto front which are located in the less crowded area. Mutation strategy MODE-ms3 intends to generate more non-dominated solutions to fill the gaps of the current Pareto front. Thus, both MODE-Ms1 and MODE-Ms3 aim to improve the distribution of solutions on the front. Mutation strategy MODE-Ms2 intends to pull to the current front toward the optimal Pareto front as close as possible. For more details on each strategy, see Wisittipanich and Kachitvichyanukul (2014).

4.2 Solution Representations

DE is initially designed for solving problems in continuous domain, thus, in order to apply DE to assignment and scheduling problems, a solution vector must be transformed. The procedures of solution mapping proposed in this study are illustrated in the following sections.

4.2.1 Encoding Procedures

In encoding procedures, a solution of the problem can be represented using a vector with dimensions equal to twice as much as the summation of the number of

inbound trucks (I) and outbound trucks (O). Consider an example of cross docking terminals with four inbound trucks, four outbound trucks, two inbound doors, and two outbound doors. Since there are total of eight trucks, the dimensions of a vector solution are set to be 16. Each value in a vector dimension is initially generated with a uniform random number between 0 and 1. The dimensions of a vector are divided into four parts sequentially; inbound trucks, outbound trucks, inbound doors, and outbound doors. An example of a solution vector is shown in Figure 2 (a).

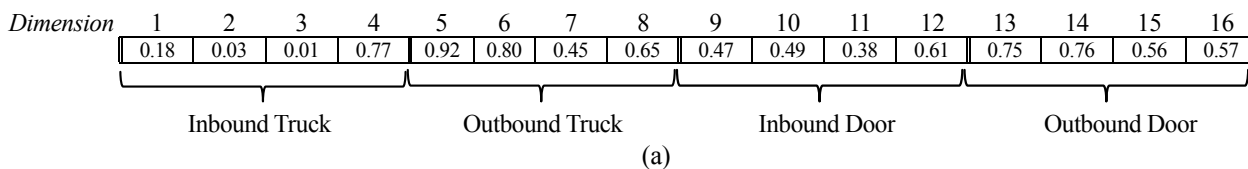
4.2.2 Decoding Procedures

When a solution vector is generated, a sorting list rule is applied to decode an individual vector into a sequence of trucks and the assignment of trucks to doors. As shown in Figure 2(b), a sequence of trucks is determined according to the order of sorted values in a vector dimension, and the door assignment is given using a sorting list rule with a repetitive-run number of dock doors. Therefore, each truck is assigned to its door correspondingly. The advantage of this decoding scheme is that it always provides a feasible solution.

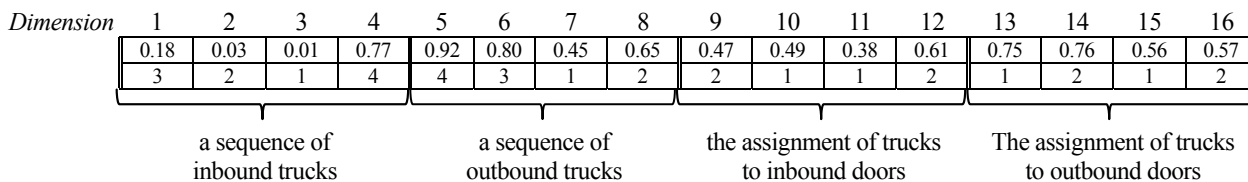
Then, this study proposes four decoding strategies to transform the door assignment and a truck sequence into the schedule with arrival time and departure time for each truck. The decoding rules proposed in this paper are divided into two stages. While, the first stage aims to find the schedule of inbound trucks, the second stage aims to determine the allocation of products to each outbound truck and the schedule of outbound trucks.

In the first stage, two decoding strategy, named as ITSH (Inbound_Truck_Shift) and ITDD (Inbound_Truck

Vector no. i



Decode Vector no. i



inbound truck sequence	3	2	1	4
assigned door	2	1	1	2

outbound truck sequence	4	3	1	2
assigned door	1	2	1	2

(b)

Figure 2. Example of solution representation of cross dock terminals with $I = O = 4$ and $R = S = 2$.

$_DueDate$), are proposed. The idea of ITSH is to determine the arrival time of each inbound truck randomly by assigning the arrival time of a truck between its ready time and its maximum value of the possible arrival time. Thus, it is expected that a truck arrives and departs not too early or too late. This algorithm starts with randomly determining shift time of each inbound truck and then calculating the arrival time and departure time of the truck. The pseudo codes of ITSH are shown in Algorithm ITSH.

Algorithm ITSH:

- I. For each inbound truck i in a sequence, identify the assigned door and the number of products for unloading
- II. Determine the shift time of inbound truck i
 $shift\ time = rand(0, 1) \times ((t-1) \times total\ unloading\ time)$
- III. Calculate arrival time and departure time of inbound truck i :
 If an inbound truck i is the first truck of inbound door
 $arrival\ time \leftarrow truck\ ready\ time + shift\ time$
 $departure\ time \leftarrow arrival\ time + total\ unloading\ time$
 Else
 $arrival\ time \leftarrow departure\ time\ of\ the\ previous\ truck + changeover\ time + shift\ time$
 $departure\ time \leftarrow arrival\ time + total\ unloading\ time$
- IV. Next inbound truck $i = i + 1$
- V. If $i =$ number of inbound trucks, stop. Otherwise, return to step I.

While ITSH randomly selects the arrival time of each inbound truck, ITDD algorithm aims to determine the departure time of inbound trucks as close to their due dates as possible. Thus the total tardiness of inbound trucks could be minimized. In ITDD, the time at the due date of an inbound truck is first checked whether it is available. If the time at the due date is available, the arrival time and departure time of a truck are then determined. However, if the time at the due date is occupied, the truck is scheduled at the best position next to the previous scheduled truck. The pseudo codes of ITDD are shown in Algorithm ITDD.

Algorithm ITDD:

- I. For each inbound truck i in a sequence, identify the assigned door, the number of products for unloading, and its due date
- II. Calculate arrival time and departure time of inbound truck i :
 If the time at due date is available
 $departure\ time \leftarrow due\ date$
 $arrival\ time \leftarrow departure\ time - total\ unloading\ time$
 Else
 $arrival\ time \leftarrow arrival\ time\ of\ the\ previous\ truck + changeover\ time$
 $departure\ time \leftarrow arrival\ time + total\ unloading\ time$
- III. Next inbound truck $i = i + 1$
- IV. If $i =$ number of inbound trucks, stop. Otherwise, return to step I.

In the second stage, product allocation for each truck and a schedule of outbound trucks are considered. This study proposes two decoding strategies, named OTSH (Outbound_Truck_Shift) and OTDD (Outbound_Truck_DueDate). In both OTSH and OTDD strategies, the ready time of each product to be unloaded from incoming trucks is first determined, and the product is allocated and loaded, according to the FCFS rule, to an outbound truck. The arrival time of an outbound truck is planned only when the last product required for that truck is ready to be loaded. Similar to ITSH, OTSH determines the arrival time of each outbound truck, derived from a sequence, randomly between the time at which the last required product is ready and its maximum value of the possible arrival time. So, it provides possibilities for a truck not to arrive beforehand or behind its due date. The pseudo codes of OTSH are shown in Algorithm OTSH.

Algorithm OTSH:

- I. Calculate the times at which each product is unloaded and transferred
 $time\ of\ a\ product\ ready\ to\ be\ loaded \leftarrow arrival\ time\ of\ an\ inbound\ truck + unloading\ time + transferring\ time\ to\ outbound\ door$
- II. Determine the time at which the last product is ready to be loaded to an outbound truck j
- III. For each outbound truck j in a sequence, identify the assigned door and its required products
- IV. Determine the shift time of outbound truck j
 $shift\ time = rand(0, 1) \times ((t-1) \times total\ loading\ time)$
- V. Calculate arrival time and departure time of outbound truck j
 If an outbound truck j is the first truck of outbound door
 $arrival\ time \leftarrow the\ ready\ time\ for\ the\ last\ required\ product\ to\ be\ loaded + shift\ time$
 $departure\ time \leftarrow arrival\ time + total\ loading\ time$
 Else
 $arrival\ time \leftarrow departure\ time\ of\ the\ previous\ truck + changeover\ time + shift\ time$
 $departure\ time \leftarrow arrival\ time + total\ loading\ time$
- VI. Next outbound truck $j = j + 1$
- VII. If $j =$ number of outbound trucks, stop. Otherwise, return to step III.

The purpose of OTDD algorithm is to schedule the departure time of outbound trucks to their due dates as close as possible, so that the total tardiness of outbound trucks could be minimized. When the time of the last product required for an outbound truck in a sequence is determined, OTDD first observes if the time at due date of the truck is occupied. If the time at the due date is available, the arrival time and departure time of an outbound truck are then determined. However, if the time at the due date is occupied, the outbound truck is scheduled at the best position next to the previous scheduled truck. The pseudo codes of OTDD are shown in Algo-

rithm OTDD.

Algorithm OTDD:

- I. Calculate the times at which each product is unloaded and transferred
 time of a product ready to be loaded \leftarrow arrival time of an inbound truck+unloading time+transferring time to outbound door
- II. Determine the time at which the last product is ready to be loaded to an outbound truck j
- III. For each outbound truck j in a sequence, identify the assigned door and its required products
- IV. Calculate arrival time and departure time of outbound truck j
 If the time at due date is available
 departure time \leftarrow due date
 arrival time \leftarrow departure time-total loading time
 if arrival time < the ready time for the last required product to be loaded
 arrival time \leftarrow the ready time for the last required product to be loaded
 departure time \leftarrow arrival time+total loading time
 Else
 arrival time \leftarrow arrival time of the previous truck+changeover time
 if arrival time < the ready time for the last required product to be loaded
 arrival time \leftarrow the ready time for the last required product to be loaded
 departure time \leftarrow arrival time + total loading time
- V. Next outbound truck $j = j + 1$
- VI. If $j =$ number of outbound trucks, stop. Otherwise, return to step III.

In this study, a pair of decoding strategies for inbound trucks and outbound trucks is used to determine a truck schedule. Since each decoding strategy has its own advantages, in decoding procedures, a combination of these pairs could provide good solution quality. There are four possible pairs of decoding strategies which are ITDD-OTDD, ITSH-OTSH, ITDD-OTSH, and ITSH-OTDD. An example illustrating the performances of different

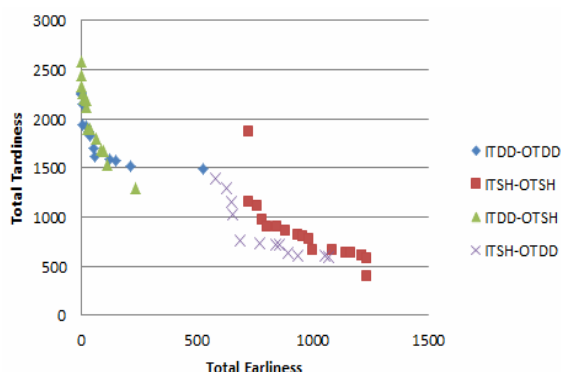


Figure 3. Comparison results among different pairs of decoding strategies.

pairs of decoding strategies is shown in Figure 3. The ITDD-OTDD and ITDD-OTSH have the potential to search for solutions with minimized total earliness (the left side of a Pareto front); while the ITSH-OTSH and ITSH-OTDD tends to perform well in searching solutions with the compromised schedules and minimized total tardiness (the center and the right side of a Pareto front). Due to different advantages of each pair of decoding strategies, a combination of these pairs is used in the experiment.

5. COMPUTATIONAL RESULTS

5.1 Parameter Setting

The performances of the MODE are not only influenced by its algorithm and movement strategies, but also by its parameters. In this study, for all MODE strategies (MODE-ms1, MODE-ms2, and MODE-ms3), the population size and number of iterations are set as 200 and 500 respectively to provide an adequate number of function evaluations in the search process. The value of scale factor F is set as a random value in order to allow a variation in the scaled difference and thus retains population diversity throughout the search process.

After some preliminary experiments, the value of F is set to be linearly increased from 0.4 to 0.9 as it provides generally good solution quality in all mutation strategies. Crossover rate (C_r) is set to be linearly increased from 0.1 to 0.5 to maintain the characteristic of generated trial vectors at the beginning of the search. As the search progress, increasing value of C_r yields more deviations for the generated trial vectors and helps the solutions to escape from being trapped at local optimal. In MODE-ms3, the potential gap is set to at least 5% of the difference between two fitness values so that more solutions are generated to fill the gap. In the MODE algorithm, a fixed Elite archive is used, and the maximum members in the archive are set as 100. According to some preliminary experiments, the combination of decoding strategies; ITDD-OTDD: ITSH-OTSH: ITDD-OTSH: ITSH-OTDD is set as 15: 15: 40: 30 since it generally provides a good quality Pareto front. The summary of MODE parameters used in this study is shown in Table 1.

Table 1. Parameters of MODE

Population size	200 Vectors
Number of iterations	500
Scale factor, F	Linearly increase from 0.4 to 0.9
Crossover rate, C_r	Linearly increase from 0.1 to 0.5
Potential gap	5% (MODE-ms3)
Upper limit of Elite group	100 vectors
Combination of decoding strategies	15:15:40:30

5.2 Experimental Results

The performances of the MODE with different mutation strategies are evaluated using 15 generated instances. The instance setting in this study is based on the setting in the research work of Van Belle *et al.* (2013). For all instances, the I-shaped cross-dock terminals are taken into a consideration and transferring times are calculated according to rectilinear distances. The cross dock problem size is characterized as $I, O, P, R,$ and S where I is the number of inbound trucks, O is the number of outbound trucks, P is number of product types, R is number of receiving doors, and S is number of shipping doors, respectively.

The objective of this study is to find a set of solutions that consists of total earliness and total tardiness values of the truck schedules without any biases. A decision maker can select any transshipment schedule that most meets his/her preference. To determine the truck due date, this study adopts the due date policy given by Eilon and Chowdhury (1976). The due date of inbound trucks and outbound trucks are calculated using the equation (20) and (21) respectively.

$$di_i = r_i + D + (t \times (L + b_{ijk})) \quad (20)$$

$$do_j = r_j + di_i + D + (t \times (L + b_{ijk} + T_{mi})) \quad (21)$$

The due date of truck (i or j) is equal to its ready time (r_i, r_j) plus the sum of all products transferring

times multiplied by a due date tightness factor, t . It is noted that, in this study, the ready time of all truck is set to be 0 and the tightness factor is set to be 1.5.

The experimental results are shown in Table 2. The notation (x, y) represents (total earliness, total tardiness). For each instance, the non-dominated solutions are determined from five independent runs.

As shown in Table 2 MODE-ms1, MODE-ms2, and MODE-ms3 perform well in generating a set of non-dominated solutions. All MODE strategies can find the optimal schedules with minimum total earliness in all instances. All strategies can also search for the optimal schedules with minimum total tardiness in most cases while some total tardiness are very close to the optimal solutions. Among MODE strategies, MODE-ms1 clearly outperforms MODE-ms2 and MODE-ms3 since, in most cases, the minimum total tardiness values obtained from MODE-ms1 are lower than those generated from MODE-ms2 and MODE-ms3, and the majority of the solutions obtained by MODE-ms2 and MODE-ms3 are dominated by those obtained by MODE-ms1. Figure 4 illustrates the comparison of non-dominated solutions generated from all MODE strategies.

5.3 Performance Measurement

Since the solutions in multi-objective problem based on a Pareto concept are generated as a set of non-dominated solutions, it makes the comparison of solutions more difficult than that in a single-objective optimization problem. This study uses \tilde{C} metric measurement to

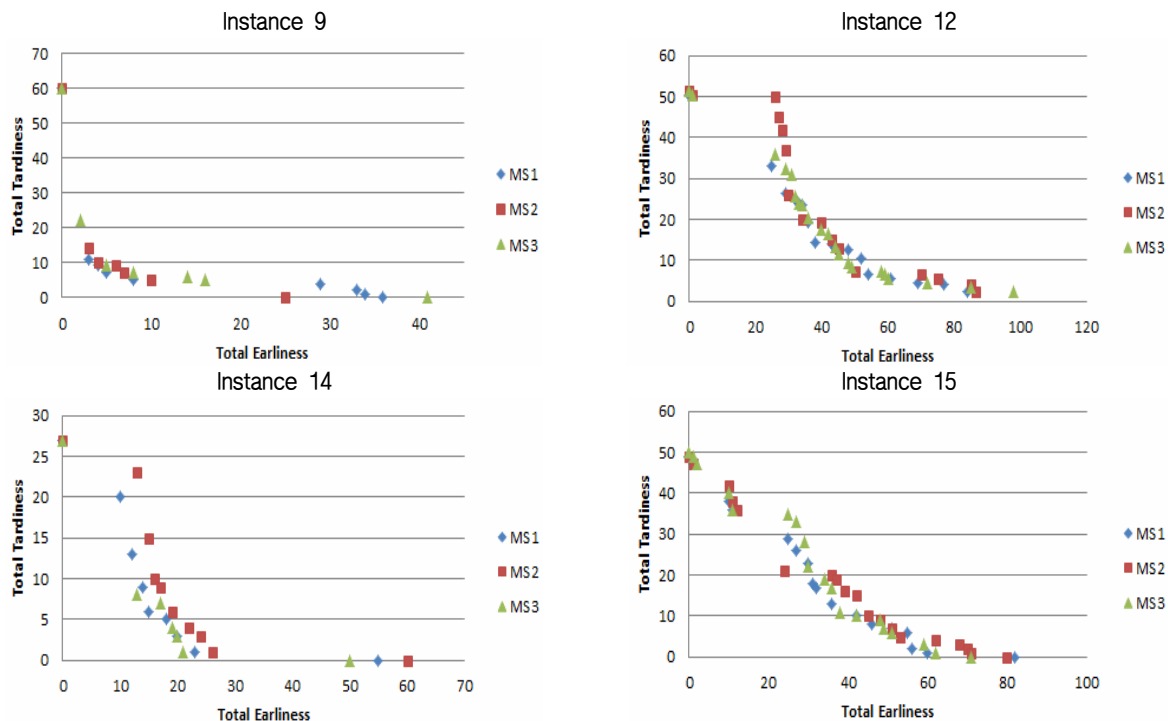


Figure 4. Comparison of non-dominated solutions among MODE strategies.

Table 2. Comparison results of MODE with different mutation strategies

Instant	Test Problem	LINGO	MODE															
			MODE-ms1			MODE-ms2			MODE-ms3									
			Total Earliness	Total Tardiness		Total Earliness	Total Tardiness		Total Earliness	Total Tardiness		Total Earliness	Total Tardiness					
1	3 4 5 1 2	0	1.5	(0, 25.3) (7, 19.3) (9, 17.3) (10, 16.3)	(12, 14.3) (14, 12.3) (16, 10.3) (17, 9.3)	(18, 8.3) (20, 7.3) (21, 6.3) (17, 9.3)	(23, 5.3) (24, 4.3) (26, 3.3) (20, 7.3)	(27, 2.5) (28, 2.3) (29, 1.5) (29, 1.5)	(0, 25.3) (7, 19.3) (10, 16.3) (11, 15.3)	(12, 14.3) (13, 13.3) (14, 12.3) (15, 11.3)	(16, 10.3) (17, 9.3) (18, 8.3) (20, 7.3)	(21, 6.3) (23, 5.3) (24, 4.3) (26, 3.3)	(27, 2.5) (28, 2.3) (29, 1.5) (26, 3.3)	(0, 25.3) (6, 20.3) (8, 18.3) (10, 16.3)	(11, 15.3) (12, 14.3) (14, 12.3) (15, 11.3)	(16, 10.3) (17, 9.3) (18, 8.3) (21, 6.3)	(23, 5.3) (24, 4.3) (26, 3.3) (21, 6.3)	
2	4 4 5 1 1	0	17	(0, 110) (19, 81) (20, 74) (22, 71) (25, 66) (28, 62)	(29, 61) (31, 58) (34, 54) (35, 53) (36, 52) (37, 50)	(38, 49) (40, 46) (42, 45) (44, 36.5) (48, 33.5)	(50, 32.5) (52, 30.5) (54, 29.5) (56, 27.5) (58, 26.5)	(60, 24.5) (95, 23) (98, 22) (101, 21) (103, 19)	(0, 110) (19, 81) (20, 74) (22, 71) (25, 66) (28, 62)	(29, 61) (31, 58) (34, 54) (35, 53) (36, 52) (37, 50)	(38, 49) (40, 46) (42, 45) (44, 36.5) (46, 35.5) (48, 33.5)	(50, 32.5) (52, 30.5) (54, 29.5) (56, 27.5) (58, 26.5) (58, 26.5)	(60, 24.5) (95, 23) (98, 22) (101, 21) (103, 19)	(0, 110) (13, 90) (15, 81) (18, 77) (25, 66) (34, 54)	(27, 65) (28, 62) (30, 60) (32, 58) (103, 19) (25, 66)	(37, 50) (38, 49) (40, 46) (46, 35.5) (50, 32.5) (34, 54)	(42, 45) (44, 36.5) (46, 35.5) (50, 32.5) (48, 33.5) (34, 54)	(60, 24.5) (97, 23) (98, 22) (103, 19) (103, 19)
3	4 4 3 1 2	0	8	(0, 90) (3, 87) (4, 79) (6, 77) (8, 76)	(9, 73) (27, 63) (29, 55) (31, 53) (32, 50)	(35, 45) (37, 40) (40, 37) (45, 31) (52, 25)	(54, 24) (56, 22) (61, 19) (62, 18) (52, 25)	(67, 15) (72, 12) (74, 10) (108, 9)	(0, 90) (3, 80) (5, 78) (6, 77) (7, 76)	(8, 75) (9, 73) (22, 68) (25, 65) (28, 60)	(31, 53) (32, 50) (35, 46) (37, 43) (41, 39)	(45, 31) (48, 26) (56, 22) (58, 20) (63, 17)	(66, 16) (69, 15) (76, 12) (86, 10) (63, 17)	(0, 90) (3, 80) (4, 79) (5, 78) (6, 77)	(7, 76) (8, 75) (9, 73) (25, 69) (26, 66)	(29, 57) (33, 47) (40, 36) (45, 35) (48, 26)	(51, 25) (56, 22) (58, 20) (63, 17) (48, 26)	(69, 13) (80, 11) (102, 10) (129, 9)
4	4 5 5 1 1	0	1	(0, 55) (1, 21)	(2, 20) (3, 19)	(3, 6) (4, 2)	(5, 3) (6, 2)	(7, 3) (7, 3)	(0, 55) (1, 21)	(2, 20) (3, 10)	(3, 8) (4, 4)	(4, 3) (4, 2)	(6, 1) (5, 2)	(0, 55) (1, 21)	(2, 20) (2, 8)	(3, 7) (3, 6)	(3, 6) (4, 3)	(5, 2) (6, 2)
5	4 6 5 2 1	0	0	(0, 60) (3, 11)	(4, 9) (5, 7)	(8, 5) (29, 4)	(33, 2) (34, 1)	(36, 0) (88, 1)	(0, 60) (3, 14)	(4, 10) (6, 9)	(7, 7) (10, 5)	(25, 0) (10, 5)	(25, 0) (45, 3)	(0, 60) (86, 1)	(5, 9) (20, 12)	(14, 6) (31, 6)	(16, 5) (44, 3)	(41, 0) (91, 1)
6	5 5 6 2 1	0	0	(0, 21) (25, 7)	(34, 6) (41, 4)	(48, 3) (49, 2)	(88, 1) (104, 0)	(88, 1) (104, 0)	(0, 21) (18, 12) (22, 11)	(24, 9) (28, 8) (32, 7)	(34, 6) (40, 4)	(45, 3) (51, 2)	(86, 1) (102, 0)	(0, 21) (18, 14) (19, 14)	(20, 12) (24, 8) (27, 7)	(31, 6) (41, 5) (42, 4)	(44, 3) (49, 2) (42, 4)	(91, 1) (111, 0)
7	5 5 6 2 2	0	1.5	(0, 50.5) (25, 33) (29, 26.5) (32, 25)	(32.5, 24.5) (34, 23.5) (36, 19.5) (38, 14.5)	(43, 14) (44, 13) (48, 12.5)	(84, 2.5) (52, 10.5) (54, 6.5)	(61, 5.5) (69, 4.5) (77, 4)	(0, 51.5) (1, 50.5) (26, 50) (27, 45)	(28, 42) (29, 37) (30, 26)	(34, 20) (40, 19.5) (43, 15)	(45, 13) (50, 7.5) (70, 6.5)	(75, 5.5) (85, 4) (86.5, 2.5)	(0, 51.5) (1, 50.5) (26, 36) (29, 32.5)	(32, 25.5) (33, 24) (44, 13.5) (45, 11.5)	(40, 17.5) (42, 16.5) (44, 13.5) (45, 11.5)	(48, 9.5) (49, 8.5) (58, 7.5) (59, 6.5)	(60, 5.5) (72, 4.5) (85, 3.5) (98, 2.5)
8	5 5 6 3 2	0	0	(0, 27) (10, 20)	(12, 13) (14, 9)	(15, 6) (18, 5)	(20, 3) (23, 1)	(55, 0) (82, 0)	(0, 27) (13, 23)	(15, 15) (16, 10)	(17, 9) (19, 6)	(22, 4) (24, 3)	(26, 1) (60, 0)	(0, 27) (13, 8)	(17, 7) (19, 4)	(20, 3) (21, 1)	(50, 0) (42, 10)	(59, 3) (62, 1) (71, 0)
9	5 6 4 2 2	0	0	(0, 49) (1, 47) (10, 38) (11, 36)	(25, 29) (27, 26) (30, 23) (31, 18)	(32, 17) (36, 13) (42, 10)	(46, 8) (51, 7) (55, 6)	(56, 2) (60, 1) (82, 0)	(0, 49) (1, 47) (10, 42) (11, 38)	(12, 36) (24, 21) (36, 20)	(39, 16) (42, 15) (45, 10) (48, 9)	(51, 7) (53, 5) (62, 4) (68, 3)	(70, 2) (71, 1) (80, 0)	(0, 50) (1, 49) (2, 47) (10, 40)	(11, 36) (25, 35) (27, 33) (29, 28)	(30, 22) (34, 19) (36, 17) (38, 11)	(42, 10) (48, 9) (49, 7) (51, 6)	(59, 3) (62, 1) (71, 0)
10	6 7 5 2 3	0	9	(0, 117) (30, 84) (33, 79) (40, 70) (42, 65) (50, 64)	(53, 56) (54, 52) (60, 46) (65, 40) (75, 36) (77, 33)	(87, 32) (94, 29) (95, 28) (99, 24) (103, 23)	(104, 22) (108, 20) (110, 18) (117, 17) (123, 16)	(131, 14) (140, 12) (267, 11) (287, 10) (292, 9)	(0, 117) (30, 112) (37, 111) (39, 73) (103, 23)	(42, 66) (50, 54) (63, 40) (73, 39)	(80, 33) (87, 30) (98, 29) (101, 28)	(110, 26) (119, 18) (122, 16) (132, 15)	(144, 12) (244, 11) (265, 10) (289, 9)	(0, 117) (31, 112) (33, 111) (43, 104) (44, 62)	(46, 48) (66, 39) (74, 36) (104, 25)	(79, 33) (91, 32) (95, 27) (126, 14)	(106, 22) (110, 19) (121, 16) (104, 25)	(139, 13) (142, 12) (274, 10) (278, 9)

evaluate and compare the solutions obtained from different MODE mutation strategies. The measurement $\tilde{C}(A, B)$ measures the portions of members of B that are dominated by members of A. The value of $\tilde{C}(A, B)$ is calculated using the following equation.

$$\tilde{C}(A, B) = \frac{|\{b \in B : \exists a \in A, a \succ b\}|}{|B|} \quad (22)$$

Where $|B|$ is the number of solutions in B. The value of $\tilde{C}(A, B)$ are real number between 0 and 1. If the value of $\tilde{C}(A, B)$ is equal to 1, it denotes that each solution in B is dominated by some solutions in A. On the other hand, if the value of $\tilde{C}(A, B)$ is equal to 0, it means that all solutions in B are non-dominated by any solution in A. Therefore, the lower the ratio $\tilde{C}(A, B)$ is, the better the solution set in B is.

Table 3 shows the comparison results among MODE with different mutation strategies; MODE-ms1, MODE-ms2, and MODE-ms3. It should be noted that 1, 2, and 3 represent MODE-ms1, MODE-ms2, and MODE-ms3, respectively.

According to the results in Table 3, it is confirmed that MODE-ms1 is superior to MODE-ms2 and MODE-ms3. In most cases, the MODE-ms1 obtains the \tilde{C} metric values $\tilde{C}(2, 1)$ and $\tilde{C}(3, 1)$ of 0 or close to 0 whereas MODE-ms2 and MODE-ms3 generally obtain the \tilde{C} metric values $\tilde{C}(1, 2)$ and $\tilde{C}(1, 3)$ respectively) of 1 or close to 1. Especially in the large-size problems, the MODE-ms1 clearly outperforms MODE-ms2 and MODE-ms3 since \tilde{C} metric values of $\tilde{C}(2, 1)$ and $\tilde{C}(3, 1)$ are equal to 1 and \tilde{C} metric values of $\tilde{C}(1, 2)$ and $\tilde{C}(1, 3)$ are equal to 0. This demonstrates that solutions generated from MODE-ms2 and MODE-ms3 are completely

dominated by those obtained from MODE-ms1. In addition, for most instances, the performances of MODE-ms3 are comparable with those of MODE-ms2. However, when the problem size increases, MODE-ms3 shows its outstanding results over MODE-ms2 since most of solutions generated from MODE-ms2 are dominated by those generated from MODE-ms3.

6. CONCLUSION

This study presents the implementation of MODE algorithm for multi-objective door assignment and truck scheduling problem in a multi-door cross docking system according to Just-In-Time (JIT) concept. The objective is to find the schedule of transshipment operations to simultaneously minimize the total earliness and the total tardiness of trucks. The MODE framework uses an Elite group to store non-dominated solutions and utilizes those solutions as the guidance of vector movements. This study proposes a particular encoding scheme and four decoding strategies, named as ITSH, ITDD, OTSH and OTDD, in the MODE solution mapping procedures to transform a vector into the truck schedule. The experimental results demonstrate that the proposed algorithm is capable of finding a set of diverse and high quality non-dominated solutions on a Pareto front.

For future research directions on a cross docking system, it would be very interesting to combine the truck scheduling problem with other cross docking related problems such as vehicle routing problem and resources allocation problem. Moreover, the synchronization of inbound trucks and outbound trucks flows for multiple cross docking systems is also desirable.

Table 3. The \tilde{C} metric comparison between MODE mutation strategies

Instance	$\tilde{C}(1, 2)$	$\tilde{C}(2, 1)$	$\tilde{C}(1, 3)$	$\tilde{C}(1, 3)$	$\tilde{C}(2, 3)$	$\tilde{C}(2, 3)$
1	0.000	0.000	0.000	0.000	0.000	0.000
2	0.000	0.000	0.115	0.037	0.115	0.036
3	0.292	0.000	0.217	0.174	0.174	0.042
4	0.111	0.333	0.000	0.667	0.100	0.556
5	0.714	0.444	0.714	0.000	0.571	0.143
6	0.250	0.500	0.385	0.250	0.538	0.500
7	0.688	0.118	0.714	0.176	0.238	0.563
8	0.900	0.000	0.143	0.333	0.000	0.900
9	0.684	0.294	0.632	0.235	0.368	0.737
10	0.500	0.296	0.476	0.296	0.190	0.550
11	0.364	0.083	0.500	0.167	0.300	0.364
12	0.941	0.040	0.720	0.120	0.040	0.882
13	1.000	0.000	0.600	0.560	0.000	1.000
14	1.000	0.000	1.000	0.000	0.000	1.000
15	1.000	0.000	1.000	0.000	0.000	1.000

ACKNOWLEDGMENTS

The authors would like to gratefully acknowledge the Excellence Centre in Logistics and Supply Chain Management (E-LSCM), Chiang Mai University for the supporting of this research work.

REFERENCES

- Alpan, G., Larbi R., and Penz B. (2010), A bounded dynamic programming approach to schedule operations in a cross docking platform, *Computers and Industrial Engineering*, **60**, 385-396.
- Arabani, A. R. B., Ghomi, S. M. T. F., and Zandieh, M. (2010), A multi-criteria cross-docking scheduling with just-in-time approach, *International Journal of Advanced Manufacturing Technology*, **49**, 741-756.
- Bartholdi, J. J. and Gue, K. R. (2000), Reducing labor cost in an LTL crossdocking terminal, *Operations*

- Research*, **48**, 823-832.
- Boysen, N., Fliedner, M., and Scholl, A. (2010), Scheduling inbound and outbound trucks at cross docking terminals. *OR Spectrum*, **32**, 135-161.
- Chen, F. and Lee, C.-Y. (2009), Minimizing the make-span in a two-machine cross-docking flow shop problem, *European Journal of Operational Research*, **193**, 59-72.
- Cohen, Y. and Keren, B. (2009), Trailer to door assignment in a synchronous cross-dock operation, *International Journal of Logistics Systems and Management*, **5**, 574-590.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, **6**, 182-197.
- Eilon, S. and Chowdhury, I. G. (1976), Due date in job shop scheduling, *International Journal of Production Research*, **14**, 233.
- Lee, K., Kim, B. S., and Joo, C. M. (2012), Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance, *Expert Systems with Applications*, **39**, 12975-12983.
- Li, Y., Lim, A. and Rodrigues, B. (2004), Crossdocking: JIT Scheduling with Time Windows, *The Journal of the Operational Research Society*, **55**, 1342-1351.
- Li, Z. P., Low, M. Y. H., Shakeri, M. and Lim, Y. G. (2009), Cross docking planning and scheduling: problems and algorithms, *SIMTech Technical Reports*, **10**, 159-167.
- Liao, T. W., Egbelu, P. J., and Chang, P. C. (2012), Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations, *Applied Soft Computing*, **12**, 3683-3697.
- Liao, T. W., Egbelu, P. J., and Chang, P. C. (2013), Simultaneous dock assignment and sequencing of
- multaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations, *International Journal of Production Economics*, **141**, 229-212.
- Wisittipanich, W. and Kachitvichyanukul V. (2014), Mutation strategies toward Pareto front for multi-objective differential evolution algorithm, *International Journal of Operational Research*, **19**, 315-337.
- Storn, R. and Price, K. (1995), Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, *International Computer Science*, Berkeley, CA.
- Tsui, L. Y. and Chang, C. H. (1990), A microcomputer based decision support tool for assigning dock doors freight yards, *Computers and Industrial Engineering*, **19**, 309-312.
- Tsui, L. Y. and Chang, C. H. (1992), An optimal solution to a dock door assignment problem, *Computers and Industrial Engineering*, **23**, 283-286.
- Vahdani, B. and Zandieh, M. (2010), Scheduling trucks in cross-docking systems: robust meta-heuristics, *Computers and Industrial Engineering*, **58**, 12-24.
- Van Belle, J., Valckenaers, P., and Cattrysse, D. (2012), Cross-docking: State of the art, *Omega*, **40**, 827-846.
- Van Belle, J., Valckenaers, P., Vanden Berghe, G., and Cattrysse, D. (2013), A tabu search approach to the truck scheduling problem with multiple docks and time windows, *Computers and Industrial Engineering*, **66**, 818-826.
- Yu, W. and Egbelu, P. J. (2008), Scheduling of inbound and outbound trucks in cross docking systems with temporary storage, *European Journal of Operational Research*, **184**, 377-396.