



A customized genetic algorithm for solving multi-period cross-dock truck scheduling problems



Kaveh Khalili-Damghani^a, Madjid Tavana^{b,c,*}, Francisco J. Santos-Arteaga^d, Mahdokht Ghanbarzad-Dashti^e

^a Department of Industrial Engineering, South-Tehran Branch, Islamic Azad University, Tehran, Iran

^b Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA 19141, USA

^c Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, D-33098 Paderborn, Germany

^d Faculty of Economics and Management, Free University of Bolzano, Bolzano, Italy

^e Department of System and Industrial Engineering, Industrial Management Institute, Tehran, Iran

ARTICLE INFO

Article history:

Received 1 February 2017

Received in revised form 15 April 2017

Accepted 10 May 2017

Available online 15 May 2017

Keywords:

Cross-docking

Truck scheduling

Supply chain transportation

Evolutionary computation

Genetic algorithm

ABSTRACT

Cross-docking is a logistics strategy for direct distribution of products from a supplier or manufacturing plant to a customer or retail outlet with little or no handling and storage time. The classical cross-docking models are used to find the optimal inbound/outbound truck schedule that minimizes the total operational time. We propose a new multi-period cross-docking model with multiple products, due dates, variable truck capacities, and temporary warehouse. The problem is formulated as mixed-integer programming and an evolutionary computation approach based on a genetic algorithm (GA) is designed to solve it. The structure of the chromosomes, the operators, and the constraint handling strategy are specifically designed for multi-period problems. Several test instances have been generated to compare the performance of the proposed GA with that of a branch and bound solution procedure. Moreover, a comprehensive statistical analysis is conducted to illustrate the performance efficacy of the proposed GA relative to the branch and bound algorithm. This analysis reveals that the GA provides a substantial decrease in the computational burden when compared to the branch and bound algorithm.

© 2017 Elsevier Ltd. All rights reserved.

1. Introduction: cross-docking basics

Cross-docking is a logistics management concept that refers to moving products from a supplier or manufacturing plant to a customer or retail chain with little or no material handling in between. Cross-docking reduces material handling and the need to store the products in a warehouse. Generally, cross-docking facilities include five major functions of warehousing: collection from suppliers, receiving, consolidation (i.e., the process of storage and order picking), shipping, and delivery to customers. Cross-docking minimizes the storage and order picking functions while still allowing of a warehouse to perform its receiving and shipping functions [25]. The objective in cross-docking is to transfer shipments directly from suppliers and incoming trailers to outgoing trailers and customers without storage in between.

* Corresponding author at: Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, PA 19141, USA.

E-mail addresses: k_khalili@azad.ac.ir (K. Khalili-Damghani), tavana@lasalle.edu (M. Tavana), fsantosarteaga@unibz.it (F.J. Santos-Arteaga), mahdokhtgh@gmail.com (M. Ghanbarzad-Dashti).

URL: <http://tavana.us/> (M. Tavana).

Several potential problems may arise in the cross-docks caused by the different door environments, operational characteristics, and objective functions [5]. These problems regard mainly location [22,27], vehicle routing [9,12,15,23,24,28], and truck scheduling [5]; [33]. In this study, we will focus on the truck scheduling problem.

Scheduling is a fundamental part of cross-docking, because a correct priority order among inbound and outbound trucks reduces the expenses incurred by the firm and improves the efficiency of the shipment process. There are many methods for finding the optimum cross-docking schedule. Most of the articles on temporary storage deal with truck scheduling in cross-docking. This illustrates the major importance that this subject has for reducing operational expenses related to the transference of products in the warehouse, preventing accumulation and disorder in internal transferences, reducing the time of products' storage in the warehouses, and minimizing the tardiness (delays) in the delivery of products to the customers.

Truck scheduling is usually defined for short periods of time and the literature on the topic can be categorized into three main groups. The first group generally considers a unique entrance and a unique exit, which simplifies the cross-docking process

considerably. In the second group, scheduling is defined only for the entry process and it is assumed that an assignment for the exit one has already been made. This type of model has plenty of applications in postal systems, where outbound trucks move toward their destinations following a fixed schedule. In the third group, scheduling is simultaneously defined for the receiving and the shipping dock. This is the group whose models are more suitable to analyze cross-docking problems.

Plenty of researchers have examined simple cross-docking and scheduling processes, where the scheduling problem is transformed into a problem of inbound and outbound trucks. However, this type of cross-docking model is not sufficiently close to real-life situations and, therefore, it is not used in practice.

Yu and Egbelu [35] investigated a scheduling problem in cross-docking in its simplest form, i.e., when there is one strip door and one stack door. Their goal was to minimize the time, similarly to a double-machine (two-stage) workshop system where the products could be interchanged. Their model was a linear complex programming one, requiring heuristic methods to obtain solutions for large-scale problems. Yu and Egbelu [35] analyzed 32 cross-docking models, which were obtained by combining three main criteria (number of docks, docking sequence of trucks, existence or non-existence of temporary warehouse). We illustrate several possible cross-docking combinations from the model proposed by Yu and Egbelu [35] in Fig. 1.

Vahdani and Zandiyeh [32] solved the problem proposed by Yu and Egbelu [35] using 5 different meta-heuristic algorithms. They tried to improve the results obtained by each algorithm by temporary storing products on shipping platforms to shorten the scheduling time. The algorithms used by these authors included simulated annealing, variable neighborhood search as well as genetic, tabu search and electromagnetic algorithms. Different meta-heuristic algorithms have also been effectively used for solving larger problems. In this regard, Boloori Arabani et al. [4] used 5 meta-heuristic algorithms, namely, particle swarm optimization, genetic, tabu search, ant colony and differential evolutionary algorithms, to solve the problem proposed by Yu and Egbelu [35].

Boloori Arabani et al. [3] revisited the problem proposed by Yu and Egbelu [35]. They assumed that outbound trucks should complete their jobs in a particular period of time. Their objective was to minimize earliness and tardiness of the outbound trucks. They used three meta-heuristic algorithms: genetic algorithm (GA), particle swarm optimization algorithm, and differential evolutionary algorithm. Vahdani and Zandiyeh [32] solved this problem without taking into account the temporary storage assumption. They assumed that products were directly transferred from the inbound truck to the outbound trucks and used a GA and an electromagnetic algorithm to solve the problem. Soltani and Sadjadi [30] used

the combinatory algorithms of simulated annealing and variable neighborhood search to solve this problem.

An examination of previous research shows that a great deal of research has focused on the scheduling aspects of cross-docking. The model proposed by Yu and Egbelu [35] is one of the major works written in this area of research. However, one of the deficiencies of this model is that it does not take into consideration the capacities of the trucks or include due dates. It should be noted that several companies have successfully implemented their cross-docking systems even before a generic framework for understanding and designing cross-docking systems would appear in the technical literature [10,11].

In this regard, Rohrer [26] addressed several cross-docking problems through simulation. For example, he used simulation to establish failure strategies that accounted for cross-docking problems before they were encountered. Similarly, Magableh et al. [19] proposed a generic simulation model to represent the main operations within a cross-docking facility.

Bartholdi and Gue [2] also surveyed the cross-docking system. They assumed that cross-dock doors were permanently assigned as strip or stack doors and proposed different layouts based on the number of doors, the ratio of strip to stack doors, and the distribution of material flows inside. Chen and Lee [8] considered the truck scheduling problem as a two-machine flow-shop problem, where the first machine was used for unloading and unpacking operations of the inbound trucks and the second machine was assigned to the collecting and loading operations of the outbound trucks.

The idea of mapping the cross-docking truck scheduling problem to the flow-shop scheduling problem was extended by Song and Chen [31] to cover multiple strip doors. They defined the problem as a two stage cross-docking logistics optimization problem where the first stage represented the inbound flow with multiple unrelated parallel machines and the second stage represented the outbound flow with one machine. Song and Chen [31] introduced a mathematical model and solved it for small scale instances. They also proposed two heuristics based on Johnson's rule to observe the performance of moderate and large scale instances.

In this study, we consider the problem proposed by Yu and Egbelu [35] and propose two models for cross-dock truck scheduling. In the first model, the capacity of the trucks is taken into consideration and inbound and outbound trucks have been assumed to have a given time-window for scheduling. In the second model, the assumptions of the first model are considered again in a multi-period planning horizon and a GA is proposed to solve several benchmark instances of the problem. We also compare the performance of the GA proposed in this study with that of the exact method of branch and bound (coded in LINGO software).

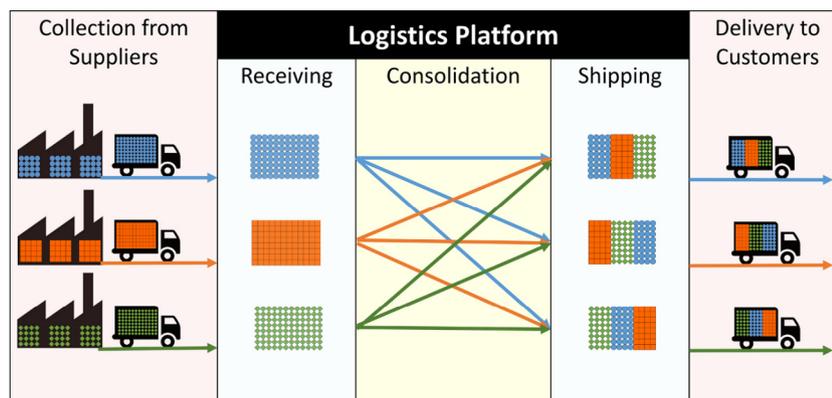


Fig. 1. Possible cross-docking combinations.

The main contributions of this paper to the cross-docking literature are: (1) modeling a novel multi-period cross-dock truck scheduling problem that considers different capacity constraints for the trucks, time windows for inbound and outbound trucks as well as a balanced workload for the trucks during the multiple periods of planning; (2) proposing a GA to solve the resulting models in real-life scales; (3) comparing the performance of the proposed GA with that of the exact method of branch and bound through several benchmark instances using statistical analysis.

The remainder of this paper is organized as follows. In Section 2, the main characteristics of the problem being analyzed are briefly described and the contribution of our model to the cross-docking literature highlighted. In Section 3, we introduce the two cross-dock truck scheduling models. In Section 4, we solve both models using an exact branch and bound method and a customized GA. In Section 5, we present and analyze our experimental results. Finally, in Section 6, we conclude and propose future research directions.

2. Model properties and contribution to the literature

The purpose of truck scheduling problems in cross-docking is to find the best sequence for inbound and outbound trucks in order to minimize the whole operation time. The literature analyzing truck-scheduling in cross-docking has developed formal frameworks similar to the one considered by Yu and Egbelu [35] in different directions [33,6]. In general, these research directions have been introduced in pairs, whose main combinations have been illustrated in Table 1. The column variables represent the main contributions introduced by the corresponding paper, which are complemented by those of the row variables. In particular:

- Lim et al. [18] considered constraints in transportation schedules and warehouse capacities together with supplier and customer time windows. Shakeri et al. [29] analyzed truck scheduling with a capacity-constrained internal workforce in a cross-dock with multiple doors. Together with truck-scheduling, these authors took internal cross-dock scheduling into account.
- The introduction of time windows when the cross-docking process and the trucks are subject to capacity constraints was initially considered by Li et al. [13]. The subsequent literature has recently focused on adding multiple docks to the time-windows-constrained setting [34].
- Lim et al. [16], Lim et al. [17] and Miao et al. [20] formalized the scheduling of inbound and outbound trucks in a setting with multiple dock doors. Moreover, Lim et al. [16] considered also different capacity constraints while time windows were added by Lim et al. [17] and Miao et al. [20].
- Among the recent developments of the literature, we must emphasize the formalization of cross-docking networks. This type of structure has been designed to account for multiple cross-docks, each one of which is usually in charge of handling individual shipments, while subject to a time window constraint [7]. The extension of this type of structure into distribution networks subject to the limited capacity of the cross-dock and the trucks defines one of the most recent directions of the literature [21].

- Finally, the current paper introduces multiple planning periods while considering different truck capacities and time windows, but lacks a network structure and does not consider the assignment of multiple doors, which would require the scheduling of internal cross-docking activities.

We consider a truck scheduling problem within a cross-dock consisting of a unique strip and a unique stack door. Both the inbound and outbound trucks have to be scheduled considering different time windows. We concentrate on the temporal dimension of the problem and its generalization to an environment with multiple time periods, while keeping the (multiple) door-assignment problem aside. The main objective of the optimization model consists of minimizing the operational time of the cross-docking process, i.e., the time required to transfer all the products from the inbound to the outbound trucks. The basic assumptions defining our truck scheduling problem are enumerated below:

- An exclusive mode of service is considered, i.e., one of the dock doors is exclusively assigned to inbound trucks and the other to outbound trucks.
- All products from the inbound trucks are unloaded in the arriving dock and transferred to the shipping dock, where they are loaded into outbound trucks.
- Different types of products are loaded and unloaded sequentially, i.e., the loading or unloading time of a truck is proportional to the different types of products involved in the process.
- Preemption is not allowed, i.e., all trucks have to be completely processed before leaving the dock.
- We allow for intermediate storage inside the cross-dock. That is, inbound trucks can unload their products before the corresponding outbound truck is available.
- We do not constrain the capacity of the storage area but impose a dynamic consistency requirement on the number of products being transferred in the cross-dock per time period.
- We do not consider internal operations such as sorting and labeling or the assignment of (limited) personnel and equipment to perform any required cross-docking operations.
- The truck changeover time as well as the time required to transfer products between doors is assumed to be fixed.

The cross-dock scheduling paradigm is generally classified in the categories of short-range and mid-term planning, although it has been mainly studied within the former one during the last decades. When considering temporary storage limitations, cross-dock scheduling is categorized as short-range planning. In this regard, the inclusion of multi-period properties brings the problem closer to mid-term planning. When the planning parameters of a cross-dock facility change through short- and mid-term periods, multi-period planning is required to make optimum decisions throughout a mid-term planning period.

The design of a cross-docking schedule through multiple planning periods requires defining the balance of loads for inbound and outbound trucks throughout the periods of time considered. Moreover, the corresponding structure must integrate time windows and variable truck capacities within the multi-period planning schedule. Therefore, the required modifications will be

Table 1
Truck-scheduling in cross-docking: research directions and literature review.

Variations with respect to Yu and Egbelu	Initial directions			New developments	
	Capacity constraints	Time windows	Multiple doors/docks	Cross-docking network	Multiple periods
Capacity constraints		Li et al. [13]	Lim et al. [16]	Mohtashami et al. [21]	X
Time windows	Lim et al. [18]		Miao et al. [20]		X
Multiple doors/docks	Shakeri et al. [29]	Van Belle et al. [34]		Chen et al. [7]	–

progressively implemented through two models. The first one extends the formal setting defined by Yu and Egbelu [35] in order to account for time windows and variable truck capacities, while the second model integrates the first one within a balanced multi-period planning schedule.

The reason for this progressive introduction of the modifications is twofold. First, it allows us to illustrate how the consistency conditions introduced to formalize the multi-period planning schedule interact dynamically with the time windows and capacity constraints imposed in the initial model. In this regard, the initial one-period model helps providing the intuition required. Second, we need a reference case against which to compare the computational requirements and performance of the extended dynamical framework when solving the model numerically. The next section describes the models proposed in detail.

3. Problem formulation

3.1. Model (1)–(19) features and details

Model (1)–(19) considers a one-period scenario where the trucks have due dates. They have to load or unload products and leave the dock before a specific due date. In this model, we consider time windows for both inbound and outbound trucks in the receiving and shipping docks, respectively. Therefore, each

Table 2
Model (1)–(19) parameters and variables.

Indices and sets	i	Indicator of inbound truck $i = 1, 2, 3, \dots, R$	
	j	Indicator of outbound truck $j = 1, 2, 3, \dots, S$	
	k	Indicator of product type $k = 1, 2, 3, \dots, N$	
	R	Set of inbound trucks	
	S	Set of outbound trucks	
	N	Set of product types	
Parameters	r_{ik}	Number of type k products that are unloaded from inbound truck i	
	s_{jk}	Number of type k products that are loaded in outbound truck j	
	$DDate_i$	Due date of inbound truck i	
	$RDate_j$	Due date of outbound truck j	
	Cap_i	Capacity of inbound truck i	
	Cap_j	Capacity of outbound truck j	
	DD	Required time for truck change	
	VV	Required time for the movement of products from the receiving dock to the shipping dock	
	M	A large positive value	
	Continuous and integer decision variables	x_{ijk}	Number of type k products that are transferred from inbound truck i to outbound truck j
		C_i	The entering time of inbound truck i into the receiving dock
		F_i	The leaving time of inbound truck i from the receiving dock
		d_j	The entering time of outbound truck j into the shipping dock
L_j		The leaving time of outbound truck j from the shipping dock	
T		The completion time of operations in the dock	
Binary variables		v_{ij}	If a product is transferred from inbound truck i to outbound truck j , this variable is equal to 1; otherwise, it is equal to 0
		p_{ij}	This variable determines the sequence of inbound trucks. If inbound truck i has priority over inbound truck j , this variable is equal to 1; otherwise, it is equal to 0
	q_{ij}	This variable determines the sequence of outbound trucks. If outbound truck i has priority over outbound truck j , this variable is equal to 1; otherwise, it is equal to 0	

inbound truck unloads the products before a due date and each outbound truck must leave the dock before another due date. Consequently, the truck due dates are taken as input parameters in the model. Truck capacities are also taken into consideration, with trucks not being allowed to transport more products than their limited capacities. Note that, the capacities of inbound and outbound trucks may be different as the inbound and outbound fleets are not homogenous and may contain different types of trucks and capacities. Table 2 presents the indices, sets, parameters, and decision variables used in Model (1)–(19).

The single period cross-dock scheduling problem defined in Model (1)–(19) has been introduced to provide a basis on which to build the multi-period cross-dock scheduling problem defined in Model (20)–(41). More precisely, Model (1)–(19) will be used to emphasize the main consistency requirements that must be implemented when defining its dynamic counterpart in Model (20)–(41). The latter framework encompasses several periods that describe how the variables defining its dynamic decision structure are forecasted and optimized in the first period of planning. The dynamic interactions that take place among the decision variables through the planning horizon are lacking in the single period problem defined in Model (1)–(19). Consequently, the results derived from optimizing the single period cross-dock scheduling problem per period of planning would differ from those obtained when optimizing a multi-period cross-dock scheduling problem. At the same time, the formal and numerical differences that arise when comparing both models provide the intuition required to analyze the dynamic framework constituting the basis of Model (20)–(41).

The following mixed integer mathematical programming is proposed as Model (1)–(19).

$$\text{Min } Z = T \quad (1)$$

$$T \geq L_j; \quad j = 1, 2, \dots, S; \quad (2)$$

$$\sum_{j=1}^S x_{ijk} = r_{ik}; \quad i = 1, 2, \dots, R; \quad k = 1, 2, \dots, N; \quad (3)$$

$$\sum_{i=1}^R x_{ijk} = s_{jk}; \quad j = 1, 2, \dots, S; \quad k = 1, 2, \dots, N; \quad (4)$$

$$x_{ijk} \leq M v_{ij}; \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \quad k = 1, 2, \dots, N; \quad (5)$$

$$F_i \geq C_i + \sum_{k=1}^N r_{ik}; \quad i = 1, 2, \dots, R; \quad (6)$$

$$C_j \geq F_i + DD - M(1 - p_{ij}); \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \quad i \neq j; \quad (7)$$

$$C_i \geq F_j + DD - M p_{ij}; \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \quad i \neq j; \quad (8)$$

$$p_{ii} = 0; \quad i = 1, 2, \dots, R; \quad (9)$$

$$q_{ij} = 0; \quad j = 1, 2, \dots, S; \quad (10)$$

$$L_j \geq d_j + \sum_{k=1}^N s_{jk}; \quad j = 1, 2, \dots, S; \quad (11)$$

$$C_j \geq F_i + DD - M(1 - p_{ij}); \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \quad i \neq j; \quad (12)$$

$$C_i \geq F_j + DD - M p_{ij}; \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \quad i \neq j; \quad (13)$$

$$L_j \geq C_i + VV + \sum_{k=1}^N x_{ijk} - M(1 - v_{ij}); \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \quad (14)$$

$$F_i \leq DDate_i; \quad i = 1, 2, \dots, R; \quad (15)$$

$$L_j \leq RDate_j; \quad j = 1, 2, \dots, S; \quad (16)$$

$$\sum_{k=1}^N r_{ik} \leq Cap_i; \quad i = 1, 2, \dots, R; \quad (17)$$

$$\sum_{k=1}^N s_{jk} \leq Cap_j; \quad j = 1, 2, \dots, S; \quad (18)$$

$$C_i, F_i, d_j, L_j, T \in \mathbb{R}^+; \quad x_{ijk} \in \mathbb{Z}^+; \quad v_{ij} \in \{0, 1\}; \quad p_{ij} \in \{0, 1\}; \quad q_{ij} \in \{0, 1\}. \quad (19)$$

The objective function (1) minimizes the maximum time required for the outbound trucks to leave the shipping dock. At the same time, this function minimizes the operational time of the cross-docking process. The set of constraints (2) is written for all the outbound trucks and guarantees that the operation time required by each outbound truck to leave the shipping dock is less than or equal to the operational time of the cross-docking process. The set of constraints (3), which is written for all inbound trucks and all types of products, guarantees that the number of type k products which are transferred from inbound truck i to all outbound trucks is equal to the number of type k products which are unloaded from truck i .

The set of constraints (4) requires the number of type k products which are transferred from all inbound trucks to outbound truck j to be equal to the number of type k products which are needed for truck j . The set of constraints (5) relates directly variables v_{ij} and x_{ijk} . It states that when a product of a given type is transferred from an inbound truck to an outbound truck, x_{ijk} can either be positive or equal to zero. Note that v_{ij} is a binary variable which equals one if a product is transferred from truck i to truck j . Otherwise, v_{ij} would be equal to zero imposing a direct constraint on the number of products transferred from truck i to truck j . We have included this relation in order to determine the leaving time of the outbound trucks.

The set of constraints (6) determines the leaving time of inbound truck i from the dock, which is greater than or equal to the time required for the truck to enter the dock plus the sum of the unloading times of all its products into the dock. The sets of constraints (7) and (8) are used to determine the entry sequence of inbound trucks into the receiving dock. In particular, the variable determining the order of precedence between inbound trucks i and j is p_{ij} . If inbound truck i has precedence over inbound truck j , then $p_{ij} = 1$, implying that the entrance of truck j into the dock takes place after truck i exits the dock. The same reasoning applies when it is inbound truck j the one that has precedence over inbound truck i and $p_{ij} = 0$. The next two constraints provide consistency to the truck schedule, with (9) guaranteeing that no receiving-truck enters the dock before itself and (10) guaranteeing that no outbound truck leaves the dock after itself.

The sets of constraints (11)–(13) determine the entry sequence of outbound trucks into the shipping dock. The set of constraints (11) calculates the time required for truck j to leave the dock, i.e., L_j . In this regard, the time required for an outbound truck to leave the shipping dock must account for the entrance time of the truck into the dock and the time required to load the truck with all its products. Similarly to the case with inbound trucks, the sets of constraints (12) and (13) determine the entry sequence of outbound trucks. However, in this case, it is the binary variable q_{ij} the one determining the order of precedence between outbound trucks i and j .

The set of constraints (14) reinforces the direct link defined in (5) between inbound and outbound trucks. These constraints require the exit time of outbound truck j from the shipping dock to be greater than or equal to the sum of

- the entrance time of inbound truck i into the receiving dock, C_i ;
- the time required to move products between docks, VV ;
- the time required to transfer the products from the inbound to the outbound truck, $\sum_{k=1}^N x_{ijk}$.

That is, this set of constraints provides dynamic consistency between the entrance and transfer processes taking place in the dock. It accounts for all inbound and outbound trucks together with all types of products. Note, in particular, that the last right hand side term determines whether or not the constraint is active. Thus, as was the case in (5), the binary variable v_{ij} will be equal to one if a product is transferred from truck i to truck j , obliging the outbound truck to wait for all the inbound trucks whose products it must ship. This means that the corresponding constraint is active. Otherwise, the value of v_{ij} will be equal to zero, relaxing the corresponding constraints, since $L_j \geq -M$, with $j = 1, 2, \dots, S$.

The sets of constraints (15) and (16) define the due dates for the inbound and outbound trucks, respectively. Note that, the inbound (outbound) trucks should unload (load) the products in the receiving (shipping) dock and leave the dock before the due date. The sets of constraints (17) and (18) have been introduced to ensure that the capacities of the inbound and outbound trucks are not exceeded. Finally, constraint (19) defines the type of decision variables considered in the model.

3.2. Model (20)–(41) features and details

Model (20)–(41) extends the formal setting of Model (1)–(19) and allows for scheduling to be defined over a multiple-period planning horizon. As already stated, standard cross-docking models only consider scheduling for a single period of planning. Our second model defines scheduling over several consecutive time periods, leading to a multi-period mixed integer mathematical model. The aim of this model is to find the best sequence for the inbound and outbound trucks so as to minimize the whole operation time over the multiple-periods considered.

Moreover, in Model (20)–(41), a balance among the products transferred in different time periods is imposed. In this way, the workload of the receiving and shipping docks is leveled throughout the multiple-periods composing the planning horizon. This constraint allows us to balance the transference of products in cross-docking and prevents products from being accumulated in the cross-dock. Therefore, the concept of temporary warehouse is implemented more efficiently relative to the existing models in the literature. Table 3 presents the indices, sets, parameters, and decision variables used in Model (20)–(41).

The following multi-period mixed integer mathematical programming is proposed as Model (20)–(41).

$$\text{Min } Z = \sum_{t=1}^T T_t \quad (20)$$

$$T_t \geq L_{jt}; \quad j = 1, \dots, S; \quad t = 1, \dots, T; \quad (21)$$

$$\sum_{j=1}^S x_{ijkt} = r_{ikt}; \quad i = 1, 2, \dots, R; \quad k = 1, 2, \dots, N; \quad t = 1, 2, \dots, T; \quad (22)$$

$$\sum_{i=1}^R x_{ijkt} = s_{jkt}; \quad j = 1, 2, \dots, S; \quad k = 1, 2, \dots, N; \quad t = 1, 2, \dots, T; \quad (23)$$

Table 3
Model (20)–(41) parameters and variables.

Indices and sets	i	Indicator of inbound truck $i = 1, 2, 3, \dots, R$	
	j	Indicator of outbound truck $j = 1, 2, 3, \dots, S$	
	k	Indicator of product type $k = 1, 2, 3, \dots, N$	
	t	Indicator of time period $t = 1, 2, 3, \dots, T$	
Parameters	R_t	Number of inbound trucks in period t	
	S_t	Number of outbound trucks in period t	
	r_{ikt}	Number of type k products that are unloaded from inbound truck i in period t	
	s_{jkt}	Number of type k products that are loaded in outbound truck j in period t	
	$DDate_{it}$	Due date of inbound truck i in period t	
	$RDate_{jt}$	Due date of outbound truck j in period t	
	Cap_{it}	Capacity of inbound truck i in period t	
	Cap_{jt}	Capacity of outbound truck j in period t	
	DD	Required time for truck change	
	VV	Required time for the movement of products from the receiving dock to shipping dock	
	M	A large positive value	
Continuous and integer decision variables	x_{ijkt}	Number of type k products that are transferred from inbound truck i to outbound truck j in period t	
	AVG	The average number of products transferred in the cross-dock through the whole planning horizon	
	C_{it}	The entering time of inbound truck i into the receiving dock in period t	
	F_{it}	The leaving time of inbound truck i from the receiving dock in period t	
	d_{jt}	The entering time of outbound truck j into the shipping dock in period t	
	L_{jt}	The leaving time of outbound truck j from the shipping dock in period t	
	T_t	The completion time of operations in the dock in period t	
	Z	The completion time of operations in the dock in all planning periods	
	Binary variables	v_{ijt}	If in time period t , a product is transferred from inbound truck i to outbound truck j , this variable is equal to 1; otherwise, it is equal to 0
		p_{ijt}	This variable determines the sequence of inbound trucks. If in time period t , inbound truck i has priority over inbound truck j , this variable is equal to 1; otherwise, it is equal to 0
q_{ijt}		This variable determines the sequence of outbound trucks. If in time period t , outbound truck i has priority over outbound truck j , this variable is equal to 1; otherwise, it is equal to 0	

$$L_{jt} \geq d_{jt} + \sum_{k=1}^N s_{jkt}; \quad j = 1, 2, \dots, S; \quad t = 1, 2, \dots, T; \quad (30)$$

$$d_{jt} \geq L_{it} + DD - M(1 - q_{ijt}); \quad i = 1, 2, \dots, R; \\ j = 1, 2, \dots, S; \quad i \neq j; \quad t = 1, 2, \dots, T; \quad (31)$$

$$d_{it} \geq L_{jt} + DD - Mq_{ijt}; \quad i = 1, 2, \dots, R; \\ j = 1, 2, \dots, S; \quad i \neq j; \quad t = 1, 2, \dots, T; \quad (32)$$

$$L_{jt} \geq C_{it} + VV + \sum_{k=1}^N x_{ijkt} - M(1 - v_{ijt}); \quad i = 1, 2, \dots, R; \\ j = 1, 2, \dots, S; \quad t = 1, 2, \dots, T; \quad (33)$$

$$F_{it} \leq DDate_{it}; \quad i = 1, 2, \dots, R; \quad t = 1, 2, \dots, T; \quad (34)$$

$$L_{jt} \leq RDate_{jt}; \quad j = 1, 2, \dots, S; \quad t = 1, 2, \dots, T; \quad (35)$$

$$AVG = \sum_{t=1}^T \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijkt} / T; \quad (36)$$

$$\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijkt} - \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijkt-1} \leq AVG; \quad t = 2, \dots, T; \quad (37)$$

$$\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijkt-1} - \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijkt} \leq AVG; \quad t = 2, \dots, T; \quad (38)$$

$$\sum_{k=1}^N r_{ikt} \leq Cap_{it}; \quad i = 1, 2, \dots, R; \quad t = 1, 2, \dots, T; \quad (39)$$

$$\sum_{k=1}^N s_{jkt} \leq Cap_{jt}; \quad j = 1, 2, \dots, S; \quad t = 1, 2, \dots, T; \quad (40)$$

$$C_{it}, F_{it}, d_{jt}, L_{jt}, T_t, Z \in \mathbb{R}^+; \quad x_{ijkt} \in \mathbb{Z}^+; \quad v_{ijt} \in \{0, 1\}; \\ p_{ijt} \in \{0, 1\}; \quad q_{ijt} \in \{0, 1\}. \quad (41)$$

Note that the formal structure of this model is very similar to that of Model (1)–(19). The main difference between both models is given by the time index t and the new time-period decision variable introduced, T_t . These modifications give place to three additional sets of constraints, (36)–(38), relative to the first model. The remaining sets of constraints are time-indexed versions of those composing Model (1)–(19) that must now be integrated within a consistent dynamical framework. We describe all of them explicitly below for completeness.

The objective function (20) minimizes the sum of the maximum amounts of time required by the trucks to complete their schedules during all the periods of planning. In other words, it minimizes the total (multi-period) completion time of operations in the dock. The set of constraints (21) guarantees that the amount of time required to transfer products in period t is higher than the operation time required by each outbound truck to leave the shipping dock. The set of constraints (22) guarantees that the number of type k products which are transferred from inbound truck i to all outbound trucks in period t is equal to the number of type k products which are unloaded from truck i .

The set of constraints (23) states that the number of type k products transferred from all the inbound trucks to outbound truck j in period t must be equal to the number of type k products which are loaded in truck j . Similarly to (5) in Model (1)–(19), the set of constraints (24) relates x_{ijkt} and the binary variable v_{ijt} each time period t of the planning horizon. The set of constraints (25)

$$x_{ijkt} \leq M v_{ijt}; \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \quad k = 1, 2, \dots, N; \\ t = 1, 2, \dots, T; \quad (24)$$

$$F_{it} \geq C_{it} + \sum_{k=1}^N r_{ikt}; \quad i = 1, 2, \dots, R; \quad t = 1, 2, \dots, T; \quad (25)$$

$$C_{jt} \geq F_{it} + DD - M(1 - p_{ijt}); \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \\ i \neq j; \quad t = 1, 2, \dots, T; \quad (26)$$

$$C_{it} \geq F_{jt} + DD - Mp_{ijt}; \quad i = 1, 2, \dots, R; \\ j = 1, 2, \dots, S; \quad i \neq j; \quad t = 1, 2, \dots, T \quad (27)$$

$$p_{iit} = 0; \quad i = 1, 2, \dots, R; \quad t = 1, 2, \dots, T; \quad (28)$$

$$q_{ijt} = 0; \quad j = 1, 2, \dots, S; \quad t = 1, 2, \dots, T; \quad (29)$$

determines the leaving time of inbound truck i from the dock in period t , which must be greater than or equal to the time required for the truck to enter the dock plus the sum of the unloading times of all its products into the dock (in period t).

Similarly to (7) and (8) in Model (1)–(19), the sets of constraints (26) and (27) use the binary variable p_{ijt} to determine the entry sequence of inbound trucks into the receiving dock for each time period t . The next two constraints provide consistency to the truck schedule in each time period t , with (28) guaranteeing that no inbound truck enters the dock before itself and (29) guaranteeing that no outbound truck leaves the dock after itself.

The sets of constraints (30)–(32) determine the entry sequence of outbound trucks into the shipping dock for each time period t : (30) calculates the time required for truck j to leave the dock, while (31) and (32) define the entry sequence of outbound trucks using the binary variable q_{ij} to determine the order of precedence between outbound trucks i and j .

The set of constraints (33) strengthens the link defined in (24) between inbound and outbound trucks for each time period t . These constraints require the exit time of outbound truck j from the dock in period t , L_{jt} , to be greater than or equal to the sum of

- the entrance time of inbound truck i into the receiving dock, C_{it} ;
- the time required to move products between docks, VV ;
- the time required to transfer the products from the inbound to the outbound truck, $\sum_{k=1}^N x_{ijk t}$.

Similarly to (14), this set of constraints provides dynamic consistency between the entrance and transfer processes taking place in the dock per time period of planning. It accounts for all inbound and outbound trucks together with all types of products and time periods. Indeed, the last right hand side term determines whether or not the constraint is active in the planning period t . Thus, as was the case in (24), the binary variable v_{ijt} will be equal to one if a product is transferred from truck i to truck j in period t , forcing the outbound truck to wait for all the inbound trucks whose products it must load. This means that the corresponding constraint is active and given by

$$L_{jt} \geq C_{it} + VV + \sum_{k=1}^N x_{ijk t}; \quad i = 1, 2, \dots, R; \quad j = 1, 2, \dots, S; \\ t = 1, 2, \dots, T.$$

Otherwise, the value of v_{ij} will be equal to zero, relaxing the corresponding constraints, since $L_{jt} \geq -M$, with $j = 1, 2, \dots, S$, and $t = 1, \dots, T$.

The sets of constraints (34) and (35) indicate the due dates imposed on the inbound and outbound trucks in period t , respectively. As already stated, the formalization of a multi-period setting with variable capacities and time windows imposed on the inbound and outbound trucks constitutes the novel feature of the model proposed in this study. As a result, and given their respective capacities,

- the inbound trucks must unload the products in the receiving dock and leave the dock, and
- the outbound trucks must load the products from the shipping dock and leave the dock before the corresponding due dates assigned each time period.

Moreover, since Model (20)–(41) has been designed to account for a multiple-period planning horizon, the schedules generated should maintain a balance between the products transferred in the different planning periods. Constraint (36) computes the average number of products transferred in the cross-dock through the

whole planning horizon. Note, in particular, that this equation provides the reference value used to balance the workload assigned throughout the different periods composing the planning horizon. We have introduced the set of constraints (37) and (38) in order to preserve the balance between the products transferred through the different planning periods. This set of constraints guarantees that the difference between the products transferred in the cross-dock in two consecutive periods is lower than or equal to the average number of products transferred in the dock during all the periods of planning.

In order to illustrate the implications derived from imposing the dynamic consistency conditions (36)–(38) on the number of products being transferred between docks each time period, consider the patterns of products transferred per time period, i.e., $\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk t}$, described in Table 4. These transfer patterns are illustrated in Fig. 2.

The first and fourth patterns both violate the consistency requirements imposed by Eqs. (36)–(38). In order to see why this is the case, note that the difference between the number of products transferred in the third and the second period equals 14 for the first pattern and 12 for the fourth. Therefore, the first pattern violates Eq. (37) while the fourth pattern violates Eq. (38). In this regard, the dynamic consistency constraints imposed on cross-dock transfers and defined by Eqs. (36)–(38) can be divided in two specific differentiated effects.

Eq. (37) can be interpreted as an upper limit condition on the variability of cross-dock transfers, while Eq. (38) represents the corresponding lower limit one. Both conditions provide substantial control over the pattern of cross-dock transfers, which allows us to prevent large increments, decrements or fluctuations in the amount of products being transferred through the different time periods. Note, in particular, how any negative value obtained from either $\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk t} - \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk t-1}$

Table 4
Different patterns of products transferred per time period and their AVG values.

$\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk t}$	Time period				AVG
	t = 1	t = 2	t = 3	t = 4	
First pattern	5	10	24	13	13
Second pattern	5	10	21	12	12
Third pattern	10	16	7	11	11
Fourth pattern	10	16	4	10	10

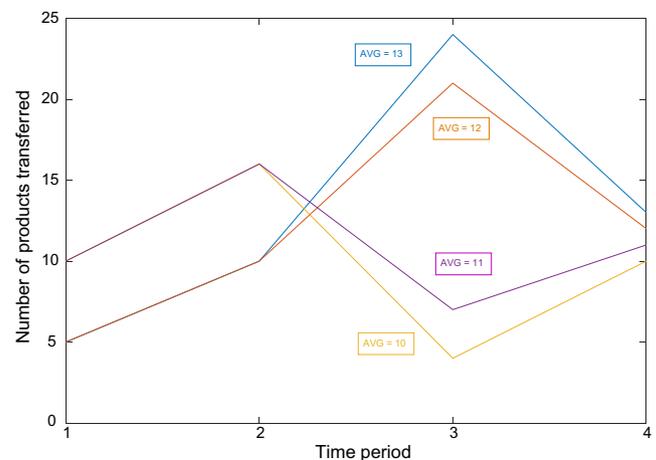


Fig. 2. Different patterns of products transferred per time period and their AVG values.

$\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t-1} - \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t}$, depending on the dynamic behavior of $\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t}$, is accounted for by its symmetric counterpart.

Imposing only one of these conditions, either (37) or (38), requires introducing an additional non-negativity constraint to prevent potential violations of the corresponding condition from being undetected. Consider, for example, the upper limit setting. In this case, the values of the variable $x_{ijk,t}$ should be sorted in increasing order through time when defining the evolution of the product transfer pattern. Eq. (37) would be used to prevent a sudden increase in this pattern, though it would allow for a progressive one. A similar reasoning applies to the lower limit setting and Eq. (38). However, in this case, the values of $x_{ijk,t}$ must be sorted in decreasing order.

Eqs. (36)–(38) can be aggregated as follows:

$$\left| \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t} - \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t-1} \right| \leq \left(\sum_{t=1}^T \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t} / T \right);$$

$$t = 2, \dots, T,$$

with the right-hand side given by the reference value computed in Eq. (36). This set of equations summarizes how AVG is used to balance the workload of the dock assigned throughout the different periods of the planning horizon.

Finally, note that additional conditions could be imposed to keep the transfer process of products consistent throughout all the time periods considered, preventing substantial increasing or decreasing trends relative to a given reference value. In this case, imposing a constraint based on transfer differences relative to those of the initial period would give place to upper and lower limits on the corresponding pattern. The upper limit on the pattern would be given by

$$\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t} - \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,1} \leq AVG; \quad t = 2, \dots, T;$$

while the lower limit would be defined by

$$\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,1} - \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t} \leq AVG; \quad t = 2, \dots, T.$$

That is, these constraints would limit the spread of the $x_{ijk,t}$ variable through time relative to its initial value, delivering a smooth distribution of transfers throughout all the time periods.

As already stated, in Model (20)–(41), the capacity of the trucks is allowed to change through the different planning periods. In this regard, the amount of products that are loaded cannot exceed the capacity of the outbound trucks in any of the planning periods. Also, the amount of products that are unloaded each period from the inbound trucks cannot exceed their capacity. The sets of constraints (39) and (40) have been introduced to ensure that the capacities of the inbound and outbound trucks are not exceeded in any of the planning periods. Finally, constraint (41) defines the type of decision variables considered in Model (20)–(41).

4. Solution procedure

Both proposed models are solved using an exact branch and bound method through LINGO software and a customized GA coded in MATLAB software for several benchmark instances. Based on the number of products and that of inbound and outbound trucks, several instances of the models are generated and solved in small, medium, and large scales. Then, the results of both methods on all instances are compared using statistical analysis.

We describe below the main stages composing the GA applied to solve the cross-docking models proposed in this paper.

4.1. Chromosome design

Two genetic algorithmic structures have been designed in order to solve Model (1)–(19) and Model (20)–(41) numerically. Generally, a solution of a truck scheduling problem should consist of two differentiated parts presenting a sequence of inbound and outbound trucks, respectively.

Fig. 3 illustrates how inbound and outbound trucks are integrated within the set of total trucks considered in the one-period cross-docking Model (1)–(19). This figure describes the chromosome designed for Model (1)–(19), which consists of R cells assigned to the inbound trucks and S cells for outbound trucks. The values of the cells composing the first part of the chromosome, which determine the sequence of inbound trucks, are given by non-repeated random numbers between 1 and R. On the other hand, the values of the cells composing the second part of the chromosome, which determine the sequence of outbound trucks, are given by non-repeated random numbers between 1 and S.

Therefore, a solution of Model (1)–(19) would consist of a row matrix with R+S cells and be based on an initial population of chromosomes (sequences of trucks) that defines a given entry sequence in the receiving and shipping docks determined by the set of available trucks.

The chromosome designed to define the solutions of Model (20)–(41) is given in the form of a table with two main dimensions. This table extends the chromosome designed for Model (1)–(19) so as to handle multiple planning periods. Each row of the chromosome designed for Model (20)–(41) corresponds to the schedule of inbound and outbound trucks for a single planning period. Therefore, the number of rows composing the chromosome is determined by the number of planning periods considered. Fig. 4 summarizes the chromosome designed for Model (20)–(41).

4.2. Selection procedure

In this stage of the algorithm, a number of chromosomes is selected from the initial population in order to start the crossover stage. In the proposed GA algorithm, a standard roulette wheel method is used for selection.

4.3. Fitness function

The fitness function consists of minimizing the total completion time of operations in the dock and is written based on the objective functions of Model (1)–(19) and Model (20)–(41).

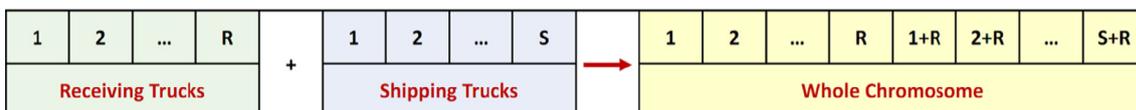


Fig. 3. Model (1)–(19) chromosome.

4.4. Crossover

The crossover scheme implemented is a variant of the standard two-point crossover operator and is described in Fig. 5. Given the assumed probability of intersection (crossover), we determine semi-randomly the genes that will be affected from each parent chromosome: two genes (trucks) from the R string and another two from the S string will be copied from the opposite parent, while the remaining genes remain unchanged for each corresponding child.

4.5. Mutation

The mutation scheme implemented modifies the docking sequence of a subset of semi-randomly selected trucks. The muta-

1	...	R+S
...
t	...	R+S
...
T	...	R+S

Fig. 4. Model (20)–(41) chromosome.

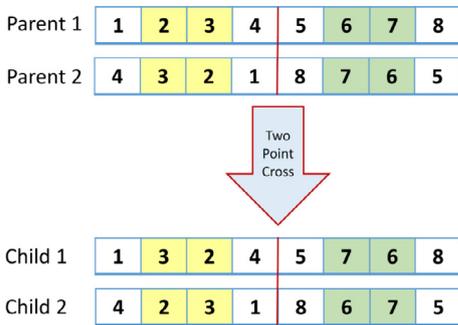


Fig. 5. Proposed cross-over.

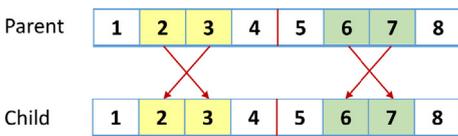


Fig. 6. Mutation operator.

tion process consist of selecting a subset of two trucks from the R string of the chromosome and swapping them. At the same time, another subset of two trucks from the S string is selected and their order swapped. Fig. 6 provides a schematic view of the proposed mutation operator using a numerical example.

4.6. Constraint handling

We have used a basic penalty strategy in order to enforce the constraints of Model (1)–(19) and Model (20)–(41) in the proposed GA method. That is, during the generation of the random initial solutions as well as through the crossover and mutation schemes, the feasibility of each chromosome is checked considering the constraints of Model (1)–(19) and Model (20)–(41). If an infeasibility arises in the solution, then a penalty is associated to the value of the fitness functions of the corresponding chromosome. In this way, infeasible chromosomes have a lower probability of being part of the next generation of chromosomes.

4.7. GA flowchart

Fig. 7 presents the flowchart of the GA proposed to solve the single and multi-period cross-dock scheduling problems with multiple products, due dates, and temporary warehouse introduced in the current paper.

5. Experimental results

In order to investigate the performance of the proposed models and solution procedures, an extensive numerical testing procedure including three class of instances (i.e., small, medium, and large problems) has been performed for both models. All test problems have been solved using both solution methods (i.e., branch and bound, and the GA). Then, the results are presented and compared for each model (i.e., Model (1)–(19) and Model (20)–(41)) and each solution method on these instances. We have generated 30 small size problem instances, 30 medium size problem instances, and 30 large size problem instances. Totally, 90 test problems have been generated for both models using the setting presented in Table 5.

Note that the parameters determining the size of the test problems have been selected using uniform distribution functions in order to generate sufficient variability. That is, sensitivity analysis has been performed by modifying the number of trucks, products and product types. The notation $U[a, b]$ in Table 5 implies that the associated parameters follow a uniform probability distribution function in the interval a to b . The performance of both solution methods has been assessed using statistical analysis on all 90 test problems.

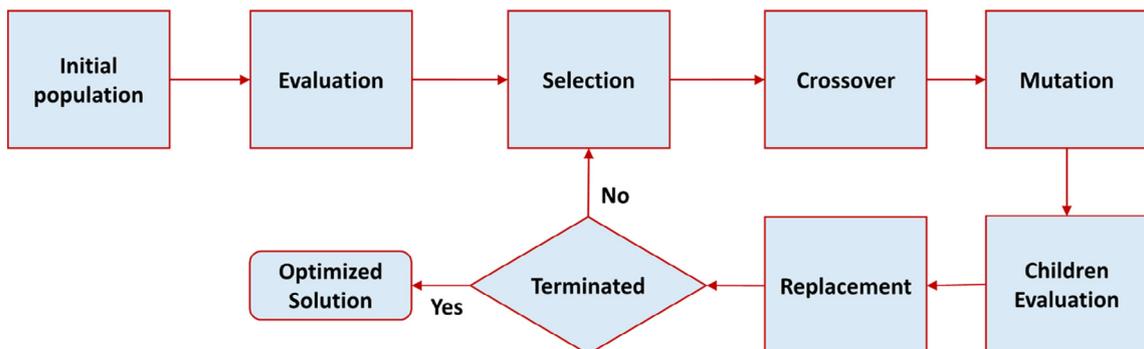


Fig. 7. Flowchart of the proposed GA.

Table 5
Dimension and size of test problems.

Problem instance	Inbound truck	Outbound truck	Types of products	Number of products	Number of time periods
Small	U[4, 8]	U[5, 8]	U[4, 9]	U[100, 500]	U[2, 5]
Medium	U[10, 20]	U[10, 20]	U[10, 20]	U[500, 2000]	U[6, 10]
Large	U[40, 50]	U[40, 50]	U[40, 50]	U[5000, 10000]	U[11, 20]

Table 6
GA parameter range and final parameter values.

Parameter	Range of changes	Initial value	Final value
Population size	30, 50, 80	10	30
Termination condition	50, 100, 150, 200	50	100
Probability of combination	0.8, 0.85, 0.9	0.8	0.9
Mutation rate	0.01, 0.03	0.02	0.02

Several settings have been tested in order to achieve the most suitable tuning of the parameters of the GA. Table 6 summarizes the ranges of the parameters used to implement the GA together with the final values employed. The full set of parameter combinations was tested and the best values were determined based on a suitable tradeoff between the CPU time required and the quality of the solutions generated.

5.1. Results of the small-size instances

It will be assumed in both models that inbound trucks enter the receiving dock, unload the products and leave the dock within the

time window assigned. Products are either transferred to a temporary warehouse, which is located in the shipping dock, or are directly transferred to the outbound trucks. Through this and the next section we will present the results obtained for Model (1)–(19) and Model (20)–(41) after implementing both solution procedures (i.e., branch and bound using LINGO and the GA using MATLAB) to solve all 90 test problems (i.e., small, medium, and large). Table 7 and Fig. 8 present the structure and results of the 30 small size test problems.

The CPU time and value of the objective functions for both models in all 30 small instances using both solution procedures are presented in Table 7. Note how, even in small-size instances, the average CPU times for both LINGO and the GA experience a noticeable increase when shifting from Model (1)–(19) to Model (20)–(41).

Consider now the averages of the error percentages described in Table 7. The error percentage values have been computed dividing the difference between the objective values obtained using the GA and the branch and bound method by the objective value obtained using the branch and bound method. That is, this variable describes the gap error arising between the GA and the branch and bound solutions. The average of the error percentages is equal to 0.006

Table 7
Structure and results of small-size test problems.

Problem structure						Solution of models										
problem size	Instance no.	In bound trucks	Out bound trucks	Product type	Product no.	Model (1)–(19)					Model (20)–(41)					
						CPU time		Objective function			Time Period	CPU time		Objective function		
						LINGO	GA	LINGO	GA	Error%		LINGO	GA	LINGO	GA	Error%
Small	1	5	7	5	180	7.00	6.98	882	885	0.0038	3.00	13.59	9.78	11340	11383	0.0038
	2	6	5	4	170	6.00	3.97	893	900	0.0081	2.00	3.93	11.26	6120	6165	0.0074
	3	8	8	7	406	12.80	11.12	2598	2618	0.0076	2.00	0.82	21.32	31181	31555	0.0120
	4	8	7	5	357	11.20	6.97	2799	2824	0.0091	3.00	6.62	13.09	35986	36188	0.0056
	5	7	6	9	203	8.40	10.74	663	667	0.0055	5.00	9.65	39.95	25578	25764	0.0073
	6	6	6	6	328	7.20	7.19	1378	1378	0.0006	4.00	17.88	16.34	28339	28806	0.0165
	7	7	6	9	247	8.40	10.79	807	808	0.0019	4.00	29.34	11.73	24898	24943	0.0018
	8	7	8	7	154	11.20	11.12	862	868	0.0062	4.00	40.08	4.75	20698	20978	0.0136
	9	5	7	5	385	7.00	6.97	1887	1891	0.0025	5.00	9.47	17.01	40425	40818	0.0097
	10	7	6	7	190	8.40	8.40	798	804	0.0080	3.00	14.18	0.92	14364	14415	0.0035
	11	5	6	7	401	6.00	8.32	1203	1205	0.0018	2.00	9.82	2.37	14436	14498	0.0043
	12	6	6	6	295	7.20	7.17	1239	1250	0.0090	2.00	0.52	0.79	12744	12792	0.0037
	13	5	5	6	418	5.00	5.99	1219	1222	0.0021	5.00	3.47	7.60	31350	31869	0.0165
	14	4	5	6	465	4.00	5.97	1085	1095	0.0094	4.00	1.44	4.91	22320	22337	0.0008
	15	7	5	6	336	7.00	5.96	1372	1381	0.0066	2.00	2.32	7.90	14112	14128	0.0011
	16	7	5	7	228	7.00	6.96	798	804	0.0071	5.00	23.93	5.17	23940	24063	0.0051
	17	5	5	6	177	5.00	5.95	516	521	0.0087	4.00	15.48	14.74	10620	10726	0.0100
	18	6	5	4	354	6.00	3.96	1859	1865	0.0037	2.00	4.36	11.46	12744	12939	0.0153
	19	7	6	7	168	8.40	8.32	706	712	0.0089	4.00	30.23	28.80	16934	17211	0.0163
	20	6	5	8	206	6.00	7.95	541	545	0.0074	3.00	17.23	12.04	11124	11254	0.0117
	21	5	8	9	193	8.00	14.34	600	602	0.0028	4.00	12.08	22.02	18528	18784	0.0138
	22	7	8	8	386	11.20	12.69	1891	1910	0.0097	3.00	17.47	1.09	38909	39256	0.0089
	23	8	8	9	191	12.80	14.26	951	958	0.0073	2.00	9.77	18.62	14669	14852	0.0125
	24	4	8	7	436	6.40	11.15	1395	1400	0.0036	4.00	23.66	15.06	33485	33825	0.0102
	25	4	6	7	292	4.80	8.38	701	704	0.0042	2.00	5.23	2.25	8410	8486	0.0090
	26	5	5	8	226	5.00	7.99	494	499	0.0088	2.00	6.61	0.54	6780	6790	0.0015
	27	8	7	6	172	11.20	8.40	1124	1128	0.0036	5.00	6.19	5.40	28896	29234	0.0117
	28	5	6	4	485	6.00	4.76	2546	2569	0.0090	4.00	17.68	10.94	34920	34965	0.0013
	29	8	6	7	461	9.60	8.35	2213	2220	0.0033	2.00	9.72	11.30	26554	26652	0.0037
	30	6	5	7	177	6.00	6.99	531	534	0.0060	4.00	20.66	14.35	12744	12788	0.0034
Average						7.67	8.27	1218.34	1225.59	0.0059	-	12.78	11.45	21104.86	21282.04	0.0081

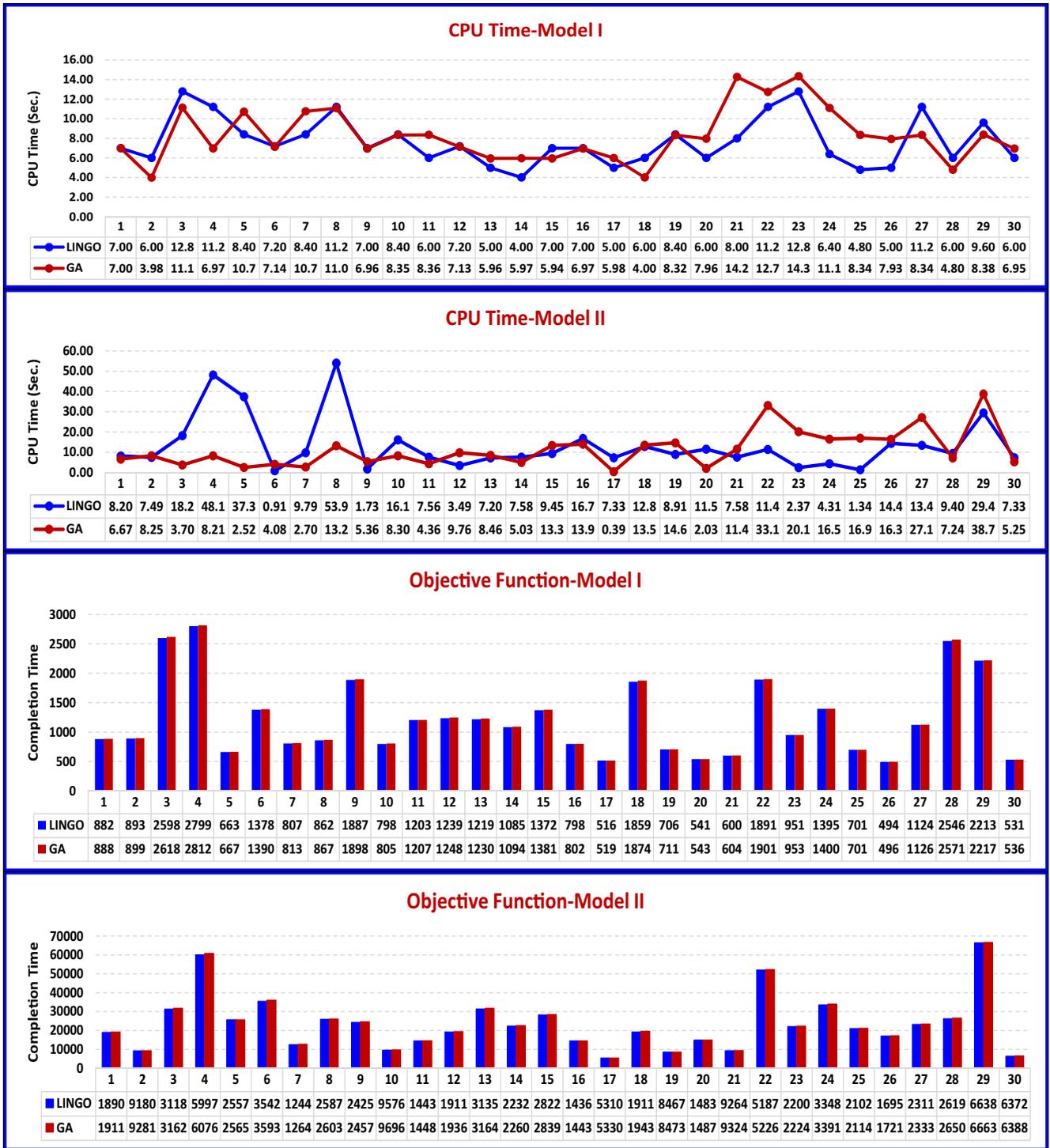


Fig. 8. CPU time and objective function values based on the GA and LINGO for Model (1)–(19) and Model (20)–(41) in small-size instances.

and 0.008 for Model (1)–(19) and Model (20)–(41), respectively. This means that the proposed GA is reliable based on the results obtained from the small size instances and can achieve the optimal solutions within a suitable approximation in all 30 instances.

5.2. Results of the medium-size and large-size instances

The structures and results of the medium-size and large-size instances are presented in Tables 8, and 9, respectively.

The CPU time and objective function values obtained for both models in all 30 medium-size instances using both solution

procedures are presented in Table 8. Note the substantial increase experienced by the average CPU times for both LINGO and the GA when shifting from Model (1)–(19) to Model (20)–(41). Note also how LINGO requires a higher average CPU time to reach a solution in both models. The average of the error percentages is equal to 0.004 and 0.009 for Model (1)–(19) and Model (20)–(41), respectively. This means that the proposed GA method can be assumed to be reliable for medium-size instances and is able to achieve the near optimal solutions in all 30 medium-size instances.

When considering large-size test problems, LINGO was unable to converge toward the optimum solutions even after long runs.

Table 8
Structure and results of medium-size test problems.

Problem structure						Solution of models										
Problem size	Instance no.	In bound trucks	Out bound trucks	Product type	Product no.	Model (1)–(19)					Model (20)–(41)					
						CPU time		Objective function			Time period	CPU time		Objective function		
						LINGO	GA	LINGO	GA	Error%		LINGO	GA	LINGO	GA	Error%
Medium	31	20	15	19	970	60.00	56.85	10721	10795	0.0069	10.00	283.17	305.01	1746000	1752014	0.0034
	32	19	18	14	1627	68.40	50.08	27822	27866	0.0016	9.00	550.41	248.62	3004744	3044589	0.0133
	33	16	15	12	1284	48.00	35.89	17976	18122	0.0081	9.00	299.35	351.61	1664064	1667242	0.0019
	34	15	19	14	1221	57.00	52.75	17399	17477	0.0044	9.00	409.35	221.90	1879119	1908999	0.0159
	35	15	17	16	981	51.00	53.88	10944	11015	0.0065	10.00	300.43	78.99	1500930	1504428	0.0023
	36	12	10	19	1975	24.00	37.83	8732	8756	0.0028	10.00	204.49	182.56	1422000	1422985	0.0007
	37	13	11	11	1251	28.60	24.10	11384	11400	0.0014	6.00	109.36	130.44	644015	650557	0.0102
	38	13	17	10	537	44.20	33.85	8307	8360	0.0064	9.00	288.79	325.69	640856	648787	0.0124
	39	14	15	16	889	42.00	47.66	8168	8212	0.0054	6.00	34.37	24.06	672084	682922	0.0161
	40	13	17	11	1155	44.20	37.20	16244	16289	0.0028	9.00	322.86	375.28	1378377	1401225	0.0166
	41	20	15	17	1732	60.00	50.86	21395	21498	0.0048	9.00	270.69	91.06	2805840	2847964	0.0150
	42	16	12	19	643	38.40	45.43	4548	4555	0.0015	10.00	283.21	272.99	740736	741859	0.0015
	43	19	13	14	1859	49.40	36.26	22959	23075	0.0051	7.00	285.54	129.62	1928527	1929796	0.0007
	44	15	11	10	984	33.00	21.94	11365	11436	0.0063	6.00	80.89	14.25	584496	588898	0.0075
	45	14	10	16	992	28.00	31.98	6076	6090	0.0023	6.00	18.58	9.93	499968	506671	0.0134
	46	18	11	14	674	39.60	30.72	6673	6706	0.0049	10.00	154.94	132.56	800712	806276	0.0069
	47	16	15	13	1468	48.00	38.97	18971	19001	0.0016	10.00	14.02	245.54	2113920	2135793	0.0103
	48	17	15	15	1120	51.00	45.00	13328	13344	0.0012	10.00	366.23	447.85	1713600	1735224	0.0126
	49	11	19	17	525	41.80	63.98	4518	4526	0.0018	9.00	339.29	122.42	592515	599767	0.0122
	50	13	13	16	952	33.80	41.42	7039	7049	0.0015	9.00	16.58	35.41	868795	872577	0.0044
	51	20	10	19	1039	40.00	37.67	7656	7674	0.0024	7.00	108.92	48.55	872760	887051	0.0164
	52	11	16	14	1739	35.20	44.76	15303	15430	0.0083	7.00	185.66	152.24	1285469	1300766	0.0119
	53	10	19	13	1711	38.00	49.01	17505	17555	0.0028	6.00	21.92	146.64	1170324	1184535	0.0121
	54	13	11	20	1726	28.60	43.58	8639	8706	0.0078	10.00	282.88	107.33	1480908	1483287	0.0016
	55	19	16	10	1948	60.80	31.80	41453	41866	0.0100	7.00	64.60	263.17	2487206	2502087	0.0060
	56	17	13	20	1963	44.20	51.49	15184	15307	0.0081	7.00	90.12	230.29	1822057	1834121	0.0066
	57	20	14	19	1517	56.00	52.88	15649	15725	0.0049	10.00	493.72	376.85	2548560	2569440	0.0082
	58	13	20	14	1647	52.00	55.98	21411	21528	0.0055	7.00	223.34	324.96	1798524	1824179	0.0143
	59	16	17	14	1370	54.40	47.26	18632	18772	0.0075	10.00	437.32	211.12	2235840	2253717	0.0080
	60	17	10	16	635	34.00	31.91	4723	4748	0.0052	8.00	254.58	174.51	518160	523648	0.0106
Average						44.45	42.77	14024.11	14096.17	0.0047	8.40	226.52	192.72	1447370.16	1460380.06	0.0091

As a result, the LINGO solution procedure implemented in Model (1)–(19) was terminated after 3600 s and the best known solutions were reported. The termination condition was set to 7200 s in Model (20)–(41). Given the findings reported in Table 9, it can be concluded that the proposed GA method outperformed the LINGO-based one in both models. The objective function values and CPU times achieved by the GA are much better than the corresponding ones obtained with the LINGO-based method.

The CPU time and objective function values obtained using both methods to solve both models within the medium and large size settings are presented in Figs. 9 and 10, respectively.

Based on the findings obtained in the small-size and medium-size instances, it could be assumed that the results of the GA for large-size instances are reliable, although additional analyses must be performed.

5.3. Statistical analysis

Although the results presented in Tables 7–9 illustrate the efficacy of the proposed GA method in all 90 test problems, an extensive statistical analysis is also performed to test whether there is a meaningful statistical difference between the performance of the exact branch and bound method implemented in LINGO and the proposed GA implemented in MATLAB. To this end, we start by running the Kolmogorov-Smirnov normality test to check whether the objective function values obtained in both models using both solution procedures follow a normal probability density function. In statistical analysis, the hypothesis of equal means is based on the probability density function of the population from which the sample is derived. If it is confirmed that the sample is normally

distributed, then it can be inferred that the population also follows a normal distribution. In this regard, the Kolmogorov-Smirnov is one of the most well-known and widely implemented normality tests.

MINITAB software was used to perform the Kolmogorov-Smirnov test on the results obtained. Recall that the null hypothesis of this test validates the distributional normality of the observations, while the alternative hypothesis rejects it. The results of the test for both models and both solution methods are summarized in Fig. 11.

Whenever the p-values presented in Fig. 11 are lower than 0.05, we can conclude that there is not enough evidence to accept the null hypothesis and the observations/results obtained cannot be assumed to be normally distributed. In contrast, when the p-values are higher than 0.05 there is not enough evidence to reject the null hypothesis and the observations/results obtained can be assumed to be normally distributed.

The next step depends on the normality of the results obtained. If the results are normally distributed, a normal parametric test (one-way ANOVA) will be run for each model to verify the equality of the means of the results obtained using the GA and LINGO. On the other hand, if the results are not normally distributed, a non-parametric test (Kruskal-Wallis) must be performed to verify the equality of the medians of the results obtained using both solution methods.

The results obtained when running the corresponding one-way ANOVA and Kruskal-Wallis tests are presented in Table 10.

The Kruskal-Wallis test was performed for both models in the small-size instances setting. In Model (1)–(19), the test statistic (H) has a p-value of 0.465, indicating that the null hypothesis

Table 9
Structure and results of large-size test problems.

Problem structure						Solution of models										
Problem size	Instance no.	In bound trucks	Out bound trucks	Product type	Product no.	Model (1)–(19)					Model (20)–(41)					
						CPU time		Objective function			Time period	CPU time		Objective function		
						LINGO	GA	LINGO	GA	Error%		LINGO	GA	LINGO	GA	Error%
Large	61	46	40	50	5648	3600.00	396.81	187062	145775	-0.2207	16.00	7200.00	129.01	16627712	3276970	-0.8029
	62	42	43	43	6188	3600.00	367.33	233906	183568	-0.2152	14.00	7200.00	2258.13	15645739	2135327	-0.8635
	63	41	42	48	7749	3600.00	400.26	250196	196197	-0.2158	16.00	7200.00	3556.96	21350045	3715530	-0.8260
	64	43	48	41	8105	3600.00	393.56	367216	286966	-0.2185	19.00	7200.00	5311.94	31784568	4701554	-0.8521
	65	43	47	47	5170	3600.00	440.46	200079	156196	-0.2193	17.00	7200.00	2661.96	17762569	2924871	-0.8353
	66	43	44	49	7617	3600.00	428.08	264699	206459	-0.2200	13.00	7200.00	4250.37	18734773	3493900	-0.8135
	67	44	50	50	7358	3600.00	496.48	291377	228460	-0.2159	15.00	7200.00	4499.12	24281400	2680747	-0.8896
	68	46	46	41	7880	3600.00	374.30	366016	286377	-0.2176	11.00	7200.00	2339.88	18341488	2788228	-0.8480
	69	47	49	50	7002	3600.00	485.92	290261	226912	-0.2182	20.00	7200.00	6891.69	32251212	4318549	-0.8661
	70	43	50	44	5238	3600.00	437.16	230353	180152	-0.2179	15.00	7200.00	2811.71	16892550	2189997	-0.8704
	71	43	43	45	8923	3600.00	385.82	329973	256963	-0.2213	14.00	7200.00	3903.11	23098078	2390344	-0.8965
	72	47	44	46	5654	3600.00	401.97	228766	179276	-0.2163	11.00	7200.00	3450.37	12861719	1991287	-0.8452
	73	50	50	50	8126	3600.00	499.18	365670	284775	-0.2212	17.00	7200.00	4226.08	34535500	3565937	-0.8967
	74	43	47	50	6785	3600.00	468.26	246825	192821	-0.2188	19.00	7200.00	5109.06	26053722	3203018	-0.8771
	75	42	40	45	8232	3600.00	357.22	276595	216284	-0.2180	19.00	7200.00	3851.88	26276544	4313575	-0.8358
	76	43	50	45	5459	3600.00	448.98	234737	182978	-0.2205	20.00	7200.00	5757.16	23473700	2958511	-0.8740
	77	46	42	49	5852	3600.00	409.13	207662	162722	-0.2164	11.00	7200.00	437.41	12436670	2260203	-0.8183
	78	42	48	40	6540	3600.00	380.73	296654	232333	-0.2168	11.00	7200.00	1911.40	14503104	1968496	-0.8643
	79	50	45	46	6834	3600.00	413.29	300845	235753	-0.2164	17.00	7200.00	3288.79	26140050	3566165	-0.8636
	80	48	41	40	6922	3600.00	325.34	306506	239635	-0.2182	15.00	7200.00	3125.34	20433744	3177262	-0.8445
	81	48	48	40	5510	3600.00	382.90	285638	223671	-0.2169	14.00	7200.00	3056.97	17773056	2069536	-0.8836
	82	50	41	48	5749	3600.00	391.96	220977	171946	-0.2219	14.00	7200.00	289.12	16499630	1892116	-0.8853
	83	41	47	50	8922	3600.00	466.52	309468	242071	-0.2178	20.00	7200.00	3386.73	34385388	5381355	-0.8435
	84	43	48	49	6006	3600.00	470.37	227689	177407	-0.2208	14.00	7200.00	1162.33	17354938	2411194	-0.8611
	85	46	46	43	5029	3600.00	393.64	222726	173519	-0.2209	13.00	7200.00	222.27	13833773	2086559	-0.8492
	86	49	49	49	5923	3600.00	477.23	261204	203938	-0.2192	14.00	7200.00	6522.68	19909572	3273660	-0.8356
	87	45	42	46	9141	3600.00	386.38	338018	265028	-0.2159	18.00	7200.00	3694.31	31097682	4286177	-0.8622
	88	50	49	41	8580	3600.00	401.56	461437	359920	-0.2200	11.00	7200.00	1436.15	23123100	2489944	-0.8923
	89	47	40	45	6060	3600.00	357.24	227856	178783	-0.2154	16.00	7200.00	427.48	18228480	2610665	-0.8568
	90	41	44	46	7491	3600.00	403.99	264400	205654	-0.2222	18.00	7200.00	1845.93	24324775	3285011	-0.8650
Average						3600.00	414.74	276493.70	216084.53	-0.2185	15.40	7200.00	3060.51	21667176.04	3046889.64	-0.8573

(i.e., the equality of the medians of the objective function values obtained by both methods in all 30 instances) can be accepted at significance levels 0.05. This means that the performance of the proposed GA method is assumed to be statistically identical to that of the exact branch and bound method. The same conclusion is true for Model (20)–(41), since the test statistic (H) has a p-value of 0.465. Thus, the performance of the proposed GA is also competitive for all 30 instances of Model (20)–(41).

The test performed on the medium-size instances setting depends on the model being considered. The ANOVA test is the one required for Model (1)–(19). In the corresponding ANOVA table, the p-value of 0.988 implies that there is sufficient evidence to consider the means of the objective function values obtained by both methods equal at significance levels 0.05. The Kruskal-Wallis test was performed for Model (20)–(41). The test statistic (H) has a p-value of 0.466, indicating that the equality of the medians of the objective function values obtained by both methods can be accepted at significance levels 0.05. Thus, the identical performance of both solution methods for small and medium-size instances is statistically verified.

In the large-size instances setting, the situation is somehow different since the LINGO solver was not able to converge toward an optimum solution. The ANOVA test has been performed on both models, whose test statistics have both an associated p-value equal to zero. This means that the hypothesis of equality of means of the objective function values achieved by both solution procedures in Model (1)–(19) and Model (20)–(41) is rejected. According to the average values presented in Table 9, this result implies that the proposed GA model outperforms the exact branch and bound method in all large-size instances.

As for the small-size and medium-size instances, the statistical analysis has shown that the proposed GA can achieve the optimum solutions obtained by the branch and bound method. Consequently, the results obtained by the GA in large-size instances can be assumed to be of suitable quality. All in all, we have verified that the proposed GA is competitive with respect to the exact branch and bound method while it also decreases the required CPU time substantially. Moreover, the proposed GA has provided suitable solutions for large-size instances whereas the exact branch and bound method had to be terminated due to non-convergence problems after a long run time.

6. Conclusion and future research directions

In this paper, we have proposed a novel problem in the area of truck scheduling in cross-docking. The problem has been formalized using two mixed integer mathematical programming models. The aim of these models was to determine the best sequence of inbound and outbound trucks in order to minimize the whole operational time and/or to maximize (optimize) the shipping process in the cross-dock. The main contributions of these models to the literature of truck scheduling in cross-docking, which makes them particularly well-posed to handle real life problems in this area, are the following:

- The truck scheduling problem in cross-docking has been developed to account for multiple time periods of planning.
- The type of trucks composing the fleet as well as their capacities have been allowed to differ during the planning horizon.

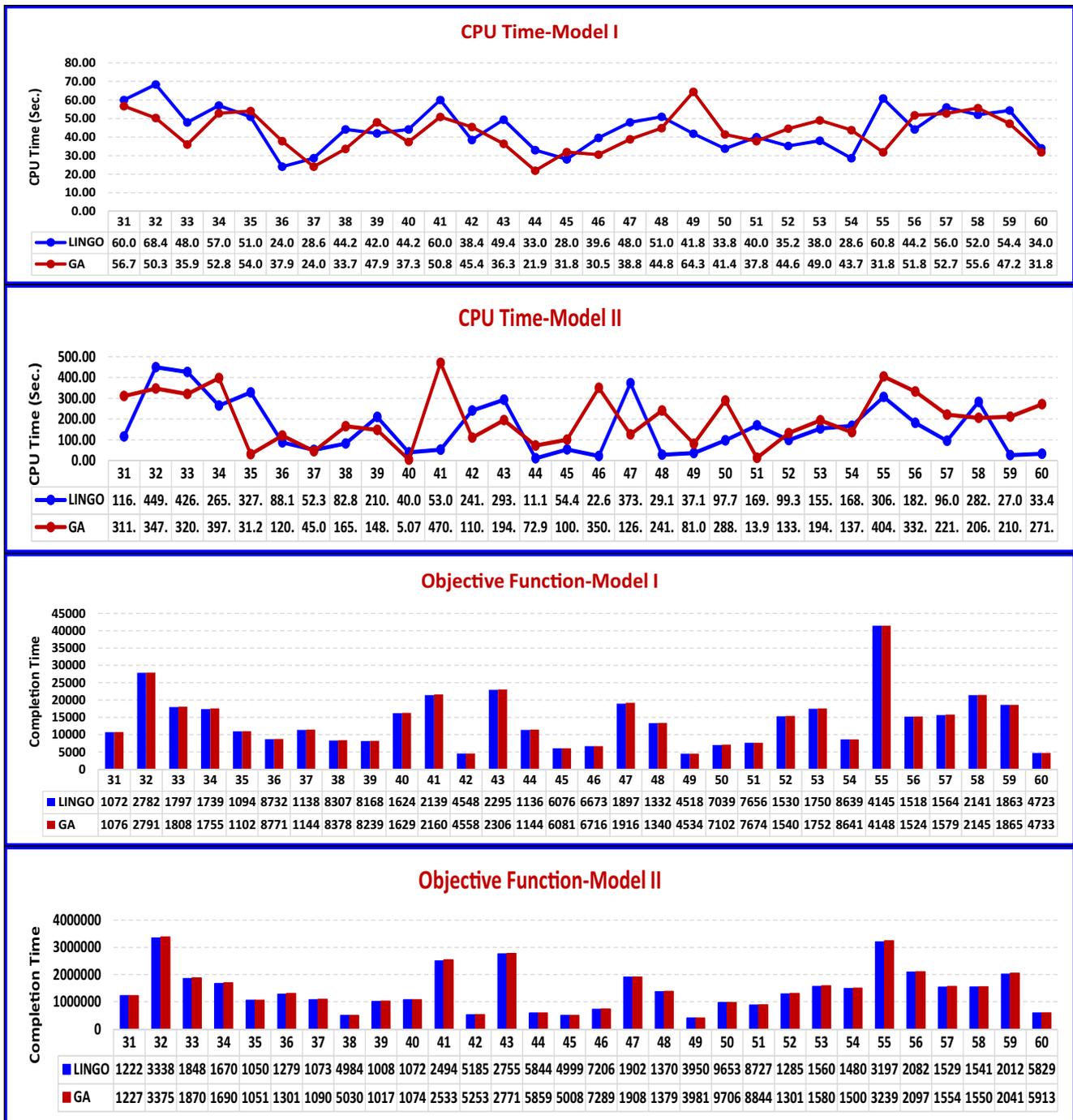


Fig. 9. CPU time and objective function values based on the GA and LINGO for Model (1)–(19) and Model (20)–(41) in medium-size instances.

- The work load of trucks and cross-docking activities have been kept in balance when designing the multi-period schedules.
- Time windows have been considered for both inbound and outbound trucks when generating the multi-period schedules.

A comprehensive experimental analysis has been conducted in order to assess the performance of the proposed models and solution procedures. Moreover, several experimental tests were performed to achieve the most suitable tuning of the proposed GA. Both models have been solved using an exact branch and bound method and the proposed customized GA through 90 benchmark instances distributed in three different size classes. The normality behavior of the CPU time and the objective function values

obtained after applying each solution method to both models was also analyzed. Finally, the performances of both solution procedures on both models were compared using suitable statistical tests.

Our numerical results indicate that the exact branch and bound method is accurate for small-size instances while the proposed GA followed it with a negligible gap error on the 30 test instances that were run. In medium-size instances, the GA has a gap error of less than 1 percent from the optimum solutions found by the exact branch and bound method while the computational times were competitive. However, the GA method is quicker for large-size instances where the branch and bound method was unable to con-

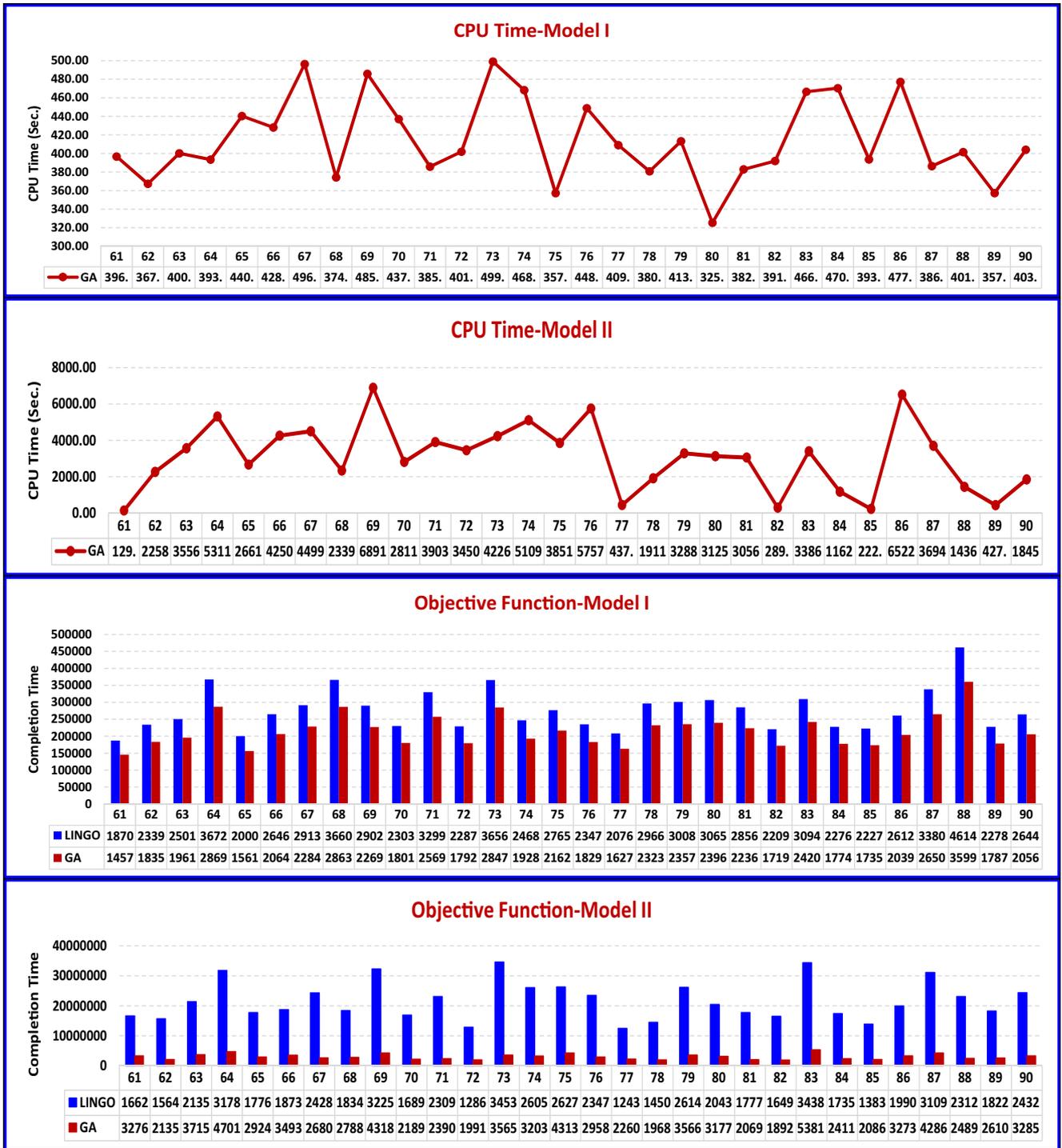


Fig. 10. CPU time and objective function values based on the GA and LINGO for Model (1)–(19) and Model (20)–(41) in large-size instances.

verge toward an optimal solution after a long CPU time. Moreover, after performing several runs of the small and medium-size instances, statistical analysis showed that there is no evidence to reject the hypothesis of equality of the means/medians of the objective function values provided by the exact branch and bound method coded in LINGO and the proposed GA coded in MATLAB. Thus, we concluded that the GA method is reliable and can be used as an applied solution procedure for large size problems in real life.

Research in this area is still in its early stages. Therefore, there are plenty of potential improvements that can be investigated in future research. Some of the areas that could be the subject of future analysis are listed as follows:

- In the models of this study, the distribution system or warehouse had only one shipping and one receiving dock. The number of receiving and shipping docks should be increased while accounting for different receiving and shipping strategies as well as for the probability of facility failure; refer, for example, to Liao et al. [14].

Intuitively, the model can be extended to consider a larger number of receiving and shipping docks by introducing binary variables that account for the dock to which a given truck is assigned each time period. This selection and distribution process has to be implemented for both inbound and outbound trucks.

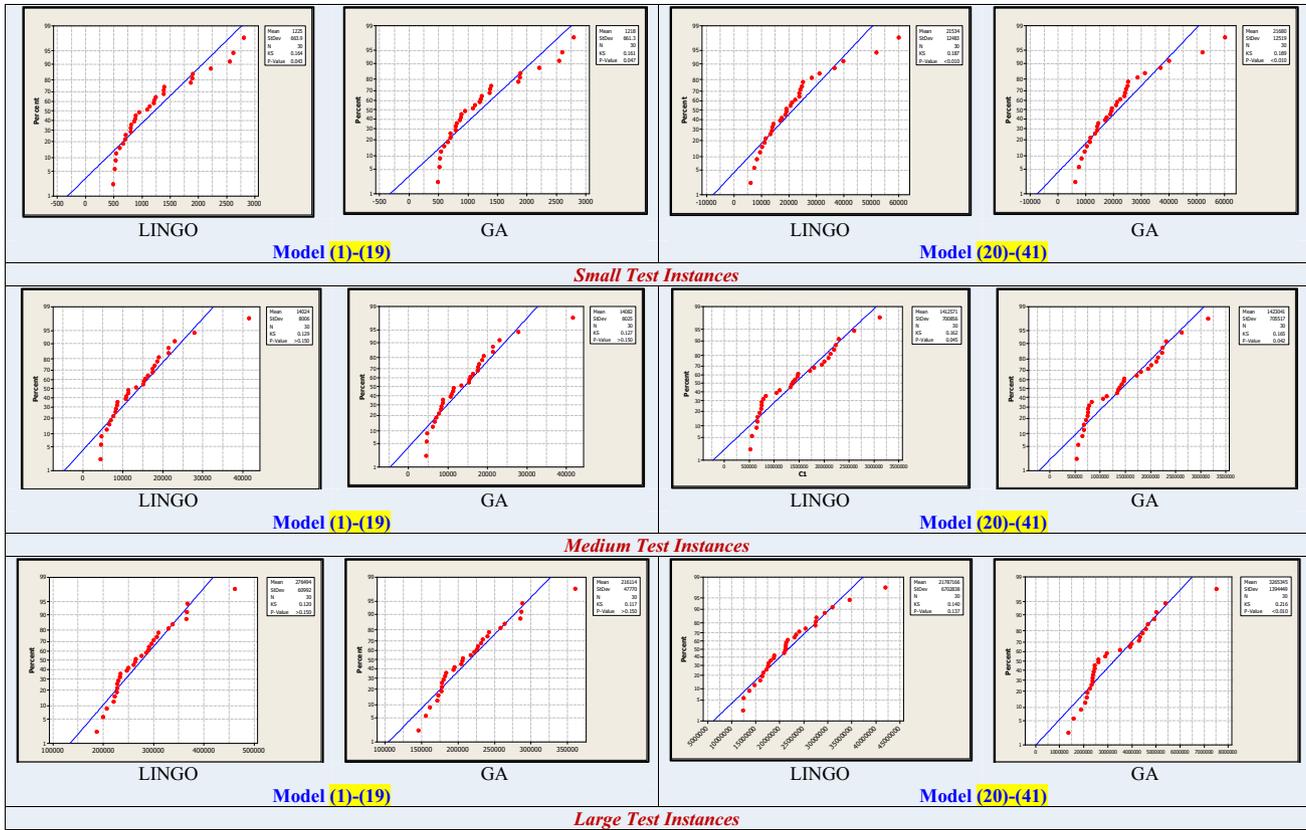


Fig. 11. Kolmogorov-Smirnov normality test on the objective function values.

Table 10
Kruskal-Wallis and ANOVA test results for Model (1)–(19) and Model (20)–(41).

<i>Small size problem</i>					
Kruskal-Wallis test results: Model (1)–(19)					
	H-Statistic	df		P-value	
Results	28.99	29		0.465	
Kruskal-Wallis test results: Model (20)–(41)					
	H-Statistic	df		P-value	
Results	28.99	29		0.466	
<i>Medium size problem</i>					
ANOVA test results: Model (1)–(19)					
Source	Sum of Squares	df	Mean Square	F	P-Value
Between groups	49709	1	14.400	0.012	0.988
Within groups	3726711214	58	1211.719		
Total	3726760923	59			
Kruskal-Wallis test results: Model (20)–(41)					
	H-Statistic	df		P-value	
Between groups	29.00	29		0.465	
<i>Large size instances</i>					
ANOVA test results: Model (1)–(19)					
Source	Sum of Squares	df	Mean Square	F	P-Value
Between groups	54686286760	1	54686286760	18.22	0.00
Within groups	1.74056E+11	58	3000957626		
Total	2.28742E+11	59			
ANOVA test results: Model (20)–(41)					
Source	Sum of Squares	df	Mean Square	F	P-Value
Between groups	5.14587E+15	1	5.14587E+15	219.57	0.00
Within groups	1.35930E+15	58	2.34363E+13		
Total	6.50517E+15	59			

For example, in the case of inbound trucks, such a variable could be denoted by y_{ilt} . This variable would be equal to one if in time period t , inbound truck i is assigned to receiving dock l , with $l = 1, 2, \dots, L$, representing the number of receiving docks available. Otherwise, y_{ilt} would be equal to zero. Additionally, we should constraint this variable so that $\sum_{l=1}^L y_{ilt} \leq 1, t = 1, 2, \dots, T$. This way, we could guarantee that each inbound truck is assigned to only one receiving dock per time period through the whole cross-docking process.

Note that we would have to adapt the shipping docks in a similar way. The binary variable w_{jnt} , such that $\sum_n w_{jnt} \leq 1$, with $t = 1, 2, \dots, T$ and $n = 1, 2, \dots, N$, representing the number of shipping docks available, could be used for this purpose.

Finally, among the subsequent modifications required to adapt the rest of the model, we would have to redefine the product transfer binary variable v_{ijt} so as to account for the receiving and shipping docks to which the inbound and outbound trucks are assigned, respectively. That is, we should redefine v_{ijnt} as follows: if in time period t , a product is transferred from inbound truck i in receiving dock l to outbound truck j in shipping dock n , this variable would be equal to one. Otherwise, it would be equal to zero. The rest of the equations composing the extended model should be adapted accordingly.

- The models proposed in this study have focused on the temporal dimension of the problem and sought to minimize the whole scheduling time of the cross-docking process. The management of the costs associated with the different cross-docking operations constitutes another important subject of analysis [1]. By adding the costs of transportation and maintenance, the proposed models can be transformed into multi-objective problems.
- The models proposed in this study have been designed to account for a multi-period setting. On the other hand, all cross-dock transfers have been assumed to be completed each period. That is, the inbound and outbound trucks did not experience any tardiness (delay with respect to the time window defined). Potential delays among inbound and outbound trucks could be considered and handled in the subsequent periods of planning, and such an extension introduced through a penalty term in a given cost objective function.

For completeness, we describe intuitively how to modify Model (20)–(41) as well as its dynamic consistency constraints so as to adapt for potential delays among inbound and outbound trucks.

First, note that a simple and direct way to introduce delays in Model (20)–(41) would be to modify Eqs. (34) and (35) as follows

$$F_{it} \leq DDate_{it} + \varepsilon_t;$$

$$L_{jt} \leq RDate_{jt} + \varepsilon_t,$$

for all the inbound ($i = 1, 2, \dots, R$) and outbound ($j = 1, 2, \dots, S$) trucks during a given time period $t \in \{1, 2, \dots, T\}$. Note that the ε_t term assumes implicitly that all the trucks are affected equally, though a particular subset of them could also be considered – defining variables such as ε_{it} or ε_{jt} –, which would complicate the corresponding analysis.

An alternative approach, which could be combined with the previous one, focusing on the dynamic consistency conditions would require modifying Eqs. (37) and (38) by adding an error term. In this case, something similar to confidence delivery intervals could be defined for the proposed model as follows

$$\sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijkt} - \sum_{k=1}^N \sum_{j=1}^S \sum_{i=1}^R x_{ijk,t-1} \leq AVG + \varepsilon_{ijt}; \quad t \in \{2, \dots, T\},$$

and similarly for Eq. (38). Note that this approach considers the effect that the x_{ijkt} variable has on the total time assigned to the cross dock through F_{it} and L_{jt} . Note also that the term ε_{ijt} refers to a delay taking place between two groups of trucks during a specific time period and applies only to a subset of them, i.e. $t \in \{2, \dots, T\}$.

Then, the corresponding penalty terms could be added to the objective function

$$\text{Min } Z = \sum_{t=1}^T T_t + \sum_{t \in \{2, \dots, T\}} \varepsilon_{ijt}$$

while fixing the i and j truck subindexes. As in the first approach, a particular subset of trucks could be selected, which would complicate the description considerably. At the same time, note that ε_{ijt} could be subtracted from the dynamic consistency conditions during any of the other time periods so as to prevent any penalty in the objective function. This type of dynamic trade-off could be considered in different potential scenarios arising from the current model.

- Finally, in the proposed models, it has been assumed that all the products were shipped to their destination within each corresponding time period. In future research, it could be assumed that products may be stored in the dock and transferred to the outbound trucks in any of the subsequent periods, an extension that would require accounting for the dynamic behavior of the cross-docking process. Similarly to the previous case, a penalty should be defined in the objective function to account for the cost of delay.

Acknowledgement

The authors would like to thank the anonymous reviewers and the editor for their insightful comments and suggestions.

References

- [1] D. Agustina, C.K.M. Lee, R. Piplani, Vehicle scheduling and routing at a cross docking center for food supply chains, *Int. J. Prod. Econ.* 152 (2014) 29–41.
- [2] J.J. Bartholdi, K.R. Gue, The best shape for a crossdock, *Transp. Sci.* 38 (2) (2004) 235–244.
- [3] A.R. Boloori Arabani, S.M.T. Fatemi Ghomi, M. Zandieh, A multi-criteria cross-docking scheduling with just-in-time approach, *Int. J. Adv. Manuf. Technol.* 49 (2010) 741–756.
- [4] A. Boloori Arabani, M. Zandieh, M.T. Fatemi Ghomi, Meta-heuristics implementation for scheduling of trucks in a cross-docking, *Expert Syst. Appl.* 38 (2011) 1964–1979.
- [5] N. Boysen, M. Fliedner, Cross dock scheduling: classification, literature review and research agenda, *Omega* 38 (2010) 413–422.
- [6] P. Buijs, I.F. Vis, H.J. Carlo, Synchronization in cross-docking networks: a research classification and framework, *Eur. J. Oper. Res.* 239 (2014) 593–608.
- [7] P. Chen, Y. Guo, A. Lim, B. Rodrigues, Multiple crossdocks with inventory and time windows, *Comput. Oper. Res.* 33 (1) (2006) 43–63.
- [8] F. Chen, C.-Y. Lee, Minimizing the makespan in a two machine crossdocking flowshop problem, *Eur. J. Operational Res.* 193 (1) (2007) 59–72.
- [9] R. Dondo, C. Méndez, J. Cerdá, The multi-echelon vehicle routing problem with cross docking in supply chain management, *Comput. Chem. Eng.* 35 (2011) 3002–3024.
- [10] K.R. Gue, Cross-docking: Just-in-time for Distribution, Teaching Notes, Graduate School of Business and Public Policy, Naval Postgraduate School, Monterey, CA, 2001, 93943.
- [11] J. Katz, Meeting at the crossdock, *Industry Week* 255 (5) (2006) 50.
- [12] Y.H. Lee, J.W. Jung, K.M. Lee, Vehicle routing scheduling for cross-docking in the supply chain, *Comput. Ind. Eng.* 51 (2006) 247–256.
- [13] Y. Li, A. Lim, B. Rodrigues, Crossdocking: JIT scheduling with time windows, *J. Oper. Res. Soc.* 55 (12) (2004) 1342–1351.
- [14] T.W. Liao, P.J. Egbelu, P.C. Chang, Simultaneous dock assignment and sequencing of inbound trucks under a fixed outbound truck schedule in multi-door cross docking operations, *Int. J. Prod. Econ.* 141 (2013) 212–229.
- [15] C.J. Liao, Y. Lin, S.C. Shih, Vehicle routing with cross-docking in the supply chain, *Expert Syst. Appl.* 37 (2010) 6868–6873.
- [16] A. Lim, H. Ma, Z. Miao, Truck dock assignment problem with time windows and capacity constraint in transshipment network through crossdocks, in: M. Gavrilova, O. Gervasi, V. Kumar, C.J.K. Tan, D. Taniar, A. Laganá, Y. Mun, H. Choo (Eds.), *Computational Science and its Applications - ICCSA 2006, Lecture Notes in Computer Science*, vol. 3982, 2006, pp. 688–697.

- [17] A. Lim, H. Ma, Z. Miao, Truck dock assignment problem with operational time constraint within crossdocks, in: M. Ali, R. Dapoigny (Eds.), *Advances in Applied Artificial Intelligence, Lecture Notes in Artificial Intelligence*, vol. 4031, 2006, pp. 262–271.
- [18] A. Lim, Z. Miao, B. Rodrigues, Z. Xu, Transshipment through crossdocks with inventory and time windows, *Naval Res. Logistics* 52 (8) (2005) 724–733.
- [19] G.M. Magableh, M.D. Rossetti, S. Mason, Modeling and analysis of a generic cross-docking facility, in: *Proceedings of the 37th Conference on Winter Simulation*, 2005, pp. 1613–1620.
- [20] Z. Miao, A. Lim, H. Ma, Truck dock assignment problem with operational time constraint within crossdocks, *Eur. J. Oper. Res.* 192 (1) (2009) 105–115.
- [21] A. Mohtashami, M. Tavana, F.J. Santos-Arteaga, A. Fallahian-Najafabadi, A novel multi-objective meta-heuristic model for solving cross-docking scheduling problems, *Appl. Soft Comput.* 31 (2015) 30–47.
- [22] S.M. Mousavi, R. Tavakkoli-Moghaddam, A hybrid simulated annealing algorithm for location and routing scheduling problems with cross-docking in the supply chain, *J. Manuf. Syst.* 32 (2013) 335–347.
- [23] S.M. Mousavi, R. Tavakkoli-Moghaddam, F. Jolai, A possibilistic programming approach for the location problem of multiple cross-docks and vehicle routing, *Eng. Optimization* 45 (2013) 1223–1249.
- [24] S.M. Mousavi, R. Tavakkoli-Moghaddam, B. Vahdani, H. Hashemi, Location of cross-docking centers and vehicle routing scheduling under uncertainty: a fuzzy possibilistic-stochastic programming model, *Appl. Math. Model.* 38 (2014) 2249–2264.
- [25] M. Napolitano, Making the move to crossdocking: a practical guide to planning, designing, and implementing a crossdock operation, Warehousing Education and Research Council (WERC), 2000.
- [26] M. Rohrer, Simulation and crossdocking, in: *Proceedings of the 27th Conference on Winter Simulation*, 1995, pp. 846–849.
- [27] A. Ross, V. Jayaraman, An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design, *Comput. Ind. Eng.* 55 (2008) 64–79.
- [28] F.A. Santos, G.R. Mateus, A. Salles da Cunha, A branch-and-price algorithm for a vehicle routing problem with cross-docking, *Electron. Notes Discrete Math.* 37 (2011) 249–254.
- [29] M. Shakeri, M.Y.H. Low, S.J. Turner, E.W. Lee, A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock, *Comput. Oper. Res.* 39 (11) (2012) 2564–2577.
- [30] R. Soltani, S. Sadjadi, Scheduling trucks incross-docking systems: a robust meta-heuristics approach, *Transp. Res. Part E: Logistics Transp. Rev.* 46 (5) (2010) 650–666.
- [31] K. Song, F. Chen, Scheduling crossdocking logistics optimization problem with multiple inbound vehicles and one outbound vehicle, in: *Proceedings of the 2007 IEEE International Conference on Automation and Logistics*, 2007, pp. 3089–3094.
- [32] B. Vahdani, M. Zandiyeh, Scheduling trucks incross-docking systems: robust meta-heuristics, *Comput. Ind. Eng.* 58 (1) (2010) 12–24.
- [33] J. Van Belle, P. Valckenaers, D. Cattrysse, Cross-docking: state of the art, *Omega* 40 (2012) 827–846.
- [34] J. Van Belle, P. Valckenaers, G. Vanden Berghe, D. Cattrysse, A tabu search approach to the truck scheduling problem with multiple docks and time windows, *Comput. Ind. Eng.* 66 (4) (2013) 818–826.
- [35] W. Yu, P.J. Egbelu, Scheduling of inbound and outbound trucks in cross docking, *Eur. J. Oper. Res.* 184 (2007) 377–396.