BULLETIN OF THE POLISH ACADEMY OF SCIENCES
TECHNICAL SCIENCES, Vol. 67, No. 2, 2019
DOI: 10.24425/bpas.2019.128113

Scheduling trucks in a multi-door cross-docking system with time windows

G. OZDEN[1]* and I. SARICICEK[2]

[1] Eskisehir Osmangazi University, Eskisehir Vocational School, Machinery Program, Eskisehir, Turkey
[2] Eskisehir Osmangazi University, Department of Industrial Engineering, Eskisehir, Turkey

Abstract. Cross-docking is a strategy that distributes products directly from a supplier or manufacturing plant to a customer or retail chain, reducing handling or storage time. This study focuses on the truck scheduling problem, which consists of assigning each truck to a door at the dock and determining the sequences for the trucks at each door considering the time-window aspect. The study presents a mathematical model for door assignment and truck scheduling with time windows at multi-door cross-docking centers. The objective of the model is to minimize the overall earliness and tardiness for outbound trucks. Simulated annealing (SA) and tabu search (TS) algorithms are proposed to solve large-sized problems. The results of the mathematical model and of meta-heuristic algorithms are compared by generating test problems for different sizes. A decision support system (DSS) is also designed for the truck scheduling problem for multi-door cross-docking centers. Computational results show that TS and SA algorithms are efficient in solving large-sized problems in a reasonable time.

Key words: multi-door cross-docking center, time window, logistics, decision support system, meta-heuristics.

1. Introduction

Cross-docking is the operation of conveying products through distribution centers without warehousing them. In a traditional warehouse, the products move from receiving to storage and then from storage to shipping processes. The cross-docking system's success depends on as short a storage period as possible in the receiving/shipping plant [1]. By using a cross-docking system, goods are delivered to the cross-docking center via inbound trucks and are directly sorted, repackaged, routed and loaded into outbound trucks to be delivered, thus bypassing storage [2].

One primary activity of a cross-docking system in particular entails the effective coordination of inbound and outbound trucks. As order volumes increase or when deliveries are uncoordinated, the inventory can increase, and thus cross-docking centers must be managed efficiently. Meanwhile, the truck scheduling problem is one of the most significant operational problems that consist in assigning and sequencing trucks at receiving or shipping doors in cross-docking centers. McWilliams, Stanfield and Geiger [3] were the first to address the issue of short-term truck scheduling and to examine a terminal of a postal service provider to minimize the time span of the transfer operation. They also proposed a simulation-based scheduling algorithm that uses a genetic algorithm (GA). Yu and Egbelu [4] proposed a mathematical model to obtain the best schedule for all inbound and outbound trucks to minimize the total operation time. Because the truck scheduling problem is NP-hard [5], meta-heuristics have often been applied. Vahdani and Zandieh [2] applied GA, TS, SA, electromagnetism-like algorithm (EMA) and variable neighborhood search (VNS), while Boloori Arabani, Fatemi Ghomi and Zandieh, [6] proposed GA, TS, particle swarm optimization (PSO), ant colony optimization (ACO) and differential evolution (DE) algorithms to schedule trucks in cross-docking centers that were previously suggested by Yu and Egbelu.

Van Belle, Valckenaers and Cattrysse [7] classify studies regarding the truck scheduling problem into three groups. The first group indicates a cross-docking center with a single receiving and single shipping door [2, 4, 6, 8–12]. The truck scheduling problem's objective is to decide where and when inbound and outbound trucks should be loaded or unloaded at multi-door cross-docking centers. Studies in the second group consider cross-docks with multiple receiving and shipping doors but solely address scheduling the inbound trucks [3, 13–17]. The last category indicates scheduling both inbound and outbound trucks at multiple receiving and shipping doors. A number of studies in this category are available in the literature. Boysen [18] suggests SA algorithm and the dynamic programming method for truck scheduling in the cross-docking process. Meanwhile, Lee, Kim and Joo [19] introduce a mixed integer programming (MIP) model to obtain door assignment and docking sequences for all trucks. The researchers propose a GA to maximize the number of products that need to be shipped within a given working period. Joo and Kim [20] consider a truck scheduling problem for three groups of trucks: inbound trucks, compound trucks and outbound trucks. The researchers first introduce compound trucks that play the roles of outbound and inbound trucks, as in this paper. Self-evolution algorithm and GA are proposed for truck scheduling to minimize the makespan. The study by Van Belle, Valckenaers,

*e-mail: gozden@ogu.edu.tr

Manuscript submitted 2018-02-06, revised 2018-06-05, initially accepted for publication 2018-07-03, published in April 2019.

Berghe and Cattrysse [21] is also concerned with scheduling both inbound and outbound trucks at multiple receiving and shipping doors and a TS algorithm is proposed to solve large-sized problems. Assadi and Bagheri [22] propose SA and GA to solve the truck scheduling problem with ready times for inbound trucks at multi-door cross-docking centers.

This paper focuses on the truck scheduling problem to minimize overall earliness and tardiness of outbound trucks considering the time window aspect in a cross-docking system with multiple receiving and shipping doors. Furthermore, a DSS is designed to solve the problem indicated. In a multi-door cross-docking center, the number of trucks is greater than the number of doors. Therefore, while some of the inbound or outbound trucks are processed, the remaining inbound and outbound trucks must wait in the line. The assignment of inbound and outbound trucks to appropriate inbound and outbound door influences the effectiveness of the cross-docking system. In the studies by Boysen [18], Lee, Kim and Joo [19] as well as by Joo and Kim [20], the arrival times of inbound and outbound trucks at the cross-docking center are the same. However, in real-life problems, the arrival times of inbound, outbound and compound trucks all differ. There are thus different arrival times for inbound, outbound and compound trucks in our study. Additionally, we focus on the truck scheduling problem with the objective of minimizing overall earliness and tardiness of outbound trucks in the cross-docking center considering the time window aspect. Moreover, a great number of studies assume that products aren't interchangeable; however, at the cross-docking centers, not only truck scheduling but also product assignment must be determined. Product assignment is also considered from inbound trucks to outbound trucks simultaneously with the door assignments and docking sequences of inbound and outbound trucks in our study. In all available research papers for truck scheduling in a cross-docking center, excluding the study by Joo and Kim [20], the trucks are grouped as inbound and outbound ones. However, certain trucks arrive at the receiving dock to unload products and then visit the shipping dock to load products. We also consider such trucks, herein referred to as compound trucks, in our study. The mixed integer programming model is formulated for the truck scheduling problem with all the above-indicated assumptions. The results of the exact solution are compared with the results of TS and SA meta-heuristics from the perspective of computational times and objective function values. The decision support system is developed by using PyCharm Community Edition 3.4 to assign trucks to the doors in order to minimize overall tardiness and earliness. The DSS provides truck and product assignment schedules to the decision maker. Based on the proposed system, managers can monitor appropriate schedules for the cross-docking system.

The remainder of this paper is organized as follows. In section 2, a mixed integer programming model is proposed for the truck scheduling problem. In section 3, the generation of the test problems is presented. The problems are then solved by TS and SA algorithms, and the performance of meta-heuristics algorithms is evaluated by means of several numerical experiments in section 4. Finally, a summary and further research issues are provided in section 5.

2. Problem description

The cross-docking center examined in this study has a receiving dock where products are unloaded from inbound trucks and a shipping dock where products are loaded onto outbound trucks with multiple doors, and the docks are separated from one another (Fig. 1).

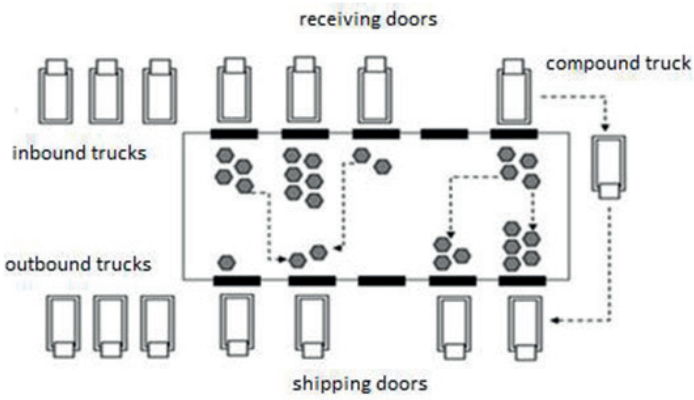

Fig. 1. Cross-docking center and truck groups

Inbound trucks arrive at the assigned receiving door successively and unload products onto the receiving dock. Outbound trucks arrive at the assigned shipping door successively and load products from the shipping dock. Compound trucks are described as both inbound and outbound ones. A compound truck is an inbound truck when it arrives at the receiving dock and it is an outbound one when it goes to the receiving dock to load products. Compound trucks unload products onto the receiving dock and transfer to the shipping dock to load products. The problem is the determination of when and where inbound, outbound and compound trucks should be processed at multi-door cross-docking centers. The minimizing of overall earliness and tardiness of outbound trucks is crucial in a cross-docking center considering the time window aspect. The time window for outbound trucks is shown in Fig. 2.

A time window is a time interval in which the outbound trucks' loading operation should be completed. If the loading operation is finished within the time window, no tardiness or earliness is incurred, and the process is considered to be completed on time. Otherwise, the operation causes a penalty, de-

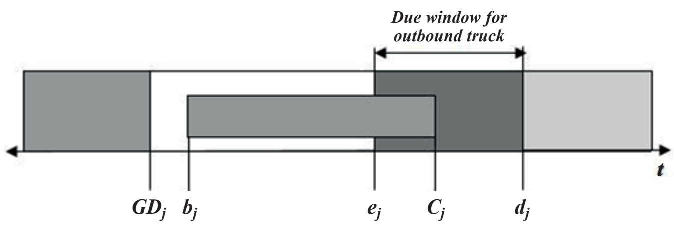

Fig. 2. Time window for an outbound truck

pending on whether the loading operation of a truck is finished before or after the time window. In Fig. 2, e_j, d_j, GD_j, b_j and C_j refer to the beginning of the time window, the end of the time window, the arrival time of outbound truck j, and then the start time and completion time of loading onto outbound truck j, respectively.

Minimizing overall earliness and tardiness can be of particular use for the cross-docking systems that use the "just in time" approach to deliver products in the customer's time window for due dates. The punctuality of product deliveries affects the system performance. There are time windows for outbound trucks that are determined by the customer. Outbound trucks should be delivered to the customer within these specific time windows.

According to the notation proposed by Boysen and Fliedner [23], the truck scheduling problem consists of three main elements which are denoted as tuple $\alpha/\beta/\gamma$ where α is the door environment, β stands for operational characteristics and γ means the objectives. This truck scheduling problem can be denoted as follows:

$$E/r_j, \bar{d}_o, change/\sum(E_j + T_j).$$

Here E, r_j, $\bar{d}_o$, $change$ and $\sum(E_j + T_j)$ represents that each door is either exclusively dedicated to inbound or outbound operations, and that arrival times are different per truck, deadlines for outbound trucks, interchangeability of products as well as the objective to minimize overall earliness and tardiness, respectively.

The assumptions of this problem are defined as follows:

- Inbound, outbound and compound trucks must remain at the respective doors until their unloading or loading operations are finished and must leave as soon as they finish their operation.
- After products are unloaded, they are transferred to the shipping doors and stay until the appropriate outbound truck arrives there.
- Because the freight is shipped in standardized pallets, the unit loading time and unloading time are assumed to be fixed. The unit unloading time and the unit loading time remain the same regardless of product type and are calculated for one unit of product.
- The expected arrival time is known for each inbound, outbound and compound truck.
- There are compound trucks that both unload and load products.
- Inbound and outbound trucks hold different products and they are known.
- The sequence in which goods are unloaded or loaded is not considered.
- The truck changeover time is the same for all trucks and unit transfer time from the receiving area to the shipping area is the same for all product types.
- The time windows are considered for outbound trucks. The time windows are not defined as hard constraints; however, tardiness and earliness of outbound trucks should be minimized.

2.1. Parameters

I: $\{1, \ldots, n_{it}\}$ is the set of inbound trucks (index i)
O: $\{1, \ldots, n_{ot}\}$ is the set of outbound trucks (index j)
C: $\{1, \ldots, n_{ct}\}$ is the set of compound trucks (index $i \in$ I and index $j \in 0$)
P: $\{1, \ldots, n_{p}\}$ is the set of product types (index k)
R: $\{1, \ldots, n_{rd}\}$ is the set of receiving doors (index m)
S: $\{1, \ldots, n_{sd}\}$ is the set of shipping doors (index n)

r_{ik} The number of products of type k that is loaded onto inbound truck i
s_{jk} The number of products of type k that is needed for outbound truck j
CT Truck changeover time (TE + TL)
UT The unit unloading time of products
LT The unit loading time of products
TA Transfer time for compound truck from receiving dock to shipping dock
GL_i Arrival time of inbound truck i at cross-docking center
GD_i Arrival time of outbound truck j at cross-docking center
TE The truck entering time to a door
TL The truck leaving time from a door
e_j Beginning of the time window
d_j End of the time window
V The moving time of products from receiving dock to shipping dock
M A positive large number which is at least as large as the sum of loading and unloading times and truck changeover times for all trucks.

2.2. Decision variables

a_i Start time of unloading for inbound truck i
L_i Completion time of unloading for inbound truck i
b_j Start time of loading for outbound truck j
C_j Completion time of loading for outbound truck i
x_{ijk} The number of products of type k that are transferred from inbound truck i to outbound truck j
E_j Earliness of outbound truck j $E_j = max\{e_j - C_j, 0\}$
T_j Tardiness of outbound truck j $T_j = max\{C_j - d_j, 0\}$

$$g_{ij} = \begin{cases} 1, & \text{if any products are transferred from inbound truck } i \text{ to outbound truck } j \\ 0, & \text{otherwise} \end{cases}$$

$$p_{ijm} = \begin{cases} 1, & \text{if truck } i \text{ is assigned before truck } j \text{ in the sequence at receiving door } m (i \neq j)\text{; or if inbound truck } i \text{ is the first truck at receiving door } m (i = j) \\ 0, & \text{otherwise} \end{cases}$$

$$q_{ijn} = \begin{cases} 1, & \text{if truck } j \text{ is assigned before truck } i \text{ in the sequence at shipping door } n (j \neq i)\text{; or if outbound truck } j \text{ is the first truck at shipping door } n (j = i) \\ 0, & \text{otherwise} \end{cases}$$

$$y_{im} = \begin{cases} 1, & \text{if inbound truck } i \text{ is assigned to receiving door } m \\ 0, & \text{otherwise} \end{cases}$$

$$z_{jn} = \begin{cases} 1, & \text{if outbound truck } j \text{ is assigned to shipping door } n \\ 0, & \text{otherwise} \end{cases}$$

2.3. Mathematical model

$$\text{Min} \sum_{j=1}^{n_{ot}} (E_j + T_j) \tag{1}$$

s.t.

$$a_i + \left(UT \cdot \sum_{k=1}^{n_p} r_{ik} \right) \leq L_i \qquad \forall (i \in \mathrm{I}) \tag{2}$$

$$L_i + CT \leq a_j + M \cdot \left(1 - \sum_{m=1}^{n_{rd}} p_{ijm} \right) \qquad \forall (i, j \in \mathrm{I})\ i \neq j \tag{3}$$

$$\mathrm{b_j} + \left(LT \cdot \sum_{k=1}^{n_p} s_{jk} \right) \leq C_j \qquad \forall (j \in \mathrm{O}) \tag{4}$$

$$C_j + CT \leq b_i + M \cdot \left(1 - \sum_{n=1}^{n_{sd}} q_{ijn} \right) \qquad \forall (i, j \in \mathrm{O})\ i \neq j \tag{5}$$

$$a_i \geq GL_i + TE \cdot \left(\sum_{m=1}^{n_{rd}} p_{iim} \right) \qquad \forall (i \in \mathrm{I}) \tag{6}$$

$$b_j \geq GD_j + TE \cdot \left(\sum_{n=1}^{n_{sd}} q_{jjn} \right) \qquad \forall (j \in \mathrm{O}) \tag{7}$$

$$L_i + V \leq b_j + M \cdot (1 - g_{ij}) \qquad \forall (i \in \mathrm{I}, j \in \mathrm{O}) \tag{8}$$

$$\sum_{m=1}^{n_{rd}} y_{im} = 1 \qquad \forall (i \in \mathrm{I}) \tag{9}$$

$$\sum_{i=1}^{n_{it}} p_{iim} = 1 \qquad \forall (m \in \mathrm{R}) \tag{10}$$

$$\sum_{j=1}^{n_{it}} p_{jim} = y_{im} \qquad \forall (i \in \mathrm{I}, m \in \mathrm{R}) \tag{11}$$

$$\sum_{j=1}^{n_{it}} p_{ijm} \leq y_{im} \qquad \forall (i \in \mathrm{I}, m \in \mathrm{R})\ i \neq j \tag{12}$$

$$\sum_{n=1}^{n_{sd}} z_{jn} = 1 \qquad \forall (j \in \mathrm{O}) \tag{13}$$

$$\sum_{j=1}^{n_{ot}} q_{jjn} = 1 \qquad \forall (n \in \mathrm{S}) \tag{14}$$

$$\sum_{i=1}^{n_{ot}} q_{ijn} = z_{jn} \qquad \forall (j \in \mathrm{O}, n \in \mathrm{S}) \tag{15}$$

$$\sum_{i=1}^{n_{ot}} q_{jin} \leq z_{jn} \qquad \forall (j \in \mathrm{O}, n \in \mathrm{S})\ i \neq j \tag{16}$$

$$\sum_{j=1}^{n_{ot}} x_{ijk} = r_{ik} \qquad \forall (i \in \mathrm{I}, k \in \mathrm{P}) \tag{17}$$

$$\sum_{i=1}^{n_{it}} x_{ijk} = s_{jk} \qquad \forall (j \in \mathrm{O}, k \in \mathrm{P}) \tag{18}$$

$$\sum_{k=1}^{n_p} x_{ijk} \leq M \cdot g_{ij} \qquad \forall (i \in \mathrm{I}, j \in \mathrm{O}) \tag{19}$$

$$E_j \geq e_j - (C_j + TL) \qquad \forall (j \in \mathrm{O}) \tag{20}$$

$$T_j \geq (C_j + TL) - d_j \qquad \forall (j \in \mathrm{O}) \tag{21}$$

$$a_i + \left(UT \cdot \sum_{k=1}^{n_p} r_{ik} \right) + TL + TA \leq b_i \qquad \forall (i \in \mathrm{C}) \tag{22}$$

$$y_{im} \in \{0, 1\} \qquad \forall (i \in \mathrm{I}, m \in \mathrm{R}) \tag{23}$$

$$g_{ij} \in \{0, 1\} \qquad \forall (i \in \mathrm{I}, j \in \mathrm{O}) \tag{24}$$

$$z_{jn} \in \{0, 1\} \qquad \forall (j \in \mathrm{O}, n \in \mathrm{S}) \tag{25}$$

$$p_{ijm} \in \{0, 1\} \qquad \forall (i, j \in \mathrm{I}, m \in \mathrm{R})\ i \neq j \tag{26}$$

$$q_{ijn} \in \{0, 1\} \qquad \forall (i, j \in \mathrm{O}, n \in \mathrm{S})\ i \neq j \tag{27}$$

$$b_j, C_j, E_j, T_j \geq 0 \qquad \forall (j \in \mathrm{O}) \tag{28}$$

$$a_i, L_i \geq 0 \qquad \forall (i \in \mathrm{I}) \tag{29}$$

$$x_{ijk} \geq 0 \qquad \forall (j \in \mathrm{O}, k \in \mathrm{P}, i \in \mathrm{I}) \tag{30}$$

Objective function (1) is the overall tardiness and earliness of loading operations of outbound trucks. Constraint sets (2–3) ensure the precedence relation of inbound trucks assigned to the same receiving door. Constraint sets (4–5) ensure the precedence relation of outbound trucks assigned to the same shipping door. Constraint sets (6–7) guarantee that start time of unloading for an inbound truck and start time of loading for an outbound truck must be later than the arrival times of these trucks at the cross-docking center and the truck entering time if these trucks are the first trucks at receiving or shipping doors. Constraint set (8) connects the start time of loading for an outbound truck to the completion time of unloading for an inbound truck if any products are moved between the trucks. Constraint set (9) ensures that each inbound truck is assigned to only one door at the receiving dock. In constraint set (10), the variable p_{ii} becomes 1 if inbound truck i is at the beginning of the sequence at the assigned door. Constraint set (11) dictates that an inbound truck is immediately preceded by one inbound truck if it is assigned to a door. Constraint set (12) dictates that an inbound truck must be succeeded by at most one inbound truck if it is assigned to a door. Constraint set (13) indicates that each outbound truck is assigned to only one door in the shipping area. Constraint set (14) guarantees that only one outbound truck is assigned at the first sequence at each shipping door. In this constraint, the variable q_{jj} becomes 1 if outbound truck j is the first positioned truck at door n. Constraint set (15) dictates that an outbound truck is immediately preceded by one inbound truck if it is assigned to a shipping door. Constraint set (16) dictates that an outbound truck must be succeeded by at most one truck if it is assigned to a door. Constraint set (17) ensures that the total number of products type k that transfer from inbound truck i to all outbound trucks is exactly equal to the number of products type k that was already loaded into inbound truck i. Constraint set (18) ensures that the total number of products type k that transfer from all inbound trucks to outbound truck j is exactly equal to the number of products type k needed for outbound truck j. Constraint set (19) guarantees the exact relation between the x_{ijk} variables and the g_{ij} variables. Constraint set (20, 21) evaluates earliness and tardiness for outbound trucks. Constraint set (22) connects the start time of unloading to the starting time of loading for a compound truck. Constraints (23–30) impose binary and non-negativity conditions on the variables.

Test problems are required to compare optimum solutions obtained by implementing the MIP model in the GAMS 23.3, CPLEX 12.1 solver with the results of the suggested meta-heuristics to evaluate their performance. To measure the effectiveness of the meta-heuristic algorithms, small-sized and large-sized problems are generated.

3. Generation of test problems

The following factors are defined to generate test problems.

3.1. Factors for test problems.

Truck per door factor (μ): This is the average number of trucks loaded or unloaded per receiving or shipping door. The truck per door factor is defined as follows:

$$\mu = \frac{O}{S} \tag{31}$$

$$\mu = \frac{I}{R}. \tag{32}$$

In Eq. (31) and (32), μ, O, S, I and R represent the truck per door factor, the number of outbound trucks, the number of shipping doors, the number of inbound trucks and the number of receiving doors, respectively.

Average number of products per truck ($\bar{p}$): The average numbers of products per inbound and outbound trucks are defined as follows:

$$\overline{p_i} = \frac{\mathrm{T}_{UP}}{\mathrm{I}} \tag{33}$$

$$\overline{p_o} = \frac{\mathrm{T}_{LP}}{\mathrm{O}}. \tag{34}$$

In Eq. (33) and (34), p_i, p_o, T_{UP}, T_{UP}, T_{LP} show the average numbers of products per inbound truck and the average numbers of products per outbound truck, the total number of unloaded products and the total number of loaded products, respectively.

Time window (e-to-d) tightness factor (δ): This is used to control the range of the time window and is denoted as:

$$\delta = \beta - \alpha. \tag{35}$$

In Eq. (35), β, $\propto$ refer to the factors for upper bound and the lower bound of the time window, respectively. The bounds of the time window are denoted as:

$$d_j = \beta \cdot \pi \tag{36}$$

$$e_j = \propto \cdot \pi. \tag{37}$$

In Eq. (36) and (37), d_j, e_j, π represents the upper bound of the time window, the lower bound of the time window and the factor for the bounds of the time window, respectively.

The factor for the time window is denoted as:

$$\pi = \overline{p_l}UT + GD_j + \overline{p_o}LT \tag{38}$$

where $\overline{p_l}$, UT, GD_j, p_o, and LT denote the average number of products per inbound truck, unit unloading time, arrival time of an outbound truck, average number of products per outbound truck and unit loading time, respectively.

Maximum arrival time for all trucks: This is the upper bound of the arrival time of the trucks at the cross-docking terminal. The maximum arrival time is derived as:

$$2\overline{GD_j} = 2\bar{p}\mu\rho/(2 - \theta\mu\rho). \tag{39}$$

In Eq. (39), ρ represents the arrival time range factor and is calculated by the following formulation:

$$\rho = 2\overline{GD_j} / \hat{C}_{max}. \tag{40}$$

Where $\hat{C}_{max}$ denotes the estimated total operation time and is defined as:

$$\hat{C}_{max} = \left(2 - \theta \overline{GD_j} + \bar{p}\right)\mu. \tag{41}$$

In Eq. (41), θ shows the effect of the arrival times on total operation time. A similar formulation was used by Kaplan and Rabadi [24].

Effect of arrival times on total operation time (θ): Coefficient (θ), which considers the effect of the arrival times of the trucks on total operation time (assumed here to be the $\theta = 0.5$ time to the considerable effect of arrival times on total operation time), then estimates the $\hat{C}_{max}$. A similar approach was used by Lee and Pinedo [25] as well as Kaplan and Rabadi [24].

Arrival time range factor (ρ): Depending on the estimated C_{max}, the variability of arrival times is indicated by ρ. This is a criterion of how dispersed the arrival times are as compared with the estimated total operation time $\left(\hat{C}_{max}\right)$.

Arrival times of inbound $(\mathrm{GL_j})$ and outbound $(\mathrm{GD_j})$ trucks: Arrival times follow uniform distribution at the interval of $\left[0,\ 2\overline{GL_l}\right]$ and $\left[0,\ 2\overline{GD_l}\right]$ where $\overline{GD_l}$ is the trucks' arrival time average.

3.2. Data generation. The parameters used to generate test problems are derived by coding a data generation program in Python 3.4 software. The user interface is demonstrated in Fig. 3, where the user loads, saves and enters a new data set.

In the screenshot as shown in Fig. 3, the user is asked for the numbers of inbound, outbound, and compound trucks, the loading and unloading time, the truck changeover time, the truck transfer time, the transfer time of the goods as well as the numbers of receiving doors, shipping doors and types of goods. Based on the information provided, the system gener-

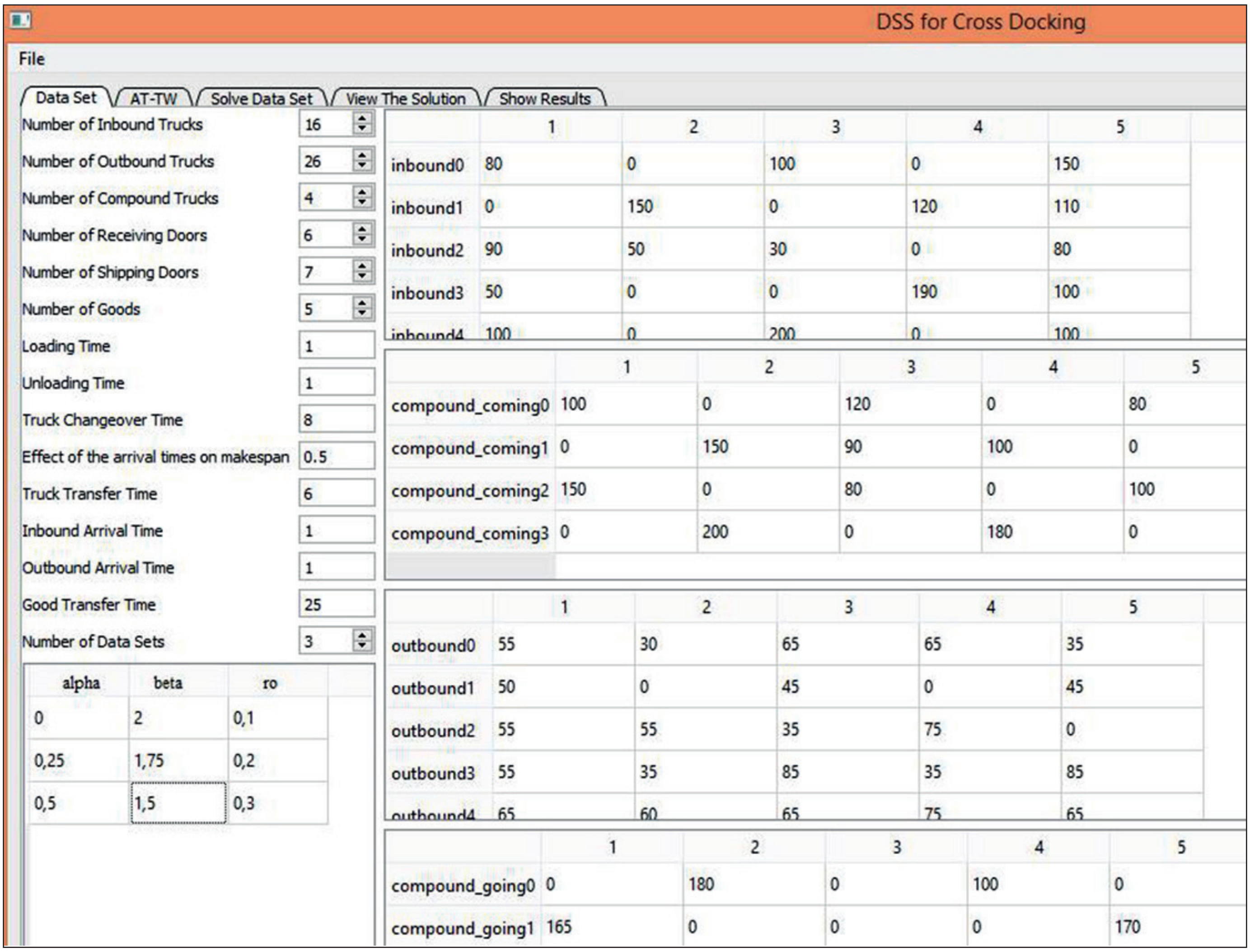

Fig. 3. Data generation interface

ates tables to enter the products in the inbound, outbound and compound trucks in terms of the product types and numbers. The numbered columns show the product numbers. There are 5 different types of products in the trucks in Fig. 3. These tables show the contents of the trucks for each product type.

The number of the unit for each product type which is loaded onto inbound trucks/outbound trucks is provided in the interface of the data generation. A decision support system (DSS) is designed for the truck scheduling problem for multi-door cross-docking centers. The DSS is a system consisting of a database, model and user interfaces. The database contains the data of the cross-docking center. DSS enables to use the proposed SA and TS algorithms and to schedule the trucks with user-friendly interfaces for decision makers in cross-docks.

4. Computational results

Optimum solutions for small problem sets are obtained by implementing the MIP model in the GAMS 23.3, CPLEX 12.1 solver and compared with the results of the suggested meta-heuristics to evaluate their performance. The truck scheduling problem is NP-hard, and it takes a long time to find optimum solutions for large-sized problems. Therefore, it is essential to use meta-heuristics to achieve suitable quality solutions in reasonable times. SA and TS algorithms are proposed to solve large-sized problems.

4.1. Meta-heuristics. Simulated annealing is a random search algorithm which works by emulating the physical annealing process of a material. It has the ability to escape from local optima via accepting the neighbor solution that is worse than the current solution with a probability in each temperature [26].

At iteration k of the procedure, $\mathrm{F}(\mathrm{Z}_\mathrm{k})$ refers to the value of the objective function for the corresponding sequence. The objective for the indicated problem is to minimize overall earliness and tardiness for outbound trucks. For a minimization problem, there is a current solution Z_k and candidate solution Z_c, selected from the neighborhood (Fig. 5). If $\mathrm{F}(\mathrm{Z}_\mathrm{c}) \geq \mathrm{F}(\mathrm{Z}_\mathrm{k})$, a move is made to Z_c with acceptance probability $\mathrm{P}(\mathrm{Z}_\mathrm{k}, \mathrm{Z}_\mathrm{c})$. The "t" is the cooling temperature decreased by each iteration. The initial temperature is 100°C and the cooling ratio is 90% for the SA algorithm. $\mathrm{F}(\mathrm{Z}_0)$ is referred to as the aspiration criterion and the value of the best solution obtained so far. The algorithm of the SA is as follows [26]:

Step 1: Let k = 1 and choose t_1
Step 2: Start with schedule Z_1 and set $\mathrm{Z}_0 = Z_1$
Step 3: Set Z_c (Generate neighborhood of Z_k)
If $F(Z_0) < F(Z_c) < F(Z_k)$, then $Z_{k+1} = Z_c$ and go to Step 4
If $F(Z_c) < F(Z_0)$ then $Z_0 = Z_{k+1} = Z_c$ and go to Step 4
If $F(Z_c) \geq F(Z_k)$, then generate $q_k = random\ (0, 1)$
If $q_k \leq P(Z_k, Z_c)$, $P(Z_k, Z_c) = exp\left(\frac{F(Z_k) - F(Z_c)}{t_k}\right)$ then set $Z_{k+1} = Z_c$.
Else set $Z_{k+1} = Z_k$ go to Step 4
Step 4: Set $t_{k+1} \leq t_k$ and set k = k + 1. Stop if stopping criteria are satisfied; otherwise go to Step 3.

Tabu search is a meta-heuristic procedure introduced by Glover [28, 29]. TS uses a neighborhood search technique by progressing iteratively from one solution to another until a stopping condition is satisfied. The size of a tabu list and the size of the neighborhood are used in 4 and 5 in small-sized test problems, respectively, and the size of a tabu list and the size of the neighborhood are used in 6 and 9 in large-sized test problems, respectively. The algorithmic description of the TS is as follows [26]:

Step 1: Let k = 1
Start with schedule Z_1 and set $Z_0 = Z_1$
Step 2: Set Z_c (Generate neighborhood of Z_k)
If the move $Z_{k\rightarrow}\ Z_c$ is forbidden by a change on the tabu list;
Set $Z_{k+1} = Z_k$ and go to step 3.
If the move $Z_{k\rightarrow}\ Z_c$ is not forbidden by any change on the tabu list;
Set $Z_{k+1} = Z_c$;
Add the reverse move to the top of the tabu list and remove the entry at the bottom.
If $F(Z_c) < F(Z_0)$, set $Z_0 = Z_c$; go to Step 3.
Step 3: Set k = k + 1
Stop if stopping criteria are satisfied; otherwise go to Step 2.

4.2. Initial solution and neighborhood-generation mechanism

Initial solution. In the first step, inbound trucks are sorted by arrival times from earliest to the latest. Then, the inbound truck which arrives earlier than the other trucks is assigned one by one to the next receiving doors. If certain inbound trucks have the same arrival time at the cross-docking center, the truck that has the most goods is assigned a receiving door first. Outbound trucks are assigned to the shipping doors in a similar manner. Figure 4 demonstrates an example of initial solution (Z_k) as a sequence of inbound and outbound truck assignments to the receiving and shipping doors.

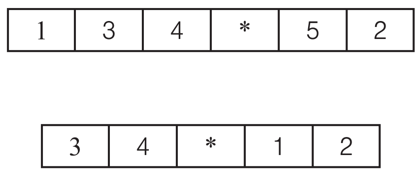

Fig. 4. Example of initial solution (Z_k) as a sequence of inbound and outbound truck assignments to receiving and shipping doors

Two receiving and two shipping doors are separated with "*" in Fig. 4, the inbound truck sequence at the receiving door 1 is 1–3–4 and at the receiving door 2 it is 5–2 (the arrival time sequence for inbound trucks is 1–5–3–2–4). Similarly, the outbound truck sequence at shipping door 1 is 3–4 and at the shipping door 2 it is 1–2 (the arrival time sequence for outbound trucks is 3–1–4–2).

the inbound truck sequence

4	5	*	3	2	*	1	current sequence
4	3	*	5	2	*	1	new sequence

the outbound truck sequence

1	3	5	*	2	*	4	current sequence
1	3	*	5	2	*	4	new sequence

Fig. 5. Current and new (Z_c) sequences for inbound and outbound trucks

Neighborhood-generation mechanism. The swap move is used to generate neighborhoods. While generating neighborhoods, two random numbers are derived and two trucks or a truck and a door "*" are interchanged by the corresponding number. The mechanism to generate a neighborhood (Z_c) for the sequences of inbound and outbound trucks is shown in Fig. 5.

In Fig. 5, two random numbers 2 and 4 are generated and two cells of the current sequence of inbound trucks are interchanged. In the new sequence, inbound truck 4 and 3 are unloaded at the first receiving door; inbound truck 5 and 2 are unloaded at the second receiving door and inbound truck 1 is unloaded at the third receiving door. In a similar way, two random numbers 3 and 4 are generated and two cells of the current sequence of outbound trucks are interchanged. In the new sequence, outbound trucks 1 and 3 are loaded at the first shipping door, outbound trucks 5 and 2 are loaded at the second shipping door and outbound truck 4 is loaded at the third shipping door.

Product assignment algorithm. At the neighborhood generation stage for meta-heuristics, during the decision-making regarding where and when inbound/outbound trucks will be unloaded/loaded, decisions regarding product assignment from inbound trucks to outbound trucks are made. For this purpose, the following algorithm is designed and coded via the Python 3.4 software.

In the algorithm, t, m and y refer to the decision time for truck-door assignment, the number of outbound trucks and the number of shipping doors, respectively. w_j is the waiting time of outbound truck j while loading. The start time of loading for outbound trucks can be defined as in the $b_j \geq max\{GD_j,\ a_i + (LT \cdot \sum_{k=1}^{n_p} s_{ik}) + CT\}$ formula. In this expression, the start time of loading for an outbound truck is equal to or later than the arrival time of the outbound truck at the cross-docking center and the finish time of the loading for previous truck at the same door. While specifying the start time of loading for outbound trucks, the parameters that need to be controlled are the arrival time of outbound trucks at the cross-docking center, the finish time of loading for the previous truck at the same door, the availability of shipping doors, the ready time when all products need to be loaded onto the outbound truck and the time window of the outbound truck.

Product assignment algorithm

Step 1: Set k = 1 and set initial solution and go to Step 3
Step 2: Set N_k (Generate neighborhood => Derive two random numbers and interchange two trucks by the corresponding random numbers.) Set $Z_{jn} \in N_k$
Step 3: Check Z_{jn} (schedule for the shipping doors)
For j = 1 to m
  For n = 1 to y
  If t > GD_j **then** calculate $C_j = \{b_j + (LT \cdot \sum_{k=1}^{n_p} s_{jk})\}$
    If $C_j \leq e_j$
      If goods are ready to load, **then** start loading
      Otherwise wait
      EndIf
    EndIf

    If $e_j \leq C_j < d_j$
      If goods are ready to load, **then** start loading
      Else => reserve the goods for loading
      EndIf
    EndIf
    If $C_j \geq d_j$
      If goods are ready to load **then**
      start loading and set $C_j = \{b_j + (LT \cdot \sum_{k=1}^{n_p} s_{jk})\}$
     Else reserve and load the goods as soon as they are ready and set $C_j = \{b_j + w_j + (LT \cdot \sum_{k=1}^{n_p} s_{jk})\}$
     EndIf
    EndIf
   EndIf
  Next n
Next j
Endfor
Step 4: set k = k + 1. If k = N then stop; otherwise go to Step 2.

4.3. Results. The different arrival times for inbound and outbound trucks and the time windows for outbound trucks are generated by using the indicated data generation program. Parameters are dependent on the problem instances in terms of the three levels of lower and upper bound coefficients of the time window, namely, optimistic ($\propto = 0$, $\beta = 2$), possible ($\propto = 0{,}25$; $\beta = 1{,}75$), and pessimistic ($\propto = 0{,}5$, $\beta = 1{,}5$), for both small and large-sized test problems. There are three levels of the arrival time range factor ($\rho = 0.1$; 0.2; 0.3) for small and large-sized test problems. The interval of arrival time of the trucks at the cross-docking center remains very narrow because there is no great difference between arrival times of the trucks at the cross-docking terminal.

Small and large-sized problem sets are randomly generated considering the total number of inbound and outbound trucks. Small test problems with less than or equal to 7 inbound and 7 outbound trucks are solved by using GAMS 23.3. The optimum

Table 1
Comparison of performance of SA and TS for small-sized problems in terms of objective function value and computational time

Test problems	I	O	R	S	(∝, β)	Objective value				Computational time (s)		
						ρ	CPLEX	SA	TS	CPLEX	SA	TS
1	4	4	2	1	(0; 2)	0.1	0	0	0	0.57	17.00	84.00
2	4	4	2	1	(0.25; 1.75)	0.2	133	133	133	0.55	18.00	93.00
3	4	4	2	1	(0.5; 1.5)	0.3	237	237	237	0.64	17.00	91.00
4	4	5	2	2	(0; 2)	0.1	21	21	21	1.47	23.00	112.00
5	4	5	2	2	(0.25; 1.75)	0.2	65	65	65	2.28	21.00	114.00
6	4	5	2	2	(0.5; 1.5)	0.3	473	481	481	3.53	22.00	118.00
7	4	6	3	3	(0; 2)	0.1	0	0	0	0.22	21.00	122.00
8	4	6	3	3	(0.25; 1.75)	0.2	0	0	0	0.52	23.00	124.00
9	4	6	3	3	(0.5; 1.5)	0.3	47	47	47	0.96	24.00	128.00
10	5	5	3	2	(0; 2)	0.1	0	0	0	0.38	19.00	117.00
11	5	5	3	2	(0.25; 1.75)	0.2	15	15	15	3.56	20.00	108.00
12	5	5	3	2	(0.5; 1.5)	0.3	374	374	374	3.38	21.00	110.00
13	6	5	2	3	(0; 2)	0.1	0	0	0	32.09	28.00	152.00
14	6	5	2	3	(0.25; 1.75)	0.2	261	261	261	74.70	30.00	155.00
15	6	5	2	3	(0.5; 1.5)	0.3	712	730	730	476.67	30.00	158.00
16	7	7	2	3	(0; 2)	0.1	459	478	478	1650.04	23.00	153.00
17	7	7	2	3	(0.25; 1.75)	0.2	483	483	483	1483.16	27.00	146.00
18	7	7	2	3	(0.5; 1.5)	0.3	985	1017	1020	14354.84	29.00	157.00
										Average	**22.94**	**124.44**

solutions determined by the CPLEX 12.1 solver of small-sized problems are compared with the solutions of TS and SA in Table 1. The termination criterion is the iteration number and it is same for each algorithm. The initial temperature is 100°C and the cooling ratio is 90% for the SA algorithm. As a result of computational experiments, tabu parameters are obtained for the TS algorithm. The size of the tabu list and the size of the neighborhood are used in 4 and 5 in small-sized test problems, respectively. Each test problem is solved ten times by using both algorithms. Meta-heuristics algorithms are coded in Python 3.4 software. All experiments are performed on a PC with 3.0 GHz Intel Core i7 processor and 12 GB RAM.

Table 1 includes the results of problems with less than or equal to 7 inbound trucks and 7 outbound trucks. The optimum solutions obtained by the CPLEX solver are compared with the solutions of SA and TS algorithms. The optimum solutions obtained using the CPLEX solver and the relative percentage deviation (RPD) of the meta-heuristic algorithms are shown in Table 2. The optimum solutions are found by using both meta-heuristics for 14 test problems from 18 test problems. The relative deviation of SA is 0.65 and TS is 0.66 for small test problems. Both meta-heuristics have almost the same RPD values.

According to Table 1, it takes a long time to find an optimum solution for a problem greater than 6 inbound and 6 outbound trucks in the CPLEX solver. When the number of trucks increases, computational time increases dramatically. Lee, Kim and Joo [19] point out that the optimization tool does not give results for problems over 6 inbound and 6 outbound trucks exactly because of the long computational time. The results in Table 1 support the above-indicated study. Objective values of the SA and TS algorithms and computational times show that SA and TS provide suitable results in a reasonable time. They can thus be used for large-sized problems.

Table 2
RPD (%) values for small-sized problems

Test problems	CPLEX Optimal Solution	RPD (%)		Test problems	CPLEX Optimal Solution	RPD (%)	
		SA	TS			SA	TS
1	0	0.00	0.00	10	0	0.00	0.00
2	133	0.00	0.00	11	15	0.00	0.00
3	237	0.00	0.00	12	374	0.00	0.00
4	21	0.00	0.00	13	0	0.00	0.00
5	65	0.00	0.00	14	261	0.00	0.00
6	473	1.69	1.69	15	712	2.53	2.53
7	0	0.00	0.00	16	459	4.14	4.14
8	0	0.00	0.00	17	483	0.00	0.00
9	47	0.00	0.00	18	985	3.25	3.55
					Average	0.65	0.66

$$\text{RPD}\,(\%) = \frac{\textit{Solution of the meta heuristic} - \textit{Optimal solution}}{\textit{Optimal solution}} \times 100$$

Table 3
Comparison of performance SA and TS for large-sized problems in terms of objective function value and computational time

Test problems	I	O	R	S	(∝, β)	ρ	Objective value		Computational time (s)	
							SA	TS	SA	TS
1	15	20	5	5	(0; 2)	0.1	2329	2027	76.00	654.00
2	15	20	5	5	(0.25; 1.75)	0.2	2675	2924	83.00	666.00
3	15	20	5	5	(0.5; 1.5)	0.3	3933	3290	89.00	711.00
4	20	20	6	5	(0; 2)	0.1	1786	1692	63.00	577.00
5	20	20	6	5	(0.25; 1.75)	0.2	2264	1917	70.00	565.00
6	20	20	6	5	(0.5; 1.5)	0.3	3911	3142	74.00	649.00
7	25	20	7	6	(0; 2)	0.1	1750	1362	75.00	641.00
8	25	20	7	6	(0.25; 1.75)	0.2	1507	1956	76.00	663.00
9	25	20	7	6	(0.5; 1.5)	0.3	2836	2505	73.00	681.00
10	20	30	6	7	(0; 2)	0.1	3043	3389	106.00	810.00
11	20	30	6	7	(0.25; 1.75)	0.2	3701	3870	113.00	832.00
12	20	30	6	7	(0.5; 1.5)	0.3	6260	5735	117.00	860.00
13	25	30	7	7	(0; 2)	0.1	3456	3143	97.00	700.00
14	25	30	7	7	(0.25; 1.75)	0.2	4024	3351	110.00	823.00
15	25	30	7	7	(0.5; 1.5)	0.3	5306	5473	108.00	866.00
16	30	30	8	7	(0; 2)	0.1	4055	3892	103.00	814.00
17	30	30	8	7	(0.25; 1.75)	0.2	3906	3956	98.00	812.00
18	30	30	8	7	(0.5; 1.5)	0.3	4635	3566	118.00	822.00
19	30	35	10	9	(0; 2)	0.1	2960	2717	109.00	812.00
20	30	35	10	9	(0.25; 1.75)	0.2	3114	3702	123.00	824.00
21	30	35	10	9	(0.5; 1.5)	0.3	5572	5025	116.00	823.00
22	35	35	10	10	(0; 2)	0.1	3272	3385	122.00	970.00
23	35	35	10	10	(0.25; 1.75)	0.2	4985	4137	115.00	940.00
24	35	35	10	10	(0.5; 1.5)	0.3	5969	5234	132.00	954.00
25	35	40	11	10	(0; 2)	0.1	3322	2915	112.00	904.00
26	35	40	11	10	(0.25; 1.75)	0.2	3774	3303	106.00	928.00
27	35	40	11	10	(0.5; 1.5)	0.3	5805	5646	114.00	968.00
28	40	40	11	11	(0; 2)	0.1	4471	3382	120.00	955.00
29	40	40	11	11	(0.25; 1.75)	0.2	6042	5278	135.00	946.00
30	40	40	11	11	(0.5; 1.5)	0.3	6810	6447	114.00	958.00
						Average	3915.77	3612.03	102.23	804.27

Data sets for large-sized problems are used for comparing the relative performance of SA and TS in Table 3. The initial temperature is 100°C and the cooling ratio is 90% for the SA algorithm. As a result of computational experiments, tabu parameters are obtained for the TS algorithm. The size of the tabu list and the size of the neighborhood are used in 6 and 9, respectively. Each test problem is solved ten times using both algorithms.

In Table 3, the algorithms are compared with each other to show their performance. TS outperforms SA for the large-sized test problems. It is observed that TS gives better results than SA when the number of trucks increases. However, the SA algorithm consumes less computational time than the TS algorithm when the iteration number is the same for each algorithm. Table 4 shows the relative percentage deviation (RPD) of the meta-heuristic algorithms.

Table 4 shows that the RPD values of the SA algorithm are greater than the RPD values of the TS algorithm. While the RPD value average of the SA algorithm is 11.18%, the RPD value average of the TS algorithm is only 2.73%. The RPD

Table 4
RPD (%) values for large-sized problems

Test problems	Best solution of the two meta-heuristics	RPD (%) SA	RPD (%) TS	Test problems	Best solution of the two meta-heuristics	RPD (%) SA	RPD (%) TS
1	2027	14.90	0.00	16	3892	4.19	0.00
2	2675	0.00	9.31	17	3906	0.00	1.28
3	3290	19.54	0.00	18	3566	29.98	0.00
4	1692	5.56	0.00	19	2717	8.94	0.00
5	1917	18.10	0.00	20	3114	0.00	18.88
6	3142	24.47	0.00	21	5025	10.89	0.00
7	1362	28.49	0.00	22	3272	0.00	3.45
8	1507	0.00	29.79	23	4137	20.50	0.00
9	2505	13.21	0.00	24	5234	14.04	0.00
10	3043	0.00	11.37	25	2915	13.96	0.00
11	3701	0.00	4.57	26	3303	14.26	0.00
12	5735	9.15	0.00	27	5646	2.82	0.00
13	3143	9.96	0.00	28	3382	32.20	0.00
14	3351	20.08	0.00	29	5278	14.48	0.00
15	5306	0.00	3.15	30	6447	5.63	0.00
					Average	11.18	2.73

$$\text{RPD}\ (\%) = \frac{\textit{Solution of the meta heuristic} - \textit{Best solution of the two meta-heuristics}}{\textit{Best solution of the two meta-heuristics}} \times 100$$

values of the two algorithms are shown in Fig. 6 with interval plots at a 95% confidence level.

Graph (a) in Fig. 6 shows that there are statistically significant differences between RPD values of the SA and TS algorithms. Graph (b) and graph (c) also indicate the differences between two algorithms with the different time windows and the number of trucks, respectively. These results show that TS manifests better performance for the truck scheduling problem with any truck number and any ($\propto$, β) values of time windows.

The results indicate that the computational time increases when the number of trucks increases for each algorithm. Figure 7 shows the changes in the average computational times of SA and TS in terms of the number of trucks. The problem has several characteristics in common with the parallel machine scheduling problem. Similar results are shown in the parallel machine scheduling study [27].

Figure 7 indicates the relation between the number of trucks and the computational time by comparing the algorithms. It can be observed that SA takes less time than TS. For large-sized problems, SA provides suitable results with an average computational time (102.23 seconds) that is 87% shorter than that of the TS.

The decision maker generates the schedules by using SA or TS algorithms in DSS. He/she can select an appropriate schedule by comparing the results of the algorithms. Fig. 7 shows a screenshot of the system for solving the problem (Fig. 8).

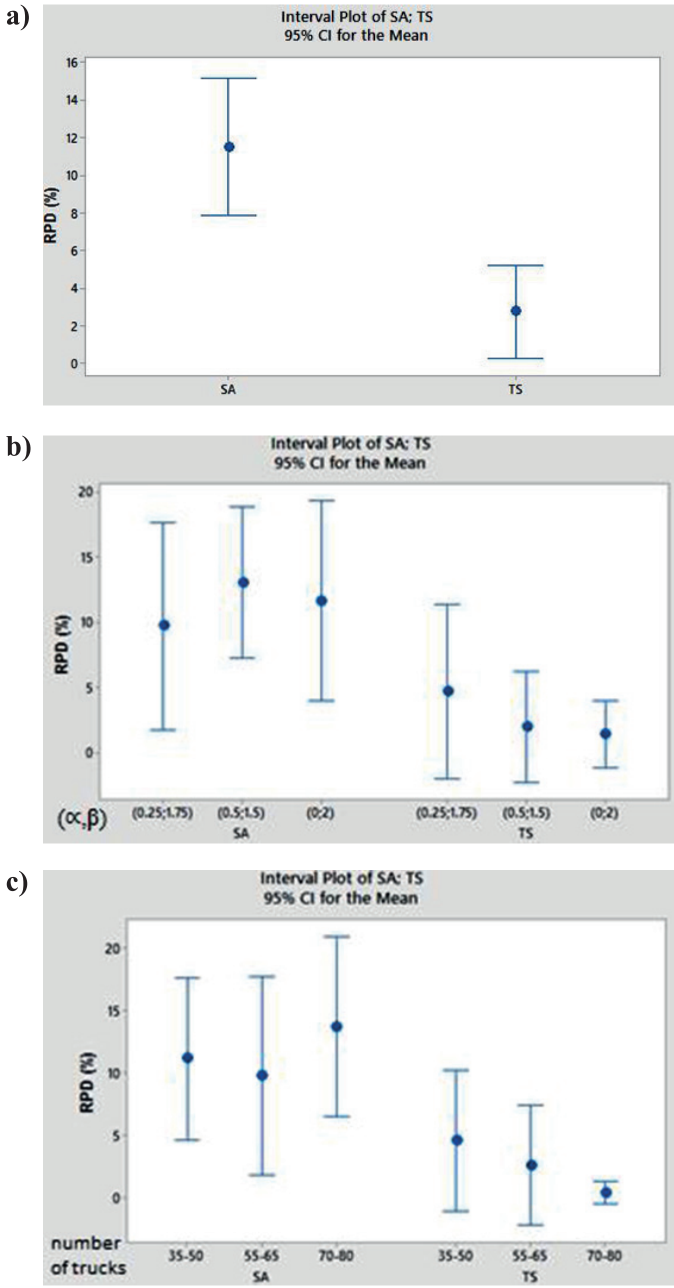

Fig. 6. RPD interval plots for SA and TS algorithm (a) Total (b) Group by time windows (c) Group by the number of trucks

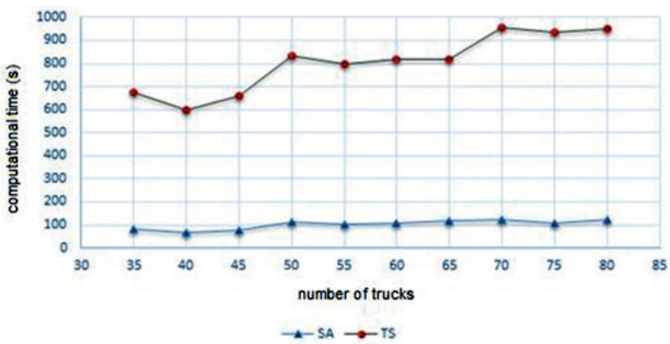

Fig. 7. Computational times versus the number of trucks

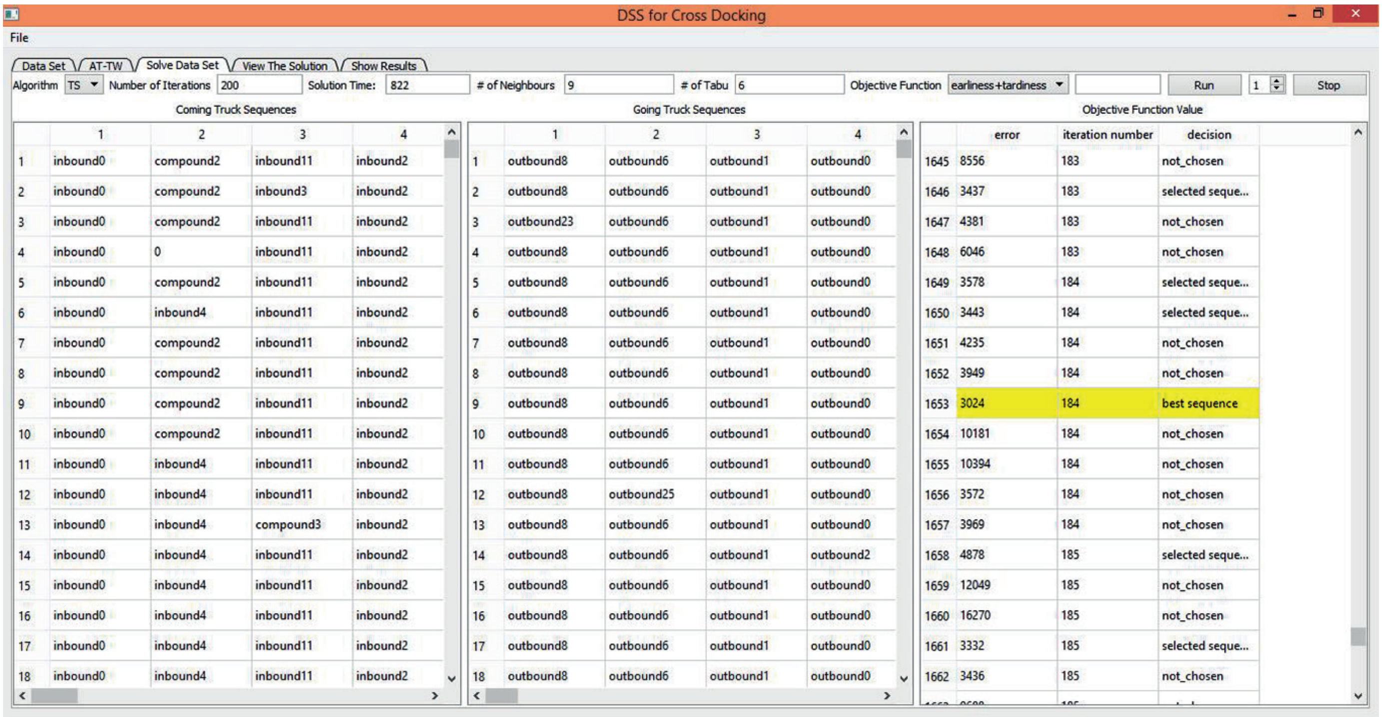

Fig. 8. Screenshot of the system for solving the problem

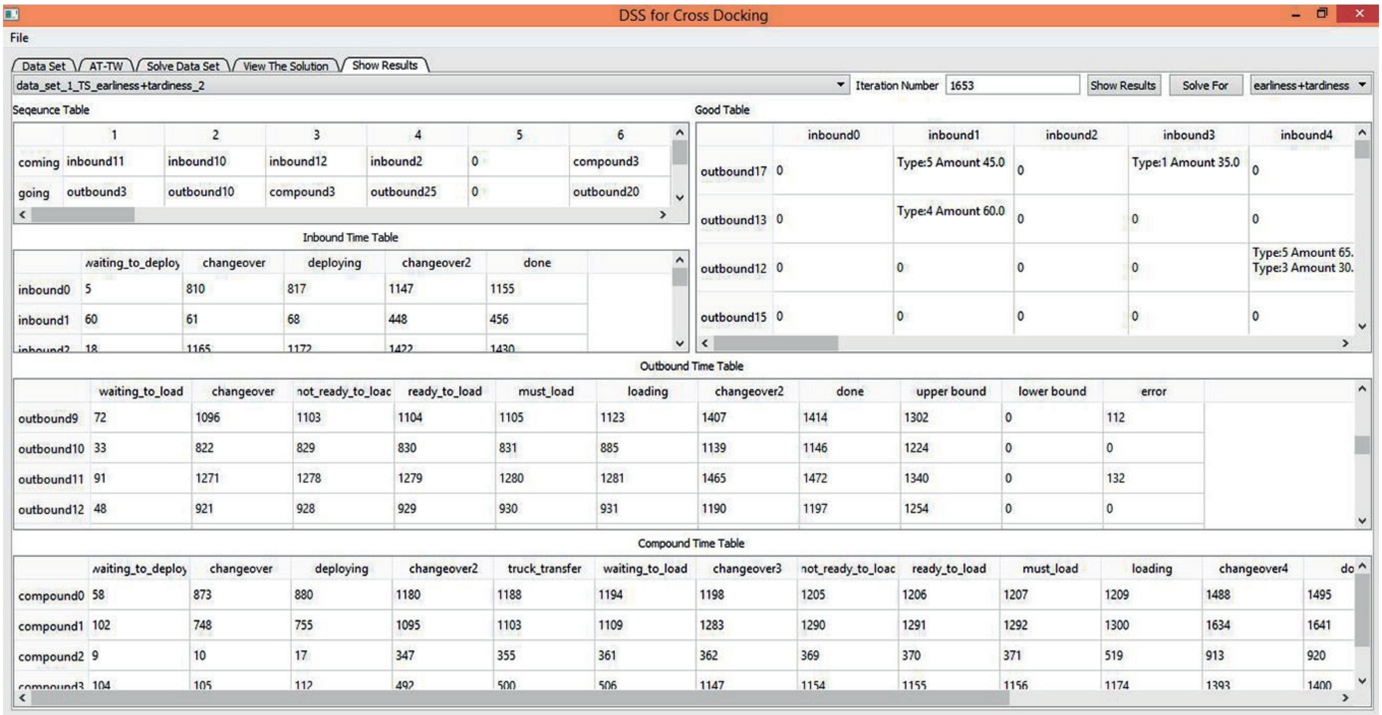

Fig. 9. Solution for overall earliness and tardiness

The user can set the algorithm's parameters such as the temperature and cooling ratio or the size of the tabu list and neighborhood. In Fig. 8, the highlighted rows show the best solutions iteration by iteration. The user can choose the best solution so far from the last highlighted row for overall earliness and tardiness. The user can monitor the number of products of type k that are transferred from inbound truck i to outbound truck j, the start time of unloading for inbound truck i, the completion time of unloading for inbound truck i, the start time of loading for outbound truck j, the completion time of loading for outbound truck j, and the sequences of the trucks at the doors as shown in Fig. 9. The outbound time table shows which outbound trucks fit into the time windows. If an outbound truck doesn't fit the time window, an error (earliness or tardiness) value is shown in the last column (Fig. 9).

In multi-door cross-docking systems, it is essential to coordinate activities such as unloading and loading and product assignment. The proposed DSS makes it possible to solve the truck scheduling problem by assigning products from inbound trucks to outbound trucks.

5. Conclusion

This paper focuses on a truck scheduling problem within pre-defined time windows in multi-door cross-docking centers. Since the punctuality and accurateness of product deliveries affects the performance of the cross-docking system, the predominant objective is to minimize earliness and tardiness. There are different arrival times for all trucks and due time intervals of outbound trucks that should be taken from customers. In this study, the problem is formulated as a mixed integer programming model to find an optimum solution. The model is used to evaluate the performance of the meta-heuristic algorithms for small-sized problems.

The numbers of product types loaded onto inbound trucks and needed for outbound trucks are known by using the RFID technology at cross-docking centers. Therefore, it is possible to assign products from inbound trucks to outbound trucks by using the information about products on inbound trucks. Product assignment is then effected by means of using a proposed product-assignment algorithm. The product -assignment algorithm uses the information obtained as a parameter to be subjected to the RFID technology.

The truck scheduling problem is NP-hard. SA and TS meta-heuristics are proposed to solve the large-sized problems. Experimental results show that TS manifests better performance for the truck scheduling problem with any truck number and any ($\propto$, β) values of time windows. According to the relation between the number of trucks and computational time when comparing the algorithms, it can be noted that SA takes less time than TS for the same iteration number.

In the cross-docking systems, it is required to coordinate activities such as unloading and loading and product assignment. A DSS is designed and proposed for solving truck scheduling problems to minimize overall earliness and tardiness for outbound trucks within time windows. The proposed DSS makes it possible to solve the truck scheduling problem by assigning products from inbound trucks to outbound trucks. The decision maker generates the schedules by using SA or TS algorithms in DSS. He/she can select an appropriate schedule by comparing the results of the algorithms.

For future work, the cross-docking system with time windows will be able to be modelled as multi-objective. Product and door constraints will then be added to the model.

Acknowledgements. This work is supported by the Committee of Eskisehir Osmangazi University Scientific Research Projects under the Project Code 2015–734, titled "Door Assignment and Truck Scheduling in Cross-Docking".

References

[1] M.U. Apte and S. Viswanathan, "Effective cross docking for improving distribution efficiencies", *International Journal of Logistics: Research and Applications*, 3, 291–302 (2000).

[2] B. Vahdani and M. Zandieh, "Scheduling trucks in cross-docking systems: Robust meta-heuristics", *Computers & Industrial Engineering*, 58, 12–24 (2010).

[3] D.L. McWilliams, P.M. Stanfield, and C.D. Geiger, "The parcel hub scheduling problem: A simulation-based solution approach", *Computers & Industrial Engineering*, 49, 393–412 (2005).

[4] W. Yu and P.J. Egbelu, "Scheduling of inbound and outbound trucks in cross docking systems with temporary storage", *European Journal of Operational Research*, *184*, *377–396 (2008).*

[5] F. Chen and C.Y. Lee, "Minimizing the makespan in two-machine cross-docking flow shop problem", *European Journal of Operational Research*, 193, 59–72 (2009).

[6] A.R. Boloori Arabani, S.M.T. Fatemi Ghomi and M. Zandieh, "Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage", *Expert Systems with Applications*, 38, 1964–1979 (2011).

[7] J. Van Belle, P. Valckenaers, and D. Cattrysse, "Cross-docking: State of the art", *Omega*, 40, 827–846 (2012).

[8] A.R. Boloori Arabani, S.M.T. Fatemi Ghomi, and M. Zandieh, "A multi-criteria cross-docking scheduling with just-in-time approach", *International Journal of Advanced Manufacturing Technology*, 49, 741–756 (2010).

[9] 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*, 46, 650–666 (2010).

[10] A.R. Boloori Arabani, M. Zandieh, and S.M.T. Fatemi Ghomi, "A cross-docking scheduling problem with sub-population multi-objective algorithms", *International Journal of Advanced Manufacturing Technology*, 58, 741–761 (2012).

[11] A. Mohtashami, "Scheduling trucks in cross docking systems with temporary storage and repetitive pattern for shipping trucks", *Applied Soft Computing*, 36, 468–486 (2015).

[12] A. Amini and R. Tavakkoli-Moghaddam, "A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks", *Computers and Industrial Engineering*, 96, 181–190 (2016).

[13] D.L. McWilliams, P.M. Stanfield, and C.D. Geiger, "Minimizing the completion time of the transfer operations in a central parcel consolidation terminal with unequal-batch-size inbound trailers", *Computers & Industrial Engineering*, 54(4),709–720 (2008).

[14] G. Alpan, R. Larbi, and B. Penz, "A bounded dynamic programming approach to schedule operations in a cross docking platform", *Computers and Industrial Engineering*, 60, 385–396 (2011).

[15] N. Boysen, D. Briskorn, and M. Tschöke, "Truck scheduling in cross-docking terminals with fixed outbound departure", *OR Spectrum*, 35, 479–504 (2013).

[16] D. Konur and M.M. Golias, "Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach", *Transportation Research Part E*, 49, 71–91 (2013).

[17] 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*, 141, 212–229 (2013).
[18] N. Boysen, "Truck scheduling at zero-inventory cross docking terminals", *Computers & Operations Research*, 37, 32–41 (2010).
[19] 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*, 39, 12975–12983 (2012).
[20] C.M. Joo and B.S. Kim, " Scheduling compound trucks in multi-door crossdocking terminals", *International Journal of Advanced Manufacturing Technology*, 64, 977–988 (2013).
[21] J. Van Belle, P. Valckenaers, G.V. Berghe, and D. Cattrysse, "A tabu search approach to the truck scheduling problem with multiple docks and time windows", *Computers and Industrial Engineering*, 66, 818–826 (2013).
[22] M.T. Assadi and M. Bagheri, "Scheduling trucks in a multiple-door cross docking system with unequal ready times", *European Journal of Industrial Engineering*, 10, 103–125 (2016).
[23] N. Boysen and M. Fliedner, "Cross dock scheduling: Classification, literature review and research agenda", *Omega*, 38, 413–422 (2010).
[24] S. Kaplan and G. Rabadi, "Exact and heuristic algorithms for the aerial refueling parallel machine scheduling problem with due date-to-deadline window and ready times", *Computers & Industrial Engineering*, 62, 776–785 (2012).
[25] Y. H. Lee and M. Pinedo, "Theory and Methodology: Scheduling jobs on parallel machines with sequence dependent setup times", *European Journal of Operational Research*, 100, 464–474 (1997).
[26] M. Pinedo, *Scheduling: Theory*, *Algorithms*, *and Systems*. (2nd ed.). Prentice Hall, 2002.
[27] I. Saricicek and C. Celik, "Two meta-heuristics for parallel machine scheduling with job splitting to minimize total tardiness", *Applied Mathematical Modelling*, 35, 4117–4126 (2011).
[28] F. Glover, *Tabu Search-Part I*, ORSA J. Comput. 1 (3), 190–206 (1989).
[29] F. Glover, *Tabu Search-Part II*, ORSA J. Comput. 2 (1) 4–32 (1990).