Computers & Industrial Engineering 139 (2020) 106134

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Truck scheduling in a multi-door cross-docking center with partial unloading – Reinforcement learning-based simulated annealing approaches

Amin Shahmardan, Mohsen S. Sajadieh*

Department of Industrial Engineering & Management Systems, Amirkabir University of Technology (Tehran Polytechnic), Tehran, Iran

ARTICLE INFO

Keywords:
Logistics
Cross docking
Truck scheduling
Simulated annealing
Reinforcement learning

ABSTRACT

In this paper, a truck scheduling problem at a cross-docking center is investigated where inbound trucks are also used as outbound. Moreover, inbound trucks do not need to unload and reload the demand of allocated destination, i.e. they can be partially unloaded. The problem is modeled as a mixed integer program to find the optimal dock-door and destination assignments as well as the scheduling of trucks to minimize makespan. Due to model complexity, a hybrid heuristic-simulated annealing is developed. A number of generic and tailor-made neighborhood search structures are also developed to efficiently search solution space. Moreover, some reinforcement learning methods are applied to intellectually learn more suitable neighborhood search structures in different situations. Finally, the numerical study shows that partial unloading of compound trucks has a crucial impact on makespan reduction.

1. Introduction

Cross docking is a logistics strategy that reduces total supply chain costs through exploiting the economies of scale in transportation through demand consolidation, and also reducing the amount of inventory held in distribution centers (Shakeri, Low, Turner, & Lee, 2012). The advantage of cross docking compared to traditional warehouses is shorter storage time. Although specifying an exact boundary is difficult, some papers (Li, Lim, & Rodrigues, 2004; Vahdani & Zandieh, 2010) define 24 h as the maximum acceptable storage time in these centers. Moreover, cross docking reduces or eliminates two of four major fundamental functions (receiving, storage, order picking, and shipping) of traditional warehouses including storage and order picking (Li et al., 2004). It is estimated that implementation of cross-docking centers instead of traditional warehouses may decrease operational cost up to 70% (Vahdani & Zandieh, 2010).

Dock-door assignment and truck-scheduling problems are two important operational decisions in cross docking. Some papers consider a simple cross-docking center with one receiving and one shipping dock door (Wooyeon & Egbelu, 2008). However, the majority of papers in the literature consider truck-scheduling problem in a multi dock-door cross-docking center (for example, Joo & Kim, 2013; Shakeri et al., 2012; Rahmanzadeh Tootkaleh, Fatemi Ghomi, & Sheikh Sajadieh, 2016; Van Belle, Valckenaers, Berghe, & Cattrysse, 2013). In addition, some papers study truck scheduling in a resource-constrained cross-docking center (Chmielewski, Naujoks, Janas, & Clausen, 2009; Hermel, Hasheminia, Adler, & Fry, 2016; Shakeri et al., 2012), while the remaining consider models without any resource limitation (see Boloori Arabani, Fatemi Ghomi, & Zandieh, 2010; Boloori Arabani, Fatemi Ghomi, & Zandieh, 2011; Boloori Arabani, Zandieh, & Fatemi Ghomi, 2011; Boloori Arabani, Zandieh, & Fatemi Ghomi, 2012; Bodnar, de Koster, & Azadeh, 2015; Joo & Kim, 2013; Konur & Golias, 2013; Van Belle et al., 2013; Vahdani & Zandieh, 2010; Rahmanzadeh Tootkaleh et al., 2016).

A number of assumptions have been studied in the literature. Some papers consider the situation in which trucks should leave the cross-docking center in a pre-defined departure time (Liao, Egbelu, & Chang, 2013; Ladier & Alpan, 2018; Molavi, Shahmardan, & Sajadieh, 2018; Van Belle et al., 2013; Fazel Zarandi, Khorshidian, & Akbarpour Shirazi, 2016). For example, Fazel Zarandi et al. (2016) Zarandi et al., 2016 studied a truck-scheduling problem with a JIT approach in a two-phase model. In the first phase, total earliness and tardiness of outbound trucks are minimized, and in the second phase, the number of preemption of outbound trucks is minimized. They developed three solution approaches to solve the model comprising integer programming, constraint satisfaction programming, and genetic algorithm. Ladier and Alpan, 2018 studied a truck scheduling problem when the number pallets transferred in the cross-docking center in each time period is limited with the number of workers and material handling equipment. The objective function was the minimization of total earliness and

* Corresponding author.
E-mail address: sajadieh@aut.ac.ir (M.S. Sajadieh).

https://doi.org/10.1016/j.cie.2019.106134
Received 31 December 2018; Received in revised form 30 June 2019; Accepted 21 October 2019
Available online 24 October 2019

Table 1
Literature review.

Papers	Partial Unloading	Compound Truck	Destination Assignment	Mixed Service	Product Allocation			Door In/Out	Different arrival	Objective	Solution Approach	Special Characteristics
					Dest	Post	Pre					
Chen and Lee (2009)								1,1		M	B&B, Heuristic	Modeling as a two-machine flow shop problem
Chen and Song (2009)								> 1, > 1		M	H	Extension of Lee and Chenc (14), Modeling as two-stage machine scheduling (flow shop) with as least two machine in a stage
Wooyeon and Egbelu (2008)						✓		1,1		M	H	Post-distribution cross-docking center
Arabani et al. (2011)						✓		1,1		M	GA, TS, PSO, ACO, DE	The same model presented by Wooyeon and Egbelu (2008)
Boysen (2010)							✓	> 1, > 1		FL, PT, TOT	DP, SA	Zero inventory, due date for outbound trucks
Lee et al. (2012)						✓		> 1, > 1		MTNSP	IGA	
Van Belle et al. (2013)							✓	> 1, > 1	✓	TWTT, T	TS	Soft time window constraint
Shakeri et al. (2012)		✓		✓		✓		> 1		M	H	Resource-constrained in cross-docking center
Joo and Kim (2013)		✓				✓		> 1, > 1		M	GA, SEA	Compound trucks are used as outbound trucks after unloading the products and moving to shipping dock doors
Arpan Rijal et al. (2019)				✓			✓	> 1, > 1	✓	C	ALNS	Evaluation of the position of mixed-mode dock doors and integrated scheduling and sequencing problems
Berghman et al. (2015)				✓			✓	> 1, > 1	✓	WSST	CS	Comparing the mixed and exclusive mode and examining the number of mixed dock doors for better solutions
Hermel et al. (2016)				✓			✓	> 1		M	D	Resource-constrained in cross-docking center
Dulebenets (2019)				✓			✓	> 1	✓	C	DSPEA	Considering parking area for temporary storage of products
Molavi et al. (2018)							✓	> 1, > 1	✓	M	GA-RVNS	Considering due date for outbound trucks and assessing the impact of shipment sorting
Antonino Chiarello et al. (2018)							✓	1,1		M	LR	"Truck synchronization at single door cross-docking terminals"
Ye et al. (2018)							✓	> 1, > 1		M	MPSO	Considering the order of unloading and loading of products
Vahdani et al. (2019)				✓			✓	> 1, > 1	✓	IH, DT, WT, EC	MOICA, MOGWO	Considering energy consumption by forklifts, waiting queue of trucks; considering both exclusive and mixed modes of services for dock doors
Wisittipanich and Hengmeechai (2017)						✓		> 1, > 1		M	MPSO	Using a modified particle swarm optimization for optimization
Serrano et al. (2017)					✓			> 1, > 1	✓	T	CS	Repacking of products inside cross-docking; capacity constraint of trucks and storage places
Alpan et al. (2011)			✓		✓			> 1, > 1		IHC, TRC	H	Studying transshipment scheduling for outbound trucks; destination assignment of outbound trucks
Larbi et al. (2009)			✓		✓			> 1, > 1		IHC, TRC	H	Studying transshipment scheduling for outbound trucks; destination assignment of outbound trucks
Larbi et al. (2011)			✓		✓			1,1		IHC, TRC	H	Transshipment problem for outbound trucks under full, partial, and no information of arrival of inbound trucks
Ladier and Alpan (2018)					✓			> 1, > 1	✓	T, E, IHC	H	Inbound and outbound truck scheduling and transshimpment planning under resource constraints
Assadi and Bagheri (2016)						✓		> 1, > 1	✓	E < T	DE, PBSA	
Zarandi et al. (2016)						✓		1,1	✓	E, T, NPfOT	IP, CP, GA	Extension to Wooyeon and Egbelu (2008); considering truck scheduling with preemption for outbound truck in JIT setting
Amini and Tavakkoli-Moghaddam (2016)						✓		1,1		Min T, Max R	NSGA-II, MOSA, MODE	Truck scheduling with due date when trucks can confront breakdown during the service

(continued on next page)

Table 1 (*continued*)

Papers	Partial Unloading	Compound Truck	Destination Assignment	Mixed Service	Product Allocation			Door In/Out	Different arrival	Objective	Solution Approach	Special Characteristics
					Dest	Post	Pre					
Mohtashami (2015)						✓		1,1		M	GA	Considering preemption for outbound trucks
Bodnar et al. (2015)				✓			✓	> 1, > 1	✓	C	ALNS	Considering exclusive and mixed mode of services for dock doors; decision on direct loading of products or storage
Keshtzari et al. (2016)						✓		1,1		M	PSO	Extension to Wooyeon and Egbelu (2008)
This Paper	✓	✓	✓	✓			✓	1,1	✓	M	RL-SA	Partial unloading, mixed mode of service, efficient RL-based SA algorithm

Solution Approaches: B&B: Branch and Bound; H: Heuristic(s); D: Decomposition; GA: Genetic Algorithm; IGA: Intelligent Genetic Algorithm; TS: Tabu Search; PSO: Particle Swarm Optimization; MPSO: Modified Particle Swarm Optimization; ACO: Ant Colony Optimization; DE: Differential Evolution; DP: Dynamic Prgoramming; SEA: Self-Evolution Algorithm; ALNS: Adaptive Large Neighborhood Search; DSPEA: Delayed Start Parallel Evolutionary Algorithm; RVNS: Reduced Variable Neighborhood Search; MOICA: multi objective imperialist competitive algorithm; MOGWO: multi objective grey wolf optimizer; PBSA: Population Based Simulated Annealing; IP: Integer Programming; CP: Constraint Programming; NSGA-II: Non-dominated Sorting Genetic Algorithm II; MOSA: Multi- Objective Simulated Annealing; MODE: Multi-Objective Differential Evolutionary; CS: Commercial Software; Lagrangian Relaxatoin.
Objective Function: M: Makespan; FT: Flow Time; PT: Processing Time; TOT: Tardiness of Outbound Trucks; T: Tardiness; E: Earliness; MTNSP: Maximum Total Number of Shipping Products; total TWTT: Total Weighted Travelling Time; C: Costs; WSST: Weighted Sum of Sojourn Times; IHC: Inventory Holding Cost; Delay of Trucks, WT: Waiting Time; EC: Energy Consumption; TRC: Truck Replacement Cost; NPfOT: Numer of Preemption for Outbound Trucks; R: Reliability;

tardiness as well as the total number of pallets places at storage. Three heuristic algorithms were developed to solve proposed model.

The literature can be classified into two general categories, multi-objective (e.g. see Amini & Tavakkoli-Moghaddam, 2016; Vahdani et al., 2019) and single objective problem where in the multi-objective setting, the decision maker tries to find Pareto optimal solutions. Makespan (total completion time of all operations inside the cross-docking center) is one of the common objectives in the literature (for example, see Chiarello, Gaudioso, & Sammarra, 2018; Arabani et al., 2011; Chen & Lee, 2009; Chen & Song, 2009; Hermel et al., 2016; Joo & Kim, 2013; Keshtzari, Naderi, & Mehdizadeh, 2016; Mohtashami, 2015; Molavi et al., 2018; Shakeri et al., 2012; Ye, Li, Li, & Hui, 2018; Wooyeon & Egbelu, 2008; Wisittipanich & Hengmeechai, 2017). When there is any due date for release time of inbound and\or outbound trucks minimizing earliness (when a truck leaves the center before its pre-defined due date) and\or tardiness (when a truck leaves the center after its pre-defined due date) plays an important role in the optimization problem (see, Amini & Tavakkoli-Moghaddam, 2016; Assadi & Bagheri, 2016; Bodnar et al., 2015; Boysen, 2010; Ladier & Alpan, 2018; Rijal, Bijvank, & de Koster, 2019; Serrano, Delorme, & Dolgui, 2017; Van Belle et al., 2013; Vahdani et al., 2019; Zarandi et al., 2016). Other objective functions are inventory holding costs, truck replacement costs, total traveling time, processing time and etc. Table 1 shows the objective functions studied in different papers.

Cross-docking centers can operate on either exclusive or mixed-mode of services. In exclusive mode, a dock-door is assigned to just either inbound or outbound trucks while in mixed-mode of service, a dock-door can be assigned to both types of trucks. This type of service increases the utilization of these centers.

Majority of papers study the exclusive mode of service (see for instance, Rahmanzadeh Tootkaleh et al., 2016; Vahdani & Zandieh, 2010; Van Belle et al., 2013 etc.).

On the other hand, some studies (Berghman, Briand, Leus, & Lopez, 2015; Bodnar et al., 2015) show that, despite higher complexity of operations handling, mixed mode of service increases the flexibility of cross-docking center, the mixed-mode of service can reduce the objective function by 18% and based on studies on real-world cross-docking centers, 5 out of 9 cross-docking centers are operating under mixed-mode of services Ladier and Alpan (2018). However, just a few papers have studied truck-scheduling problem under this assumption amongst with (Arpan Rijal et al., 2019; Berghman et al., 2015; Bodnar et al., 2015; Dulebenets, 2019; Hermel et al., 2016; Shakeri et al., 2012; Vahdani et al., 2019). Berghman et al. (2015) presented a mathematical program for truck scheduling such that some of dock doors are operating in mixed-mode to minimize the total weighted sojourn times. In this paper, an outbound truck can be assigned to a dock door if the precedence inbound truck is assigned to a dock door. They assessed how operating on mixed-mode of service will increase the chance to find feasible or optimal solutions and how changing some of exclusive dock doors to mixed-mode will increase the efficiency of cross-docking centers. Arpan Rijal et al. (2019) studied integrated scheduling and assignments of trucks in cross-docking center with mixed-mode of service for dock doors. They investigated the impact of integrated modeling of the problem and how mixed-mode of service will improve costs and how the position of mixed dock-doors effects the cost. Also, Shakeri et al. (2012) investigated truck scheduling problem in a resource constrained cross-docking center with mixed-mode of service in which pallets are moved to appropriate dock door if there is any forklift available. Hermel et al. (2016) also studied truck (container) scheduling with mixed-mode of service where there is resource limitation inside the cross-docking center. They first assigned trucks to dock doors and then scheduled trucks by considering resource limitations. Dulebenets (2019) considers a truck scheduling problem such that products are immediately transported to the dock door to be loaded into the appropriate outbound truck or are temporarily stored in the parking area until the appropriate outbound trucks is allocated to the dock door.

The author developed a delayed start parallel evolutionary algorithm to minimize total cost including total truck handling, truck waiting, earliness and tardiness, and storage space utilization costs. Moreover, Vahdani et al. (2019) investigate truck scheduling problem when forklifts are used for material handling inside the cross-docking center and one of the objective functions tries to minimize the energy consumption by these forklifts. On the other hand, trucks are assigned to dock doors with mixed-mode of services and scheduled such that total cost as the second objective function is minimized.

One of the characteristics, known as product allocation, which can be considered to distinguish between the types of cross-docking centers is when and how customers are assigned to products. Van Belle, Valckenaers, and Cattrysse (2012) defined two types of cross-docking centers based on product allocation criterion namely *pre-distribution* and *post-distribution*. On the other hand, Serrano et al. (2017) present a more general classification and add another category called *destination*. Based on the definition by Serrano et al. (2017), the production allocation is known as *Destination* if just the destination of products are known. Moreover, the production allocation is considered *pre-distribution* if in addition to the destination of products, the information on the exact outbound truck in which the product must be loaded in known. Finally, if the destination of products and the content of outbound trucks is defined by the number of product per each type, the allocation mode is known as *post-distribution*. Based on this definition, the production allocation mode of our model is *destination*. Table 1 shows the classification of some papers regarding production allocation characteristic of cross-docking centers.

There are some papers which study transshipment planning and scheduling problem inside the cross-docking centers such that some objective functions such as maximum number of products which are unloaded from inbound trucks and directly loaded into outbound trucks, minimum truck pre-emption costs, and etc. are optimized. In these problems, the manager determines not only the scheduling of outbound trucks, but also the assignment of outbound trucks to destinations. Larbi, Alpan, and Penz (2009) studied a transshipment problem inside a multi-door cross-docking center. In this problem, unloaded products are directly loaded into outbound trucks if the outbound trucks are available or temporary stored inside the center which incurs inventory holding cost. Moreover, it is possible to replace an outbound truck with another one to directly load product resulting a replacement costs. The model address the transshipment scheduling and assignment of outbound trucks to destinations to minimize total costs amongst with inventory holding and truck replacement costs. Alpan, Ladier, Larbi, and Penz (2011) developed several heuristics to solve the model presented by Larbi et al. (2009). Similarly, Larbi, Alpan, Baptiste, and Penz (2011) studied transshipment problem inside a cross-docking center under three conditions: first, there is full information regarding the order of arrival time and contents of inbound trucks; second, and third, there are partial and no information about the arrival of inbound trucks. In a more comprehensive model, Ladier and Alpan (2018) studied both inbound and outbound trucks scheduling and transshipment planning problem in a multi-door cross-docking center under resource constraints. In this paper, there is no difference between dock doors and the transportation time inside the cross-dock does not depend on the distance between dock doors (neglected) and the model determines the assignments of trucks to time slots and how products allocated to each destination and outbound trucks. Serrano et al. (2017) studied the scheduling of arrival time of inbound trucks, departure of outbound trucks, and material handling operations inside the cross-docking centers. In the paper, the repackaging operations inside the center, the capacity of storage area, the capacity of trucks and the dimension of components are considered. Bodnar et al. (2015) studied inbound and outbound truck scheduling when there are both exclusive and mixed dock doors to minimize operational cost including handling costs of storages and tardiness of outbound trucks. The model should determine the assignment of truck to time slots and if products should directly be loaded into outbound trucks or moved to storage area. They also investigate the impact of using a subset of dock doors in mixed mode with respect to total operational costs.

Recently, a few papers consider truck-scheduling problem in such a way that inbound trucks can also be used as outbound trucks after unloading of products (see for instance Joo & Kim, 2013). Although this assumption has been studied in the vehicle routing problem with cross docking (for instance, see Morais, Mateus, & Noronha, 2014; Tarantilis, 2013), truck scheduling decision has not been included. For more detail, readers are referred to Ladier and Alpan (2016),an Belle et al. (2013) and Boysen and Fliedner (2010). In the following, we review the paper more related to this research in details.

Joo and Kim (2013) studied a truck-scheduling problem in a cross-docking center with the exclusive mode of service for dock doors, where trucks are classified into three groups:

- *Inbound-only trucks*: Trucks, visiting only receiving dock doors, just bring incoming products to the cross-docking center to unload them.
- *Outbound-only trucks*: Trucks, visiting only shipping dock doors, are just responsible for delivery of products to destinations.
- *Compound trucks*: Trucks, visiting both receiving and shipping dock doors, that are responsible for both bringing incoming products to the cross dock, and then loading and delivery of products to destinations. In fact, compound trucks are inbound trucks which are then used as outbound trucks to deliver the demand of destinations.

The motivation of our paper is to compensate enumerated drawbacks of Joo and Kim's (2013) model Joo and Kim (2013). Despite the model by Joo and Kim (2013), we develop a truck scheduling model in a cross-docking center with a mixed mode of service, that is, a dock door can be assigned to both outbound and compound trucks. In this situation, two trucks with high amount of exchanged products can be assigned to the same dock door, and as a result, it hugely removes unnecessary transportation inside the cross-docking center. Therefore, it reduces makespan and the need to material handling equipment.

In addition, in the proposed model, it is not necessary to completely unload compound trucks and in this situation, i.e., some of the products can be kept in trucks without unloading. Besides,due to mixed-mode dock doors, compound trucks can start their loading operations without changing their dock doors (despite the model by Joo & Kim (2013)). Sensitivity analysis shows the impact of partial unloading on the objective function and how this assumption can result in lower makespan.

Moreover, because of partial unloading of compound trucks, the unloading and re-loading of time is saved for some products. As the destination assigned to each compound truck can change, the unloading time (compound trucks are not completely unloaded and the amount of unloaded products depends on the destination assigned to those trucks), the designation assignment should be optimized to minimize makespan. The contributions of this paper as follows:

- Addressing partial unloading of compound trucks and assessing the impact of partial unloading.
- Developing a truck scheduling in a cross docking with mixed mode of services.
- Developing an efficient heuristic algorithm to find good initial solutions in a very short computational time.
- Proposing a simulated annealing with a number of generic and tailor-made neighborhood search structures.
- Using intelligent and self-adaptive (reinforcement learning) approaches for neighborhood search structure selection.

it is notable that Shakeri et al. (2012) studied a truck scheduling problem where at first, trucks unload their products and then, load other products provided by other trucks. Based on these operations, it is possible to inferred that all trucks are operating as compound truck whereas it is not explicitly stated that these trucks are compound trucks or they are used in a closed-loop SCM where they bring and unload the demand of customers to the cross-docking center and then load the returned products from customers to deliver to suppliers. Moreover, no information is presented regarding partial unloading and destination assignment of trucks.

The rest of paper is organized as follows: Section 2 describes assumptions, notations, and proposed model. In Section 3, the heuristic and meta-heuristic approaches are presented. Computational results and sensitivity analysis are presented in Sections 4 and 5, respectively. Conclusions and future researches are provided in Section 6.

2. Problem description and formulation

Cross-docking (CD) center considered is one in a few-to-many configuration distribution network which is common in retailing supply chains (see Buijs, Vis, & Carlo, 2014). In this type of cross-docking networks, the number of origins is less than that of destinations in which a destination can be a single destination or a zone. Trucks are divided into two categories amongst with compound trucks and outbound trucks. Compound trucks bring products from origins to CD, unload some of them, and then are loaded with other products to be delivered. In Fig. 1, there are three compound trucks assigned to origins and destinations after partial unloading and reloading. Regarding the fact that the number of destinations can be larger than that of origins in a few-to-many distribution network, there should be some trucks for delivery of products from cross-docking center to destinations. These trucks which just load and deliver products to destinations are called outbound trucks.

As compound trucks can also be served as outbound trucks after unloading process, the demand of the assigned destination is not unloaded. Fig. 2 shows an example with four destinations and three compound trucks. Each color represents a destination; while circle, and triangle demonstrates product types. As the number of destinations is more than that of compound trucks, an outbound truck is needed to be allocated to the remained destination.

According to Fig. 3, compound trucks 1, 2, and 3 are assigned to destinations 2, 4, and 3, respectively. In this situation, compound truck 1 would not unload the demand for destination 2 (products in blue), and so on. When each compound truck unloads the demand of unassigned destinations (for example, the demand of destinations 1, 3, and 4 should be unloaded from compound truck 1), they are moved toward the appropriate dock doors to be loaded into their corresponding trucks. As destination 1 is not assigned to any compound trucks, outbound truck 1 deliver the demand of this destination.

The proposed model can be applied in both pre-distribution and post-distribution cross docking. In pre-distribution, at first, planning and scheduling at the cross-docking center should be done, and then, origins are informed about the destination assigned to each compound truck. The demand for assigned destination to each compound truck is then loaded.

On the other hand, this model can be applied in post-distribution cross docking such that all products loaded into a compound truck are the same. In this situation, partial unloading is possible if an origin provides single type products and each destination needs a known amount of products.

The main advantage of this model is the unloading, loading, internal transportation, makespan reduction, as well as less need to material handling equipment and resources. In other words, it reduces the delivery lead time from origins to destinations. Although partial unloading of compound trucks and mixed mode of service of dock doors increase the complexity of operations planning, the numerical results reveal the usefulness of proposed model and justify its applications in real-world cross-docking centers.

In this problem, cross-docking center is controlled in a centralized manner. As third-party logistics (3PLs) and less-than-truckload (LTL) companies usually use cross-docking as the major transportation strategy (see Ertek, 2012; Terreri, 2001), all these operations can be controlled under a unique entity. To make this planning easier, some technologies such as RFID, tracking tools, and Internet of Things can be useful. The assumptions, notation, and proposed model are presented then as follows:

2.1. Assumptions and notation

- Each truck, whether compound or outbound truck, should be assigned to just one destination which can be a single customer, or a zone. The number of compound trucks is less than or equal to that of doors, i.e. scheduling and sequencing of the compound truck is reduced to just the door assignment.
- All compound and outbound trucks are available at the beginning of horizon time.
- A pre-distribution cross-docking is considered, so products handled in the cross-docking center are not interchangeable.
- Travel time depends on dock doors assigned.
- There is a storage space in front of each dock door with enough capacity.

The objective is to find the best door assignment, destination assignment of compound and outbound trucks and the scheduling of outbound trucks to minimize makespan.

Indices

I	set of compound trucks
F	set of outbound trucks
K	set of product types
M	set of dock doors

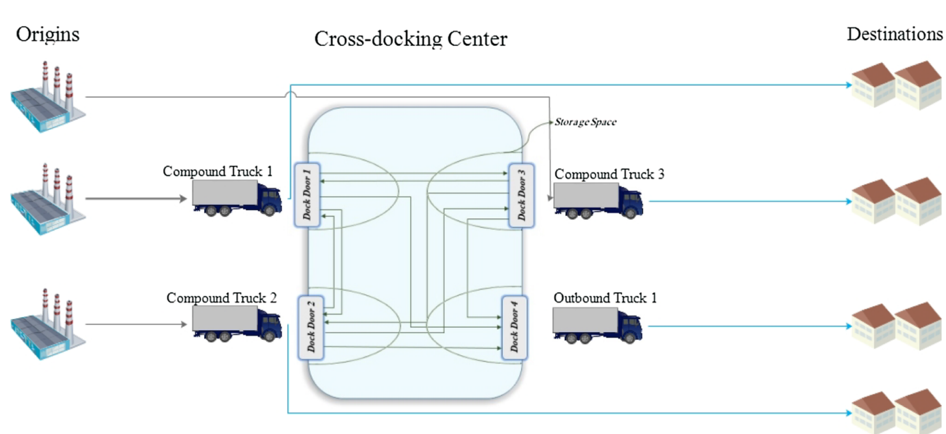

Fig. 1. The structure of studied distribution system with a cross-docking center.

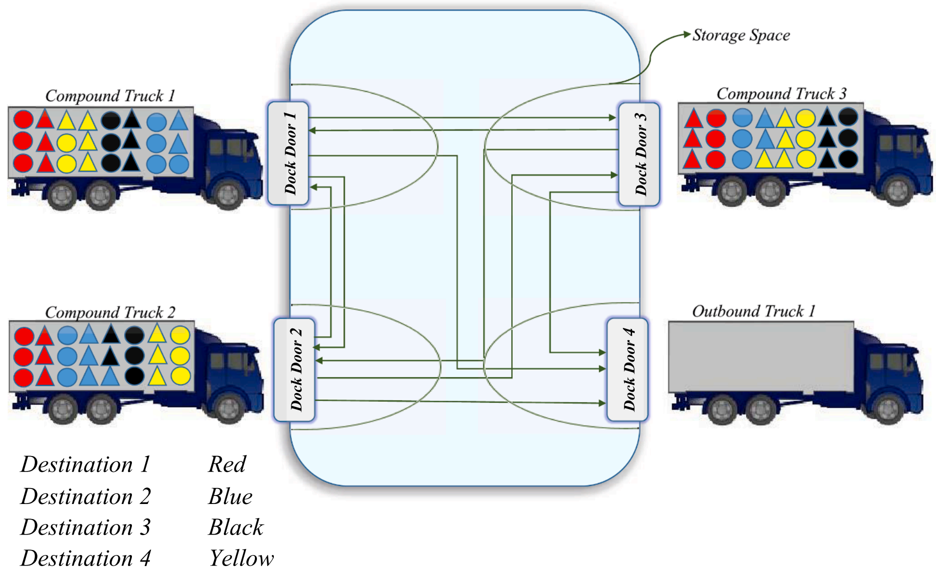

Fig. 2. Initial state of compound and outbound trucks.

D set of destinations

Problem parameters

- DE_i entering time of truck $i \in I \cup F$ (the time required for truck i to be located at dock door and be ready for unloading process).
- DL_i exit time of truck $i \in I \cup F$ (the time required for truck i to leave the dock door and move to the yard).
- f_{idk} the number of product type k initially loaded into the compound truck i that should be delivered to the destination d.
- t_{mn} the time required to move a unit of products from dock-door m to dock-door n.
- t_k loading/unloading time of a product unit of type k
- b_{id} 1 if compound truck i was loaded with products needed by destination d ($\sum_{k \in K} f_{idk} \geqslant 0$), otherwise 0 ($\sum_{k \in K} f_{idk} = 0$)

Continues variables

- τ makespan
- a_i time when loading of compound truck i starts
- d_f time when loading of outbound truck f starts

Binary Variable

- $Q_f\,(L_f)$ 1 if outbound truck f is processed as the first (last) one among outbound trucks assigned to the same door.
- P_{fl} 1 if outbound truck f and outbound truck l are both assigned to the same dock door and truck f precedes truck l, otherwise 0
- Y_{idm} 1 if compound truck i is assigned to the dock door m and destination d, otherwise 0
- Z_{fdn} : 1 if outbound truck f is assigned to dock door n and destination d, otherwise 0

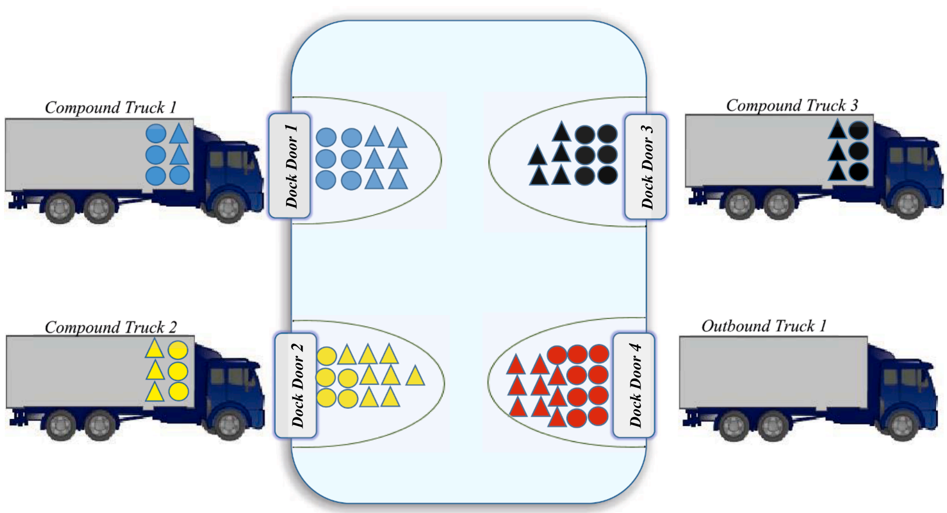

Fig. 3. Partial unloading of compound trucks.

2.2. Mathematical model

The proposed mathematical model to minimize the makespan is given below:

$$Min\ \tau$$

$$\tau \geqslant a_i + \sum_{j\in I\backslash i}\sum_{d\in D}\sum_{m\in M}\sum_{k\in K} f_{jdk}.\,t_k.\,Y_{idm} + DL_i \qquad \forall\, i \in I \tag{1}$$

$$\tau \geqslant d_f + \sum_{j\in I}\sum_{d\in D}\sum_{n\in M}\sum_{k\in K} f_{idk}.\,t_k.\,Z_{fdm} + DL_f \qquad \forall\, f \in F \tag{2}$$

$$a_i \geqslant DE_i + \sum_{d\in D}\sum_{k\in K} f_{idk}.\,t_k.\left(1-\sum_{m\in M} Y_{idm}\right) \qquad \forall\, i \in I \tag{3}$$

$$\begin{aligned} a_i \ &\geqslant DE_j + \sum_{d'\in D}\sum_{k\in K} f_{jd'k}.\,t_k.\left(1-\sum_{n\in M} Y_{jd'n}\right) + \sum_{m\in M}\sum_{d'\in D}\sum_{k\in K} f_{jdk}.\\ &\quad t_{mn}.\,Y_{jd'n}\\ &\quad - M_1.\,(1-b_{jd}.\,Y_{idm}) \qquad \forall\, i \neq j \in I, \forall\, m \in M, \forall\, d \in D \end{aligned} \tag{4}$$

$$\begin{aligned} d_f \ &\geqslant a_i + \sum_{j\in I\backslash i}\sum_{d\in D}\sum_{k\in K} f_{jdk}.\,t_k.\,Y_{idm} + DL_i + DE_f\\ &\quad - M_2.\left(2-\sum_{d\in D}(Y_{idm}+Z_{fdm})\right)\\ &\quad \forall\, i \in I, f \in F, m \in M \end{aligned} \tag{5}$$

$$\begin{aligned} d_f \ &\geqslant DE_i + \sum_{d'\in D}\sum_{k\in K} f_{jd'k}.\,t_k.\left(1-\sum_{m\in M} Y_{id'm}\right)\\ &\quad + \sum_{m\in M, m\neq n}\ \sum_{d'\in D}\sum_{k\in K} f_{idk}.\,t_{mn}.\,Y_{id'm}\\ &\quad - M_3.\,(1-b_{id}.\,Z_{fdn}) \qquad \forall\, i \in I, f \in F, n \in M, d \in D \end{aligned} \tag{6}$$

$$\begin{aligned} d_f &\geqslant d_l + \sum_{i\in I}\sum_{d\in D}\sum_{m\in M}\sum_{k\in K} f_{jdk}.\,t_k.\,Z_{ldm} + DL_l + DE_f - M_4\\ &\quad .\,(1-P_{lf}) \qquad \forall\, f \neq l \in F, m \in M \end{aligned} \tag{7}$$

$$Q_f + \sum_{l\in F\backslash f} P_{lf} = 1 \qquad \forall\, f \in F \tag{8}$$

$$L_f + \sum_{l\in F\backslash f} P_{fl} = 1 \qquad \forall\, f \in F \tag{9}$$

$$Q_f + Q_l \leqslant 3 - \sum_{d\in D}(Z_{fdn}+Z_{ldn}) \qquad \forall\, f \neq l \in F, n \in M \tag{10}$$

$$L_f + L_l \leqslant 3 - \sum_{d\in D}(Z_{fdn}+Z_{ldn}) \qquad \forall\, f \neq l \in F, n \in M \tag{11}$$

$$P_{fl} - 1 \leqslant \sum_{d\in D}(Z_{fdn}-Z_{ldn}) \leqslant 1 - P_{fl} \qquad \forall\, f \neq l \in F, n \in M \tag{12}$$

$$\sum_{d\in D}\sum_{m\in M} Y_{idm} = 1 \qquad \forall\, i \in I \tag{13}$$

$$\sum_{i\in I}\sum_{d\in D} Y_{idm} \leqslant 1 \qquad \forall\, m \in M \tag{14}$$

$$\sum_{d\in D}\sum_{n\in M} Z_{fdn} = 1 \qquad \forall\, f \in F \tag{15}$$

$$\sum_{i\in I}\sum_{m\in M} Y_{idm} + \sum_{f\in F}\sum_{n\in M} Z_{fdn} = 1 \qquad \forall\, d \in D \tag{16}$$

$$Y_{idm},\ Z_{fdn},\ P_{fl},\ Q_f,\ L_f \in \{0, 1\} \tag{17}$$

$$\tau,\ a_i,\ d_f \geqslant 0 \tag{18}$$

The objective function minimizes the makespan. Constraints (1) and (2) ensure that makespan is equal to the maximum completion time of all compound or outbound trucks. Constraints (3) state that compound trucks can start loading products when the unloading operations of demands for unassigned destinations is finished. No need to say, if the compound truck is assigned to a destination, the unloading time of that destination is not be considered. Constraints (4) make sure that all products should be available to be loaded into a specific compound truck. Constraints (5) declare that outbound truck can start loading products after the compound truck leaves the dock door, assuming both trucks are assigned to the same door. Outbound trucks can start loading products when products arrive at the assigned dock door, according to Constraints (6). Constraints (7) ensure that if two outbound trucks are allocated to the same door, loading process of latter truck begins when loading process of former is done. Constraints (8)–(12) determine the sequence of outbound trucks when two outbound trucks are assigned to the same door. Constraints (13) enforce that each compound truck needs to be allocated to just one door and one destination. Constraints (14) guarantee that at most, one compound truck can be assigned to each door. Each outbound truck should be dedicated to one door and one destination that it is depicted by Constraints (15). Constraints (16) ensure that each destination is assigned to exactly one of all trucks, either compound truck or outbound. Constraints (17) and (18) show the type of variables. Some constraints such as Constraints (4)–(7) should only be active if some conditions are satisfied. So, large positive numbers (M_1-M_4) are used to activate these constraints and areobtained through Eqs. (19)–(22).

$$\begin{aligned} M_1 \\ &= \max_{i\in I}(DE_i) + \max_{i\in I}\left(\sum_{d\in D}\sum_{k\in K} f_{idk}.\,t_k\right) + \max_{i\in I}\left(\sum_{d\in D}\sum_{k\in K} f_{idk}.\right.\\ &\quad \left.\max_{m,n\in M}(t_{mn})\right) \end{aligned} \tag{19}$$

$$M_2 = M_1 + \max_{d\in D}\left(\sum_{i\in I}\sum_{k\in K} f_{idk}.\,t_k\right) + \max_{i\in I}(DL_i) + \max_{f\in F}(DE_f) \tag{20}$$

$$\begin{aligned} M_3 \\ &= \max_{i\in I}(DE_i) + \max_{i\in I}\left(\sum_{d\in D}\sum_{k\in K} f_{idk}.\,t_k\right) + \max_{i\in I}\left(\sum_{d\in D}\sum_{k\in K} f_{idk}.\right.\\ &\quad \left.\max_{m,n\in M}(t_{mn})\right) \end{aligned} \tag{21}$$

$$M_4 = M_3 + F.\left(\max_{d\in D}\left(\sum_{i\in I}\sum_{k\in K} f_{idk}.\,t_k\right) + \max_{f\in F}(DL_f) + \max_{f\in F}(DE_f)\right) \tag{22}$$

It is notable that truck changeover time is equal to $DE_j + DL_i$ if truck j is the immediate successor to truck i.

3. Solution approach

If we relax the destination assignment of trucks in the proposed model, the problem reduces to a truck scheduling problem, which has been proved to be NP-hard (Kuo, 2013). Thus, the proposed model is also an NP-Hard problem and it can then be solved to optimality just for small problems in a reasonable time which is not applicable in real-world cross-docking centers. A heuristic algorithm and reinforcement learning-based simulated annealing algorithms are then used to find optimal/near optimal solution in large-scale problems.

3.1. Heuristic algorithm

This algorithm is a constructive and single stage which provides a feasible solution. One advantage of this algorithm is that it does not need any predefined parameter.

Because compound trucks both unload and load products, it takes a long time to finish their unloading and loading operation. Also, the destination assignment of compound trucks affects their operations. So, we assign each compound truck to a destination that leads to the least unloading and loading time. To find dock-door assignment of trucks, a truck with the highest unloading and/or loading time is in priority. Therefore, it should be assigned to most desirable dock door, i.e. the doors located in the middle of the cross-docking center. Based on this consideration, one truck is assigned to each dock door. For unassigned trucks, whenever a free dock door is available, a truck with the highest priority is assigned. In the following, parameters used in the heuristic method are presented:

T_{id} total partial unloading time of compound truck i as well as loading time of demand of destination d which was initially loaded in other compound trucks, if compound truck i to destination d.

$$T_{id} = \sum_{d' \in D, d \neq d'} \sum_{k \in K} f_{id'k} \cdot t_k + \sum_{j \in I \setminus i} \sum_{k \in K} f_{jdk} \cdot t_k \qquad \forall\, d \in D, i \in I \tag{23}$$

T_m the summation of time distance from other doors to dock-door m.

$$T_m = \sum_{n \in M} t_{mn} \tag{24}$$

UT_i total unloading time of compound truck i. This parameter is calculated after finding the destination allocated to compound trucks i (d_i).

$$UT_i = \sum_{d' \in D, d' \neq d} \sum_{k \in K} f_{id'k} \cdot t_k \qquad \forall\, i \in I \tag{25}$$

LT_d total loading time for demand of destination d. CT and OT stand for compound truck and outbound truck, respectively.

$$LT_d = \begin{cases} \sum_{j \in I \setminus l} \sum_{k \in K} f_{jdk} \cdot t_k & \text{if CT } l \text{ is assigned to destionation } d \\ \sum_{i \in I} \sum_{k \in K} f_{idk} \cdot t_k & \text{if an OT is assigned to destionation } d \end{cases} \tag{26}$$

T_d total waiting time for destination d. This parameter shows the time when the truck assigned to destination d can finish its loading process with the known allocation of trucks to destinations neglecting the time traveling between dock doors. This parameter considers the time when the trucks can start the loading process and total loading time. Eq. (27) shows this parameter.

$$T_d = \begin{cases} \max_{i \in I \setminus l}(UT_l,\ UT_i \cdot b_{id}) & \text{if CT truck } l \text{ is assigned to destination } d \\ \max_{i \in I}(UT_i,\ b_{id}) & \text{if an OT is assigned to destionation } d \end{cases} \tag{27}$$

TEE_f total entering and exiting time of outbound trucks. $TEE_f = DE_f + DL_f \quad \forall\, f \in F$

d_f^1 the time when outbound truck f starts loading process based on Constraints (5). This value is calculated for outbound trucks which are successor to a compound truck, assuming both are assigned to the same dock door, where a_i^1 and a_a^2 are calculated from Constraints (3) and (4) with known Y_{idm} and $Y_{jd'n}$, respectively.

$$d_f^1 = \max(a_i^1,\ a_a^2) + \sum_{j \in I} \sum_{k \in K} f_{jd_ik} \cdot t_k + DL_i + DE_f \quad f \in F, i \in I \tag{28}$$

d_f the time when outbound truck f stars its loading process, where d_d^2 and d_d^3 are calculated from constraints (6) and (7) with known $Y_{id'm}$, Z_{fdn} and P_{fl}. $d_f = \max(d_f^1, d_f^2, d_f^3) \quad \forall\, f \in F$.

FT_i t he time when the compound truck i leaves the assigned door.

$$FT_i = a_i + \sum_{j \in I \setminus i} \sum_{k \in K} f_{jd_ik} \cdot t_k + DL_i \qquad \forall\, i \in I \tag{29}$$

FT_f the time when the outbound truck f leaves assigned dock door.

$$FT_f = d_f + \sum_{i \in I} \sum_{k \in K} f_{id_fk} \cdot t_k + DL_f \qquad \forall\, f \in F. \tag{30}$$

FT_m the time when operations on dock door finish. This parameter is updated when operations of truck assigned to this dock door finishes.

$$FT_m = \begin{cases} FT_i & \text{ifCT } i \text{ is assigned todock door } m \\ FT_m & \text{ifOT } f \text{ is assigned todock door } m \end{cases} \tag{31}$$

FAT *(First Assigned Trucks)* a list of compound and outbound trucks processed as the first truck on each door. Thus, the number of trucks in the list is equal to the number of doors. In order to minimize makespan ($C_{max} = \max_{m \in M} FT_m$), trucks with higher processes time are processed first. Processing compound trucks are expected to be more time consuming as they should unload and reload the products. So, all compound trucks are on the list. If the number of doors exceeds that of compound trucks, some of the outbound trucks can also be processed as the first truck. These outbound trucks are the ones with the longer loading process.

The steps of the proposed heuristic algorithm are as follows:

Step 1: Calculate T_{id} for each pair of compound truck and destination using Eq. (23).

Step 2: Use Vogel's approximation algorithm to allocate each compound truck to one destination. Rest of destinations will be assigned to outbound trucks.

Step 3: Calculate UT_i, LT_d and T_d for each destination using Eqs. (25)–(27).

Step 4: Find TEE_f for each outbound truck and sort them in ascending order.

Step 5: Assign the outbound truck with lowest TEE_f to an unassigned destination with highest T_d. Remove these destination and outbound truck from the list. Repeat this step till all outbound trucks are allocated.

Step 6: Calculate T_m using Eq. (24) and sort them in ascending order.
Step 7: Construct *FAT* and sort it in descending order based on T_d.
Step 8: Assign truck in *FAT* list with highest T_d to dock door with lowest T_m. In the case of more than one dock door with the same value of T_m, dock door with the lowest destination from assigned dock doors in the previous steps is selected. Remove the truck and the dock door selected from the list and repeat this step if there is any unassigned truck on the list.
Step 9: Calculate a_i^1, a_i^2, FT_i and FT_m using Eqs. (29) and (31) for each compound truck and assigned dock door.
Step 10: Calculate d_i^1, d_i^2, d_i^3, d_i and FT_f for outbound trucks in *FAT* list and FT_m using Eqs. (28), (30), and (31).
Step 11: Assign remained outbound truck with highest T_d to dock door with lowest FT_m. Remove outbound truck assigned from the list. Calculate d_f^1, d_f^2, d_f^3, d_f, FT_f and FT_m using Eqs. (28), (30) and (31). Repeat this step to allocate all remained outbound trucks.
Step 12: Calculate C_{max}.

Vogel's approximation algorithm (VAA) employed in Step 2 is a known heuristic algorithm used to find a good basic solution for Transportation Problem (TP) (see Korukoğlu & Ballı, 2011). The basis of VAA is based on the concept of penalty cost or regret. In this paper, this algorithm is used to assign compound trucks to destinations. It is possible that two compound truck are competing to be allocated to the same destination (both trucks i and j have the lowest T_{id} and T_{jd}), while one of them should be selected. Therefore, the regret is calculated for each one, i.e. the penalty if the truck is not assigned to this destination. Based on these regrets, a good answer for the assignment problem is obtained.

3.2. Simulated annealing

This section presents the simulated annealing algorithm developed to solve the proposed model. Generally, SA starts the optimization process from a random initial solution while in this paper, it takes the advantage of heuristic algorithm as initial solution. As numerical results show, this heuristic algorithm can provide good initial solutions in a very short computational time. Moreover, several generic and problem-specific neighborhood structures are developed to efficiently explore solution space. In addition, in order to consciously select a neighborhood structure to generate a new solution, several reinforcement learning approached are used to learn the value and efficiency of each neighborhood structure. In the following, some aspects of SA algorithm are presented. Besides, reinforcement learning approaches are presented in the next Section 3.3.

Solution Representation: The destination and dock-door assignment and sequencing of each compound and outbound truck are represented in a solution. A $3 \times D$ matrix is used for this purpose. In this matrix, columns 1 to I and columns $|I| + 1$ to D correspond to compound and outbound trucks, respectively. The first and the second rows shows the destination allocation of trucks and dock-door assignment, respectively. Finally, the third row demonstrates the sequencing of all trucks. Fig. 4 shows proposed solution representation of the model.

In this paper, some generic and some tailor-made neighborhood search structures are developed to efficiently explore and exploit solution spaces. Here, we present these neighborhood search structures:

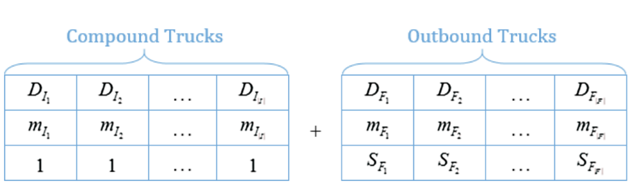

Fig. 4. Solution representation of SA algorithm.

- *Swap of destinations*, $k = 1$: Two trucks, either compound or outbound, are randomly chosen and their assigned destinations are swapped.
- *Swap of dock door for compound (outbound) trucks*, $k = 2$ ($k = 3$): Two compound (outbound) trucks are randomly chosen and their assigned dock doors are swapped.
- *Insertion of outbound trucks*, $k = 4$: An outbound truck is chosen and assigned to another door.

In addition to these generic neighborhood search structures, some problem-specific ones are developed as follows:

- *Insertion of Compound Truck to a Dock Door*, $k = 6$: Based on unloading time of compound truck (UT_i), one of them is selected according to roulette wheel selection mechanism and is assigned to a dock door selected through roulette wheel on the basis of T_m.
- *Insertion of outbound Truck to a dock door*, $k = 7$: Based on loading time of outbound truck (T_{d_f} total loading time of destination assigned to outbound truck f), one of them is selected according to roulette wheel selection mechanism and is assigned to a dock door selected through roulette wheel on the basis of T_m.
- *Insertion of compound Truck to a destination*, $k = 8$: A compound truck is selected according to $\max_{d \in D} T_{id}$ and is assigned to a destination. Both selections are done through roulette wheel selection mechanism.

It is worth mentioning that the SA algorithm uses the final solution of developed heuristic algorithm as the initial solution. As numerical results represent, it help the SA algorithm to find better solutions in shorter computational time. Moreover, we use $T_{new} = \alpha.\ T_{old}$ as the cooling scheme and to equally compare all algorithms, we set a total computational CPU time limitation as the termination criteria. In this regards, all algorithms are run for $\left(\frac{I+D}{2} \times M \times 0.7\right)$seconds.

3.3. Reinforcement learning-based selection mechanism

One of the main concern in meta-heuristic algorithms is that how neighborhood search structures (NS) are selected to generate a new solution. In most cased, NSs are randomly selected and it is not obvious if this NS is a good choice or not. On the other hand, it possible to oversee the performance of each NS and then on this basis, select a NS. For example, if a NS is selected while no improvement is achieved, it may imply that this neighborhood search is not appropriate for the current state, so it should receive some punishment. Contrarily, if a NS provides significant improvement, it may be inferred that this NS is good enough and should receive some reward. This concept is modeled as a reinforcement learning problem and the model can learn the value of each action by tracing the punishment and reward of each action. Fig. 5 represents the general concept of reinforcement learning and how it interacts with environment.

Multi-armed bandit (MAB) is one of the basic problems in reinforcement learning. Suppose there are K slot machines and each machine provides different rewards according to a probability distribution. The player who does not know the probability distribution of reward of each slot machine should maximize the expected value of his

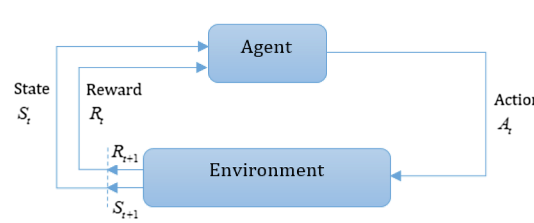

Fig. 5. Agent-Environment interaction in RL (adopted from Sutton et al. (1998)).

Table 2
Comparison of all algorithms according to objective function in small-scale problems.

#	Size	Exact	H	SA	SA-NS	SA-RL1	SA-RL2	SA-RL3	SA-RL4	SA-RL5	SA-RL6
1	(2,3,2)	257.00	257.00	257.00	257.00	257.00	257.00	257.00	257.00	257.00	257.00
2	(2,3,2)	724.00	724.00	724.00	724.00	724.00	724.00	724.00	724.00	724.00	724.00
3	(2,3,2)	652.00	661.00	652.00	652.00	652.00	652.00	652.00	652.00	652.00	652.00
4	(2,4,3)	1368.00	1394.00	1368.00	1368.00	1368.00	1368.00	1368.00	1368.00	1368.00	1368.00
5	(2,4,3)	1020.00	1086.00	1020.00	1020.00	1020.00	1020.00	1020.00	1020.00	1020.00	1020.00
6	(2,4,3)	1644.00	1692.00	1675.00	1644.00	1644.00	1644.00	1644.00	1644.00	1644.00	1644.00
7	(3,4,4)	1715.00	1846.00	1715.00	1715.00	1715.00	1715.00	1715.00	1715.00	1715.00	1715.00
8	(3,4,4)	1543.00	1723.00	1543.00	1543.00	1543.00	1543.00	1543.00	1543.00	1543.00	1543.00
9	(3,4,4)	1721.00	1796.00	1721.00	1721.00	1721.00	1721.00	1721.00	1721.00	1721.00	1721.00
10	(3,5,3)	617.00	624.00	617.00	617.00	617.00	617.00	617.00	617.00	617.00	617.00
11	(3,5,3)	2009.00	2037.00	2009.00	2009.00	2009.00	2009.00	2009.00	2009.00	2009.00	2009.00
12	(3,5,3)	1973.00	1996.00	1973.00	1973.00	1973.00	1973.00	1973.00	1973.00	1973.00	1973.00
13	(3,6,4)	1890.00	1944.00	1890.00	1890.00	1890.00	1890.00	1890.00	1890.00	1890.00	1890.00
14	(3,6,4)	2107.00	2177.00	2115.75	2107.85	2107.05	2107.20	2107.15	2107.25	2107.25	2107.35
15	(3,6,4)	2753.00	2851.00	2800.80	2753.95	2753.45	2753.45	2753.50	2753.60	2753.60	2753.65
16	(3,6,4)	2726.00	2900.00	2789.00	2727.15	2726.00	2726.00	2726.10	2726.10	2726.05	2726.30
17	(3,6,4)	1527.00	1631.00	1575.40	1527.95	1527.35	1527.35	1527.10	1527.60	1527.30	1527.60
18	(3,6,4)	910.00	931.00	927.05	911.00	910.00	910.00	910.25	910.15	910.10	910.50
19	(4,6,4)	1760.00	1823.00	1760.15	1760.60	1760.00	1760.00	1760.00	1760.00	1760.15	1760.15
20	(4,6,4)	2951.00	3068.00	2951.05	2952.85	2951.05	2951.10	2951.00	2951.05	2951.00	2951.20
21	(4,6,4)	1609.00	1650.00	1609.40	1611.10	1609.10	1609.00	1609.00	1609.10	1609.10	1609.40
22	(4,7,4)	6360.00	6447.00	6360.25	6362.00	6360.05	6360.00	6360.05	6360.05	6360.05	6360.45
23	(4,7,4)	2320.00	2445.00	2322.40	2327.10	2320.10	2320.20	2320.65	2320.30	2320.80	2321.50
24	(4,7,4)	1448.00	1495.00	1449.80	1451.35	1449.30	1449.55	1449.50	1449.15	1449.25	1449.55
25	(5,7,6)	4681.00	4682.00	4681.00	4681.00	4681.00	4681.00	4681.00	4681.00	4681.00	4681.00
26	(5,7,6)	2612.00	2771.00	2619.80	2618.60	2617.15	2616.25	2615.60	2616.95	2615.10	2616.80
27	(5,7,6)	2536.00	2699.00	2536.20	2536.20	2536.00	2536.10	2536.00	2536.00	2536.00	2536.00
28	(5,7,5)	4113.00	4197.00	4113.00	4116.00	4113.00	4113.05	4113.10	4113.00	4113.65	4113.00
29	(5,7,5)	1775.00	1847.00	1775.40	1786.50	1775.00	1775.60	1775.15	1775.10	1775.25	1775.65
30	(5,7,5)	1634.00	1758.00	1635.90	1646.75	1635.00	1635.00	1635.00	1635.00	1635.00	1635.50
31	(6,7,6)	3923.00	4171.00	3923.00	3923.00	3923.00	3923.00	3923.00	3923.00	3923.00	3923.00
32	(6,7,6)	1679.00	1852.00	1679.00	1679.00	1680.20	1680.15	1679.70	1679.70	1679.60	1679.35
33	(6,7,6)	1992.00	2179.00	1992.00	1992.00	1992.00	1992.00	1992.00	1992.00	1992.00	1992.00
34	(5,8,5)	5146.00	5207.00	5148.10	5167.30	5144.70	5144.60	5145.15	5145.30	5145.25	5149.55
35	(5,8,5)	2761.00	2811.00	2762.75	2785.50	2762.30	2762.10	2761.35	2761.50	2762.65	2763.70
36	(5,8,5)	1678.00	1775.00	1678.00	1681.05	1678.00	1678.00	1678.00	1678.00	1678.00	1678.00
37	(5,9,6)	6473.00	6695.00	6474.20	6473.90	6473.85	6473.65	6474.00	6473.65	6474.05	6474.10
38	(5,9,6)	2990.00	3307.00	2993.25	3006.95	2991.60	2990.95	2991.25	2991.70	2991.15	2989.90
39	(5,9,6)	1618.00	1718.00	1635.95	1638.15	1637.20	1635.35	1636.50	1635.85	1634.40	1637.05
40	(6,9,6)	7158.00	7349.00	7158.45	7158.45	7158.60	7158.65	7158.40	7158.85	7158.55	7158.50
41	(6,9,6)	3557.00	3666.00	3555.65	3556.60	3556.05	3556.65	3556.40	3556.95	3556.10	3556.00
42	(6,9,6)	2004.00	2050.00	2008.35	2008.15	2007.30	2008.25	2007.50	2008.65	2007.30	2008.20
Average		2427.00	2522.19	2433.22	2430.50	2427.82	2427.79	2427.77	2427.85	2427.75	2428.05

gained rewards meanwhile learn the true value of expected reward of each slot machine. In our problem, each neighborhood search structure can be considered a slot machine. We want to select a NS that maximizes the expected value of probability of improvement in the current solution. Here, the reward is defined as 1 if the objective value of new solution is equal or better than that of current solution, otherwise 0. In this paper, four variants of MAB are addressed to learn the true value function of each NS.

Suppose the value function of each action (neighborhood search structure) is represented by $Q(a) \quad \forall\, a \in A$, and $N(A)$ is the number of times that action (NS) A is selected. Moreover, assume that when an action is taken, the system receive reward R, where α shows the learning rate. In this situation, each of four MAB variants can be presented as follows:

Incremental Implementation: This method estimates the value function of each action by averaging the rewards actually received. Incremental implementation which uses less memory to store data, updates value functions as follows:

$$Q(a) \leftarrow Q(a) + \frac{1}{N(a)}.\ [R - Q(a)]$$

Non-Stationary Problem: In incremental implementation, it is supposed that the environment is stationary and the probability distribution of reward of each slot machine is the same all the time. However, this assumption can be easily violated and the current approach updates the value functions with different weight to follow any changes in probability distribution as follows:

$$Q(a) \leftarrow Q(a) + \alpha.\ [R - Q(a)]$$

Upper-Confidence-Bound Action Selection - Type 1: Many algorithms use ϵ -greedy method to try non-greedy actions and so, learn their value functions. However, this method, does not take into account how good each action is. Upper-confidence-bound (UCP) method chose an action with largest possible value of value function in a confidence interval as follows:

$$A_t = \underset{a \in A}{\operatorname{argmax}} \left[Q_t(a) + c.\ \sqrt{\frac{lnt}{N_t(a)}} \right]$$

It is worth mentioning that in this formulation, $Q_t(a)$ is updated same as incremental implementation method.

Upper-Confidence-Bound Action Selection - Type 2: This method is same as the former method while $Q_t(a)$ is updated similar to non-stationary problem one.

These method are some common methods to balance exploration and exploitation in RL field. However, these are not considered fully RL methods. On the other hand, there are some state variables (contextual variables in MAB literature) which may help us to make decisions.

Table 3
Comparison of all algorithms according to computational time in small-scale problems.

		CPU Time (s)			Best found solution (s)							
#	Size	Exact	H	Meta	SA	SA-NS	SA-RL1	SA-RL2	SA-RL3	SA-RL4	SA-RL5	SA-RL6
1	(2,3,2)	0.05	0.15	3.50	0.04	0.03	0.04	0.04	0.04	0.04	0.04	0.04
2	(2,3,2)	0.14	0.14	3.50	0.04	0.03	0.04	0.04	0.04	0.04	0.04	0.04
3	(2,3,2)	0.08	0.14	3.50	0.05	0.04	0.05	0.04	0.05	0.05	0.05	0.05
4	(2,4,3)	0.19	0.13	6.31	0.06	0.05	0.06	0.07	0.06	0.06	0.06	0.06
5	(2,4,3)	0.44	0.14	6.31	0.10	0.11	0.10	0.11	0.10	0.10	0.13	0.08
6	(2,4,3)	0.27	0.13	6.31	0.11	0.15	0.15	0.14	0.17	0.13	0.14	0.08
7	(3,4,4)	0.14	0.15	9.81	0.12	0.16	0.26	0.27	0.33	0.23	0.16	0.13
8	(3,4,4)	0.34	0.15	9.81	0.09	0.73	0.41	0.48	0.50	0.52	0.37	0.38
9	(3,4,4)	0.20	0.15	9.81	0.12	0.34	0.34	0.33	0.29	0.24	0.22	0.20
10	(3,5,3)	0.45	0.14	8.40	0.54	1.48	0.80	0.79	0.85	0.85	0.53	0.86
11	(3,5,3)	0.44	0.13	8.40	0.53	1.04	0.96	0.65	0.72	0.85	0.51	0.53
12	(3,5,3)	0.42	0.13	8.40	0.28	0.61	0.42	0.28	0.45	0.35	0.29	0.23
13	(3,6,4)	8.75	0.13	12.60	1.77	3.54	1.85	2.93	2.45	2.59	2.58	2.04
14	(3,6,4)	8.25	0.14	12.60	6.65	4.25	7.15	6.22	7.12	6.54	6.43	7.41
15	(3,6,4)	10.67	0.13	12.60	6.52	5.37	6.94	7.42	7.38	7.18	6.87	6.56
16	(3,6,4)	10.30	0.14	12.61	7.39	7.31	6.28	8.40	6.26	7.22	7.68	6.93
17	(3,6,4)	9.34	0.14	12.61	7.49	5.83	7.79	8.91	7.48	6.75	6.86	6.89
18	(3,6,4)	7.28	0.14	12.61	5.91	6.43	7.25	6.55	6.85	7.05	7.24	8.11
19	(4,6,4)	4.91	0.13	14.01	6.87	5.56	7.38	5.42	5.61	6.03	8.18	5.70
20	(4,6,4)	6.27	0.13	14.00	10.31	6.41	7.64	8.43	6.48	8.17	8.34	7.82
21	(4,6,4)	11.09	0.13	14.00	8.35	5.86	6.99	7.73	8.31	9.43	9.01	8.52
22	(4,7,4)	43.20	0.13	15.40	12.82	9.01	10.19	9.67	10.44	10.82	10.70	11.15
23	(4,7,4)	64.33	0.13	15.41	11.16	9.47	12.76	12.87	12.89	11.57	11.71	13.77
24	(4,7,4)	78.73	0.13	15.41	10.22	8.76	11.64	11.41	10.12	11.19	11.74	10.16
25	(5,7,6)	1645.61	0.14	25.20	0.61	0.82	0.69	0.60	0.61	0.48	0.67	0.53
26	(5,7,6)	1143.83	0.14	25.20	13.48	20.95	11.73	13.69	12.49	13.08	14.17	15.14
27	(5,7,6)	1049.80	0.14	25.21	12.14	15.97	10.01	9.84	10.08	11.07	11.07	12.25
28	(5,7,5)	199.41	0.13	21.01	10.33	14.03	10.32	11.61	10.72	10.24	10.94	11.18
29	(5,7,5)	73.53	0.13	21.01	18.35	13.87	16.63	16.67	16.11	17.50	17.49	17.33
30	(5,7,5)	173.56	0.13	21.01	18.01	12.63	15.73	17.08	15.62	16.95	17.76	19.33
31	(6,7,6)	700.31	0.15	27.31	1.91	1.18	1.85	3.27	2.33	2.87	3.43	1.44
32	(6,7,6)	558.84	0.15	27.31	14.68	14.51	11.11	13.32	12.72	13.27	14.44	15.73
33	(6,7,6)	251.38	0.15	27.31	1.79	1.85	2.35	2.78	2.50	3.27	2.20	2.38
34	(5,8,5)	8277.33	0.13	22.76	19.24	13.87	18.81	18.90	19.54	19.82	20.39	20.18
35	(5,8,5)	1611.88	0.13	22.76	20.98	16.70	18.80	19.13	18.47	18.60	19.91	19.59
36	(5,8,5)	4986.67	0.13	22.76	15.40	15.25	14.87	13.16	14.60	13.44	14.04	13.66
37	(5,9,6)	10800.13	0.14	29.41	17.38	25.09	15.33	17.00	16.11	16.70	16.14	18.35
38	(5,9,6)	10800.00	0.18	29.41	24.43	24.93	20.42	21.16	22.03	22.23	22.27	24.75
39	(5,9,6)	10800.00	0.14	29.41	19.21	24.34	15.43	15.94	16.74	16.18	17.99	17.89
40	(6,9,6)	10800.00	0.13	31.51	22.71	21.83	20.45	19.76	20.79	21.65	21.89	23.59
41	(6,9,6)	10800.39	0.13	31.51	26.53	25.21	22.47	22.69	23.03	23.02	23.40	24.97
42	(6,9,6)	10800.63	0.13	31.51	25.32	24.95	22.13	22.91	22.67	22.83	24.11	25.92
Average		2041.42	0.14	17.13	9.05	8.82	8.25	8.54	8.39	8.60	8.86	9.09

Q-learning: Q-Learning is a kind of reinforcement learning algorithms (off-policy) which tries to find optimal (near-optimal) solutions in Markov Decision Process (MPD) problems. In order to use Q-learning for action selection, it is important to define states, actions, and rewards. In this context, actions and rewards in Q-learning are the same as the ones in former MAB methods, while states are defined based on the number of times that current solution is not improved (improvement is occurred if objective function of new solution is as good as the current solution). If *NI* represents the number of times that the current solution is not improved, state variable is defined as follows (Ahmadi, Goldengorin, Süer, & Mosadegh, 2018):

$$S = \begin{cases} 1 & 0 \leqslant NI \leqslant \nu_1 \\ 2 & \nu_1 \leqslant NI \leqslant \nu_2 \\ 3 & \nu_2 \leqslant NI \leqslant \nu_3 \\ 4 & \nu_3 \leqslant NI \leqslant \nu_4 \\ 5 & \nu_4 \leqslant NI \end{cases}$$

Suppose that the system state is s_t and action a_t is chosen and reward r_t is received. Therefore, the action-value function is updated as follows where $\gamma \in (0, 1)$ is discount factor.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha.\ [r_t + \gamma.\ \max_{b \in A} Q(s_{t+1}, b) - Q(s_t, a_t)]$$

Note: It is of great importance to mention that in order to simultaneously exploit current good neighborhood search structure and exploit other ones, ϵ percent of the times a NS is randomly chosen and P_{rol} percent of the times a NS is chosen through roulette wheel mechanism based on state-action or action-value functions. Otherwise, a greedy NS based on one of the reinforcement learning algorithms is selected as follows Ahmadi et al. (2018). For more details about MAB, Q-learning and generally RL, see Sutton et al. (1998).

$$NS = \begin{cases} \text{Random Selection} & \text{if } rand \leqslant \epsilon \\ \text{Roullete wheel based on } Q(a)\backslash Q(s, a), & \text{if } \epsilon < rand \leqslant \epsilon + Pr_{rol} \\ \text{Greedy Selection based on } Q(a)\backslash Q(s, a), & \text{if } rand \geqslant \epsilon + Pr_{rol} \end{cases}$$

4. Experimental results

To evaluate the performance of proposed algorithms, it is necessary to assess these methods in different test problems. As there is no standard test problem in truck scheduling problems, data sets are generated randomly on the basis of data used by Van Belle et al. (2013). In this paper, an I-shape cross-docking center is studied and transporting time

Table 4
Comparison of all algorithms according to objective function in large-scale problems.

#	Size	H	SA	SA-NS	SA-RL1	SA-RL2	SA-RL3	SA-RL4	SA-RL5	SA-RL6
1	(8,10,8)	3388.00	3269.23	3268.70	3254.33	3253.45	3254.95	3253.28	3253.98	3253.45
2	(8,10,8)	5603.00	5382.65	5383.30	5379.55	5378.60	5377.25	5376.80	5376.85	5377.50
3	(8,10,8)	4992.00	4944.60	4946.00	4937.93	4935.88	4936.83	4940.15	4933.70	4934.28
4	(8,12,9)	6332.00	6144.40	6144.55	6139.55	6139.30	6139.50	6139.45	6139.60	6139.30
5	(8,12,9)	6587.50	6337.03	6330.58	6331.73	6323.53	6321.30	6324.85	6325.13	6320.00
6	(8,12,9)	6272.00	6102.00	6102.70	6099.00	6098.15	6096.05	6095.70	6096.50	6097.55
7	(10,13,10)	3002.00	2735.38	2733.65	2733.05	2733.25	2732.45	2732.93	2732.25	2729.75
8	(10,13,10)	6081.50	5619.18	5616.65	5614.13	5616.70	5621.85	5612.03	5612.08	5618.05
9	(10,13,10)	6792.00	6065.10	6068.40	6074.00	6068.10	6059.60	6066.05	6062.00	6058.95
10	(10,15,11)	3614.00	3198.85	3198.05	3193.75	3195.50	3195.55	3191.10	3191.40	3191.75
11	(10,15,11)	5783.00	5708.80	5706.85	5701.40	5702.50	5701.45	5704.30	5700.05	5702.10
12	(10,15,11)	3912.00	3788.20	3787.75	3783.05	3783.90	3783.55	3784.05	3783.95	3781.95
13	(11,15,12)	5611.00	4858.50	4858.50	4853.50	4853.50	4853.50	4853.50	4853.50	4853.50
14	(11,15,12)	5562.00	4897.35	4895.55	4888.75	4894.60	4891.85	4898.20	4891.90	4890.90
15	(11,15,12)	5441.00	5134.35	5134.80	5135.75	5141.85	5126.80	5131.85	5144.45	5135.95
16	(12,16,12)	3812.00	3543.25	3547.85	3541.35	3538.00	3540.10	3538.65	3540.15	3534.60
17	(12,16,12)	5120.00	4856.55	4849.80	4846.50	4847.65	4854.15	4846.75	4851.25	4851.55
18	(12,16,12)	3741.00	3572.20	3576.05	3567.05	3567.85	3562.55	3569.45	3568.45	3564.25
19	(12,18,14)	4093.00	3652.40	3668.55	3647.15	3657.45	3655.50	3644.65	3651.95	3649.00
20	(12,18,14)	3487.00	3054.35	3057.50	3048.90	3049.25	3053.60	3047.65	3050.20	3053.45
21	(12,18,14)	4337.00	4142.70	4142.35	4133.50	4138.55	4135.60	4137.85	4134.50	4134.05
22	(14,19,14)	5869.00	5374.00	5382.40	5366.10	5358.75	5362.30	5362.10	5372.45	5364.90
23	(14,19,14)	5174.00	4976.70	4975.90	4972.85	4969.90	4971.10	4974.00	4972.50	4972.60
24	(14,19,14)	5637.00	5229.95	5224.00	5220.00	5231.65	5231.95	5223.00	5223.65	5223.50
25	(14,22,16)	6041.00	5646.23	5643.88	5640.60	5638.25	5640.95	5643.18	5640.70	5641.05
26	(14,22,16)	5795.00	5586.80	5581.50	5583.40	5578.90	5581.30	5576.10	5580.25	5577.60
27	(14,22,16)	5354.00	5323.50	5319.25	5317.45	5313.20	5316.45	5315.90	5314.40	5315.20
28	(15,23,15)	7429.00	6982.40	6982.15	6978.00	6982.90	6978.80	6986.50	6987.95	6994.10
29	(15,23,15)	6190.00	5980.00	5978.05	5972.05	5970.20	5973.95	5971.10	5977.15	5976.90
30	(15,23,15)	5992.00	5769.75	5768.10	5765.40	5766.25	5761.90	5761.75	5765.70	5764.10
31	(17,25,17)	6291.50	5740.78	5734.13	5745.80	5738.95	5750.38	5731.73	5728.88	5744.38
32	(17,25,17)	5053.00	4879.65	4880.90	4875.65	4875.65	4870.85	4876.60	4875.55	4872.35
33	(17,25,17)	6315.00	6113.80	6112.65	6109.90	6109.30	6106.80	6104.80	6104.45	6106.20
34	(18,25,18)	4814.00	4174.55	4168.10	4167.10	4169.25	4169.85	4172.10	4166.50	4165.40
35	(18,25,18)	4851.00	4543.05	4540.65	4537.95	4535.70	4539.40	4536.20	4533.90	4534.85
36	(18,25,18)	4949.00	4647.60	4647.05	4636.75	4642.40	4640.30	4647.25	4639.15	4641.20
37	(18,25,20)	5564.00	5048.85	5048.20	5059.05	5064.95	5069.05	5060.40	5057.85	5053.60
38	(20,27,20)	7989.00	6860.00	6856.40	6852.20	6853.40	6856.55	6853.55	6855.65	6850.00
39	(20,27,20)	8495.00	7513.30	7514.05	7511.20	7509.80	7509.80	7501.90	7509.80	7506.20
40	(20,30,20)	4328.00	3714.90	3716.95	3712.20	3713.40	3712.55	3709.90	3708.60	3708.40
41	(20,30,20)	6823.00	6705.30	6698.40	6694.65	6695.50	6694.15	6691.75	6692.90	6693.05
42	(20,30,20)	5597.00	5386.90	5382.45	5373.55	5375.90	5379.05	5372.80	5372.90	5371.65
Average		5431.27	5076.31	5075.55	5071.33	5071.71	5071.70	5070.52	5070.83	5070.22

between two adjacent dock doors is set 1s. Other parameters are generated as follows: DE_i, $DL_i \sim U(3, 10)$, $t_k \sim U(0, 20)$ and $f_{idk} \sim U(0, 20)$. All proposed algorithms are executed in MATLAB v 7.13 on an Intel Corei7, 2.2 GHz computer with 4 GB memory. Also, GAMS commercial software with solver Cplex is used to find optimal solutions.

The size of test problems is defined according to the number of compound trucks, destinations, and dock doors. In small size class, the number of destinations, compound trucks, and dock doors are less than 10, 7 and 7, respectively. The size of each test problem in computational results are represented as a tuple (I, D, M), where I, D, and M show the number of compound vehicles, of destinations, and dock doors, respectively.

The proposed algorithms are compared with exact solutions obtained from GAMS software in small sizes. The solutions gained by algorithms are compared with optimal solutions (if optimality is proven within 10,800s) or best-found solutions (due to the incapability of GAMS for solving test problems in large-scale problems, the best solution obtained through all algorithms is considered optimal). In addition, because metaheuristic algorithms use a random procedure to select solutions and moves, there are 20 replications for each small or large scale problem instance.

In small-scale problem, we compare 9 algorithms with exact solution obtained through GAMS software and present the outcome of all these algorithms for large-scale problems. These algorithms comprises as:

- Heuristic algorithm (H)
- Simple simulated annealing without learning and without tailor-made neighborhood structures (SA).
- Simple simulated annealing without learning and with tailor-made neighborhood structures (SA-NS).
- Simulated annealing with the learning approach "*Incremental Implementation*" with tailor-made neighborhood structures (SA-RL1).
- Simulated annealing with the learning approach "*Non-Stationary Problem*" with tailor-made neighborhood structures (SA-RL2).
- Simulated annealing with the learning approach "*Upper-Confdence-Bound Action Selection - Type 1*" with tailor-made neighborhood structures (SA-RL3).
- Simulated annealing with the learning approach "*Upper-Confdence-Bound Action Selection - Type 2*" with tailor-made neighborhood structures (SA-RL4).
- Simulated annealing with "*Q-learning*" approach with tailor-made neighborhood structures (SA-RL5) where $(\nu_1, \nu_2, \nu_3, \nu_4) = (5, 10, 15, 20)$.
- Simulated annealing with "*Q-learning*" approach with tailor-made neighborhood structures where (SA-RL6) $(\nu_1, \nu_2, \nu_3, \nu_4) = (10, 20, 50, 100)$.

Table 5
Comparison of all algorithms according to computational time in large-scale problems.

		CPU Time (s)		Best found solution (s)							
#	Size	H	Meta	SA	SA-NS	SA-RL1	SA-RL2	SA-RL3	SA-RL4	SA-RL5	SA-RL6
1	(8,10,8)	0.18	50.41	47.45	47.91	31.07	29.20	29.77	28.94	30.63	37.34
2	(8,10,8)	0.17	50.41	39.98	43.91	23.52	23.16	22.81	23.36	24.35	29.54
3	(8,10,8)	0.17	50.41	34.44	34.59	19.25	17.93	18.76	19.09	21.80	21.44
4	(8,12,9)	0.18	63.01	48.61	50.81	25.57	27.54	26.77	26.51	29.50	32.77
5	(8,12,9)	0.18	63.01	48.46	50.30	28.16	28.18	28.34	27.72	30.48	31.48
6	(8,12,9)	0.18	63.01	46.30	46.14	25.52	25.39	25.27	27.05	29.42	28.60
7	(10,13,10)	0.17	80.51	46.76	48.39	25.40	24.33	26.42	27.42	29.28	31.14
8	(10,13,10)	0.17	80.51	48.23	47.70	24.44	25.33	25.87	25.28	30.08	29.75
9	(10,13,10)	0.17	80.51	33.75	34.47	19.20	19.58	18.68	18.89	23.35	21.66
10	(10,15,11)	0.18	96.26	59.36	64.46	38.49	39.50	38.13	39.84	46.61	43.34
11	(10,15,11)	0.18	96.26	83.91	81.29	49.33	47.62	51.41	49.34	55.74	54.25
12	(10,15,11)	0.18	96.26	55.18	57.51	32.87	33.65	35.46	33.63	36.51	39.60
13	(11,15,12)	0.18	109.21	36.47	39.18	22.18	20.05	22.43	24.45	25.04	23.90
14	(11,15,12)	0.19	109.21	66.21	65.15	34.06	36.63	35.76	33.82	41.04	42.49
15	(11,15,12)	0.18	109.21	60.16	55.83	30.21	29.40	33.64	41.15	33.37	34.20
16	(12,16,12)	0.18	117.61	63.39	64.48	35.30	35.43	37.30	36.41	42.05	41.87
17	(12,16,12)	0.17	117.61	79.56	72.13	49.12	44.33	45.58	43.52	51.86	49.45
18	(12,16,12)	0.17	117.61	78.41	73.06	48.33	45.70	47.55	44.82	49.81	48.88
19	(12,18,14)	0.18	147.01	87.57	93.07	59.79	55.75	59.31	65.43	55.01	60.48
20	(12,18,14)	0.18	147.01	72.19	77.93	45.38	46.59	43.09	41.97	45.34	47.14
21	(12,18,14)	0.18	147.01	78.04	73.12	42.22	42.09	43.78	47.70	52.79	44.08
22	(14,19,14)	0.17	161.71	67.40	74.40	51.19	42.36	43.54	45.29	47.79	52.06
23	(14,19,14)	0.17	161.71	86.44	88.16	49.49	62.11	65.97	59.43	68.04	58.66
24	(14,19,14)	0.17	161.71	83.89	83.22	69.16	59.92	52.82	63.74	66.50	53.74
25	(14,22,16)	0.18	201.61	100.70	107.35	62.23	57.53	68.75	65.26	69.40	65.24
26	(14,22,16)	0.18	201.61	104.72	112.06	85.02	80.26	78.37	79.57	84.28	66.67
27	(14,22,16)	0.18	201.61	106.49	99.68	63.31	66.35	62.09	75.90	66.53	73.05
28	(15,23,15)	0.18	199.51	109.56	117.45	91.17	75.58	73.79	82.62	85.99	80.26
29	(15,23,15)	0.18	199.51	113.54	125.41	73.02	74.48	88.25	82.95	84.65	70.21
30	(15,23,15)	0.18	199.51	97.04	90.59	55.39	52.62	69.70	62.16	69.47	66.05
31	(17,25,17)	0.18	249.91	97.74	105.80	69.00	64.16	59.78	66.74	81.42	69.99
32	(17,25,17)	0.18	249.91	182.38	167.27	120.33	141.25	119.19	143.39	120.94	134.17
33	(17,25,17)	0.18	249.91	167.79	155.23	122.07	128.53	114.93	156.76	145.96	131.00
34	(18,25,18)	0.18	270.91	120.55	119.67	94.96	92.10	83.33	94.26	98.68	112.07
35	(18,25,18)	0.18	270.91	156.43	146.15	107.82	116.05	124.35	113.38	123.85	131.98
36	(18,25,18)	0.18	270.91	154.41	153.62	114.98	114.71	103.12	110.39	119.13	112.39
37	(18,25,20)	0.18	301.01	99.37	92.53	66.15	67.23	64.24	68.05	83.44	66.54
38	(20,27,20)	0.18	329.01	121.82	123.41	89.02	83.78	85.71	101.78	107.81	90.97
39	(20,27,20)	0.20	329.01	127.77	121.33	97.23	106.26	85.07	87.63	96.30	110.58
40	(20,30,20)	0.18	350.01	172.43	162.89	124.36	123.84	130.56	132.80	159.09	153.44
41	(20,30,20)	0.18	350.01	239.28	203.44	176.21	182.72	165.86	178.55	215.03	151.85
42	(20,30,20)	0.19	350.01	219.33	239.04	229.81	194.58	179.09	189.94	225.18	177.65
Average	0.18	172.66	93.89	93.10	64.79	63.90	62.73	66.36	71.51	67.19	

Table 6
Influence of partial unloading and complete unloading in the objective function.

Demand density	Objective function				Improvement (%)			
	t = 4	t = 6	t = 8	t = 10	t = 4	t = 6	t = 8	t = 10
0.21	634	927	1219	1511	0	0	0	0
0.32	588	862	1137	1413	7.26	7.01	6.73	6.49
0.41	560	828	1095	1361	11.67	10.68	10.17	9.93
0.61	527	774	1020	1266	16.88	16.5	16.32	16.21
0.71	470	688	906	1124	25.87	25.78	25.68	25.61
0.87	316	463	611	759	50.16	50.05	49.88	49.77

Table 2 represents the performance of all algorithms on the basis of objective value in small-scale problems. According to Table 2, problem-specific neighborhood structures help the basic simulated annealing algorithm to find better solutions. Moreover, learning approaches enhance the performance of SA algorithm. Based on numerical results, there is not significant difference between the outcome of all learning-based SA algorithms. Table 3 also shows the performance of all algorithms in small-scale problems in terms of computational time.

In addition, based on Tables 2 and 3, heuristic algorithm provides good initial solutions in a very short computational time and both heuristic and meta-heuristic algorithms are less dependent up on the size of the problem and they can provide good solutions in acceptable computational time.

Similarly, all algorithms are compared in large-scale problem where commercial solvers are not able to solve the problems. As the results show (Table 4), all learning-based SA algorithms outperform basic SA with\without problem-specific neighborhood structures, where *Q-learning* based simulated annealing algorithms are slightly better than others. Similarly, all learning-based algorithms converge in shorter computational time. Because the developed truck scheduling problem is an operational one, it is vital to find good solutions shortly. Therefore, *Q-learning* based algorithm with $(\nu_1, \nu_2, \nu_3, \nu_4) = (10, 20, 50, 100)$ is suggested for this problem. Similar to small-scale problems, the heuristic algorithm can find good solutions in negligible time. It is worth mentioning that our algorithms, i.e. SA-RL5 and SA-RL6 are compared with commonly used algorithms in the literature namely Genetic Algorithm, Particle Swarm Optimization, and Differential Evolution algorithms as benchmarks and results are provided in Appendix A (see Table 5).

Table 7
Influence of DBPR and loading/unloading in the objective function with one product type.

Demand density	Objective function with our assumption				Objective function without our assumption				Improvement (%)			
	t = 4	t = 6	t = 8	t = 10	t = 4	t = 6	t = 8	t = 10	t = 4	t = 6	t = 8	t = 10
0.21	634	927	1219	1511	717	1052	1386	1720	11.6	11.9	12	12.2
0.32	588	862	1137	1413	712	1054	1396	1736	17.4	18.2	18.6	18.6
0.41	560	828	1095	1361	716	1057	1402	1746	21.8	21.7	21.9	22.1
0.61	527	774	1020	1266	708	1044	1380	1718	25.6	25.9	26.1	26.3
0.71	470	688	906	1124	701	1041	1377	1709	33.0	33.9	34.2	34.2
0.87	316	463	611	759	703	1045	1387	1723	55.0	55.7	55.9	55.9

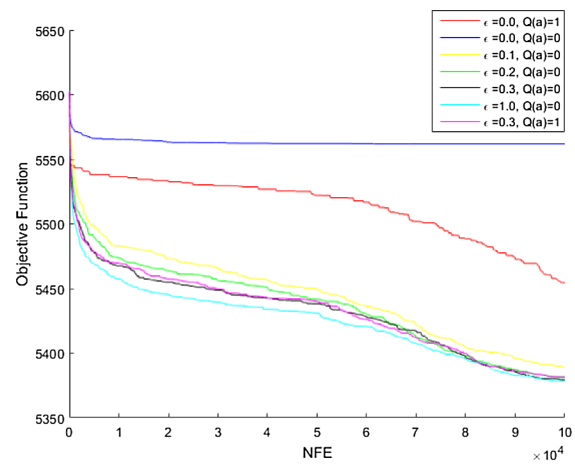

Fig. 6. Convergence of objective function of MAB problem with different settings.

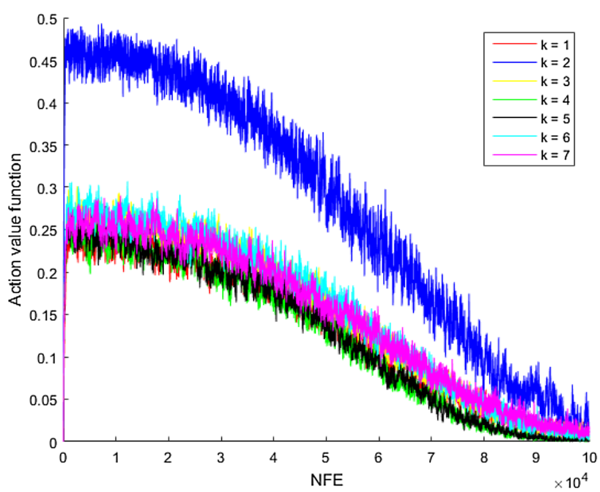

Fig. 7. Average value function over NFE in 50 runs.

5. Sensitivity analysis

In this section, we evaluate the impact of unloading/loading time and destination bound product ratio (DBPR) on the makespan. Also, the influence of partial unloading of compound trucks is investigated and compared with the situation in which all products should be unloaded. Destination bound product ratio is defined as the maximum proportion of demand of a destination which was initially loaded into a compound truck.

Actually, this factor shows whether the demand of each destination is distributed between compound trucks or not. It is possible that a compound truck brings the demand of many destinations into the cross-docking center and in this situation, it is expected that just a small proportion of its capacity is dedicated to each destination. In this case, if a compound truck is assigned to a destination, a little saving can be achieved through partial unloading (as the majority of the capacity of a compound truck is dedicated to unassigned destinations). It is worth mentioning that in this case, the value of this factor is small.

However, it would be possible that each compound truck brings the demand of a few destinations (for example, three or four destinations), i.e., a majority of its capacity is filled with the demand of a few destinations. In this case, a considerable proportion of capacity of compound trucks is dedicated to a single destination. In this situation, partial unloading will considerably decrease makespan. Generally, destination bound product ratio is a factor which is used to measure how demand for destinations are distributed in different trucks. Eq. (32) represents the value of this parameter:

$$DBPR_d = \max_{i \in I} \left(\frac{\sum_{k \in K} f_{idk} \cdot t_k}{\sum_{i \in I} \sum_{k \in K} f_{idk} \cdot t_k} \right) \qquad \forall\, d \in D \tag{32}$$

At first, we analyze these assumptions in a problem with 5 compound trucks, 7 destinations (or 2 outbound trucks), 5 doors and 1 product type. All test problems in sensitivity analysis are solved using GAMS software to find optimal solutions. Table 6 shows the impact of these parameters on the objective function. By increasing the destination bound product ratio, improvement in the objective function arises up to 50% in comparison with the situation that destination bound product ratio is diverse as much as possible. It can then be concluded that in a certain density by different loading/unloading time, the same improvement is obtained and destination bound product ratio has more influence than unloading/loading time on the improvement of objective function.

To investigate the influence of partial unloading on the objective function, our model is compared with the situation where all products should be unloaded and then reloaded (complete unloading). Table 7 shows that partial unloading can improve the objective function at least 11.6% when one product type is considered. As DBPR arises, the improvement obtained increases up to 55.9%. As results show, the improvement does not strongly depend on unloading/loading time.

5.1. Sensitivity analysis on learning component

One of the main concern in reinforcement learning is the exploration-exploitation trade-off. The agent wants to maximize its own reward and so take greedy decisions. On the other hand, it wants to explore other decisions to determine the average reward of other actions. In this regard, the agent takes the greedy action most of the time ($1-\epsilon$ percent of times) and take random ones in ϵ percent of times. In this situation, it can balance the exploration-exploitation trade-off. Therefore, it is of great importance to appropriately tune the value of ϵ.

Fig. 6 shows the convergence of objective function for different setting in a single test problem when the problem is replicated for 50 iterations and then average value is reported. The dark blue curve shows the situation when the agent takes greedy action all the time and the initial value of all action-value functions ($Q(a) = 0 \quad \forall\, a \in A$). In this setting, the algorithm traps in local optima in the first iterations of the algorithm which results in poor performance. On the other hand, the red curve presents the setting when the agent takes greedy action all the time while the initial value of all action-value functions is set to $Q(a) = 1 \quad \forall\, a \in A$. As none of neighborhood structures can improve the current solution all the time, the true value of action value functions is definitely less than 1. Therefore, when the agent chooses one of the NSs two scenario are possible: first, if the selected NS produces a new solution with lower objective function, then the value of action-value function remains the same. Second, if the selected NS produces a new solution with worse objective function, then the value of action-value function decreases. In this situation, at the beginning of algorithm all actions are selected for at least a few times. Therefore, the algorithm can find better solutions compared to the first setting. This technique for encouraging the algorithm for exploration is called *optimistic initial values* Sutton et al. (1998).

Fig. 7 represents the value of value functions over time in setting $\epsilon = 0.3$, $Q(a) = 0 \quad \forall\, a \in A$. At the beginning of the algorithm, the value of action-value functions increases and stays almost constant. Then, they decrease converge to zero. At the beginning of the algorithm, it is more likely to improve the current fathom solution, so, action-value functions increase or stays constant. However, when the algorithm continues, each NS can improve the current solution just a few times and the value of value functions decreases. These behavior suggests that the learning problem is non-stationary and the probability distribution of rewards changes over time. Because of non-stationary probability distribution of rewards, it can be inferred why $Q-learning$ based approached results in better solutions.

Remark: From Fig. 6, it is obvious that the *optimistic initial values* technique can help the algorithm to better explore the solution space. However, from Fig. 7, the problem is non-stationary and it is important to take random actions in all stages of the algorithm to find the value of each value function when its reward probability distribution changes. Fig. 6 justify why ϵ-greedy policy is better than the *optimistic initial values* technique.

6. Conclusion and future researches

This paper studies a truck-scheduling problem in a cross-docking center where compound trucks can also serve as outbound trucks. Moreover, compound trucks can be partially unloaded, so the time of unloading and reloading of some products is saved. An integrated model is proposed to find the optimal dock-door assignment, destination assignment, and truck scheduling to minimize makespan. Because of NP-Hardness of the model, a new heuristic, and several reinforcement learning-based simulated annealing algorithms are developed. Moreover, to better search the solution space, several generic and tailor-made neighborhood search structures are proposed. Moreover, because of unclear performance of each NS in different stages of optimization process, reinforcement learning approaches are used to learn appropriateness of each neighborhood search structure. Based on the performance of each NS, RL methods decide on which NS should be chosen to generate a new solution.

Sensitivity analysis shows that partial unloading has a crucial impact on the reduction of objective function, i.e. it can improve the objective function up to 56%. The results also show that if the demands are distributed among few compound trucks, partial unloading is more desirable. In other words, the improvement of objective function has a high dependency on destination-bound-product ratio while low dependency on unloading/loading time.

As directions for future research, it can be considered that the demand of destinations exceeds the capacity of trucks, so more than one truck should be assigned to each destination. Therefore, integrated truck scheduling and vehicle routing problems are more desirable. Moreover, this model can be studied in a resource-constrained cross-docking center. In addition, other objective functions such as minimization of total cost or combination of several objective functions can be used as criteria to assign trucks to destinations. This paper assumes that the number of compound trucks is at most the as same as that of dock door. Future research can study the case when the number of compound trucks is significantly larger than of dock doors. This problem will be more realistic if the model considers uncertainty in the problem. Besides, other reinforcement learning approaches such as SARSA, double *Q-learning* can be embedded in SA algorithm. Also, it is possible to take the advantage of the problem to defined different states in MDP problem. Finally, the proposed reinforcement learning-based simulated annealing can be applied to different problems.

Appendix A

A.1. Benchmarking

In this section, we compare the performance of our developed reinforcement learning-based simulated annealing algorithms with the conventional algorithms used in the literature. In truck scheduling problem in cross-docking center, several papers have used *Genetic Algorithm (GA)* (e.g. Arabani et al., 2011; Molavi et al., 2018; Mohtashami, 2015), *Particle Swarm Optimization (PSO)* (see e.g. Chen, Hsiao, Reddy, & Tiwari, 2016; Keshtzari et al., 2016; Wisittipanich & Hengmeechai, 2017), and *Differential Evolution (DE)* algorithms (e.g. Arabani et al., 2011; Assadi & Bagheri, 2016; Liao et al., 2013; Warren Liao, Egbelu, & Chang, 2012; Yazdani, Naderi, & Mousakhani, 2015). Therefore, we consider these algorithms as benchmark and evaluate the performances of *SA-RL5* and *SA-RL6*. Table A1 represents the performance of benchmark and our developed algorithms for small-scale problems. It is obvious that, in general, our developed algorithms outperform benchmarks.

Similarly, Table A2 shows the performance of benchmarks and our algorithms and out algorithms not only can find better solutions and lower objective value, but also they converge to their best solution faster than benchmarks. From Tables A1,A2, it can be concluded that our algorithms are better in terms of objective function and computational time and they can be recommended to solve the proposed model.

Table A1
Comparison of Benchmark Algorithms With Our Developed Algorithms in Small-Scale Problems.

#	Size	Objective Function					Best found solution (s)				
		DE	GA	SA-RL5	SA-RL6	PSO	DE	GA	SA-RL5	SA-RL6	
1	(2,3,2)	257.00	257.00	257.00	257.00	257.00	0.26	0.18	0.17	0.04	0.04
2	(2,3,2)	724.00	724.00	724.00	724.00	724.00	0.26	0.18	0.17	0.04	0.04
3	(2,3,2)	652.00	652.00	652.00	652.00	652.00	0.26	0.18	0.17	0.05	0.05
4	(2,4,3)	1368.00	1368.00	1368.00	1368.00	1368.00	0.28	0.19	0.18	0.06	0.06
5	(2,4,3)	1020.00	1020.00	1020.10	1020.00	1020.00	0.38	0.24	0.20	0.13	0.08
6	(2,4,3)	1644.00	1644.00	1644.10	1644.00	1644.00	0.44	0.27	0.21	0.14	0.08
7	(3,4,4)	1741.00	1741.00	1741.00	1715.00	1715.00	0.49	0.32	0.21	0.16	0.13
8	(3,4,4)	1577.80	1577.00	1577.00	1543.00	1543.00	0.68	0.30	0.23	0.37	0.38
9	(3,4,4)	1756.00	1756.00	1756.10	1721.00	1721.00	0.56	0.38	0.24	0.22	0.20
10	(3,5,3)	617.00	617.00	617.45	617.00	617.00	2.28	0.65	0.77	0.53	0.86
11	(3,5,3)	2009.00	2009.00	2012.60	2009.00	2009.00	1.31	1.14	0.63	0.51	0.53
12	(3,5,3)	1975.50	1973.00	1973.00	1973.00	1973.00	1.30	0.69	0.35	0.29	0.23
13	(3,6,4)	1890.00	1890.00	1891.20	1890.00	1890.00	2.68	1.48	1.19	2.58	2.04
14	(3,6,4)	2108.85	2107.00	2108.35	2107.25	2107.35	5.41	4.09	0.96	6.43	7.41
15	(3,6,4)	2753.70	2753.30	2756.30	2753.60	2753.65	6.64	4.91	0.94	6.87	6.56
16	(3,6,4)	2730.70	2726.15	2729.10	2726.05	2726.30	5.54	3.83	0.76	7.68	6.93
17	(3,6,4)	1528.00	1527.15	1528.25	1527.30	1527.60	3.26	4.84	0.69	6.86	6.89
18	(3,6,4)	914.70	910.00	913.90	910.10	910.50	5.37	2.14	0.69	7.24	8.11
19	(4,6,4)	1760.15	1760.00	1767.40	1760.15	1760.15	2.84	4.92	1.07	8.18	5.70
20	(4,6,4)	2961.85	2951.05	2961.10	2951.00	2951.20	3.81	3.77	0.78	8.34	7.82
21	(4,6,4)	1613.15	1609.00	1611.85	1609.10	1609.40	3.44	4.59	0.82	9.01	8.52
22	(4,7,4)	6366.05	6360.00	6364.85	6360.05	6360.45	6.71	8.03	1.48	10.70	11.15
23	(4,7,4)	2323.70	2321.55	2331.35	2320.80	2321.50	8.16	10.20	1.79	11.71	13.77
24	(4,7,4)	1452.95	1449.50	1452.15	1449.25	1449.55	9.57	6.19	1.77	11.74	10.16
25	(5,7,6)	4681.00	4681.00	4681.00	4681.00	4681.00	1.20	0.60	0.32	0.67	0.53
26	(5,7,6)	2621.05	2616.15	2627.05	2615.10	2616.80	8.46	15.50	3.65	14.17	15.14
27	(5,7,6)	2539.30	2536.00	2540.80	2536.00	2536.00	7.20	4.51	1.06	11.07	12.25
28	(5,7,5)	4125.95	4113.05	4142.65	4113.65	4113.00	6.13	10.30	1.45	10.94	11.18
29	(5,7,5)	1800.25	1775.45	1786.00	1775.25	1775.65	7.89	11.93	1.71	17.49	17.33
30	(5,7,5)	1655.85	1637.20	1647.60	1635.00	1635.50	3.89	14.01	1.98	17.76	19.33
31	(6,7,6)	3923.00	3923.00	3925.25	3923.00	3923.00	1.82	1.33	0.76	3.43	1.44
32	(6,7,6)	1684.05	1679.00	1688.75	1679.60	1679.35	6.58	9.62	1.63	14.44	15.73
33	(6,7,6)	1992.00	1992.00	1993.40	1992.00	1992.00	0.70	1.83	1.51	2.20	2.38
34	(5,8,5)	5163.00	5160.90	5170.45	5145.25	5149.55	11.62	14.41	2.97	20.39	20.18
35	(5,8,5)	2774.60	2769.05	2788.10	2762.65	2763.70	10.35	16.13	2.12	19.91	19.59
36	(5,8,5)	1678.90	1678.00	1684.00	1678.00	1678.00	6.52	7.57	2.86	14.04	13.66
37	(5,9,6)	6494.60	6478.05	6514.70	6474.05	6474.10	14.20	19.60	2.22	16.14	18.35
38	(5,9,6)	3015.60	3004.35	3026.95	2991.15	2989.90	17.41	22.06	3.99	22.27	24.75
39	(5,9,6)	1668.80	1653.30	1680.95	1634.40	1637.05	17.54	23.27	8.23	17.99	17.89
40	(6,9,6)	7182.40	7161.05	7178.30	7158.55	7158.50	16.72	20.75	5.80	21.89	23.59
41	(6,9,6)	3573.95	3562.40	3574.80	3556.10	3556.00	17.74	23.04	4.90	23.40	24.97
42	(6,9,6)	2015.40	2017.30	2029.35	2007.30	2008.20	18.13	24.07	4.79	24.11	25.92
Average		2436.54	2431.90	2439.01	2427.75	2428.05	5.86	7.24	1.63	8.86	9.09

Table A2
Comparison of Benchmark Algorithms With Our Developed Algorithms in Large-Scale Problems.

#	Size	Objective Function					Best found solution (s)				
		DE	GA	SA-RL5	SA-RL6	PSO	DE	GA	SA-RL5	SA-RL6	
1	(8,10,8)	3266.80	3261.45	3275.68	3253.98	3253.45	19.88	38.53	5.29	30.63	37.34
2	(8,10,8)	5398.58	5381.28	5409.20	5376.85	5377.50	19.78	31.66	5.46	24.35	29.54
3	(8,10,8)	4957.45	4933.98	4973.78	4933.70	4934.28	27.31	33.61	5.17	21.80	21.44
4	(8,12,9)	6170.70	6153.45	6160.50	6139.60	6139.30	39.81	46.99	16.29	29.50	32.77
5	(8,12,9)	6395.00	6377.30	6385.35	6325.13	6320.00	39.56	43.08	21.37	30.48	31.48
6	(8,12,9)	6139.60	6115.65	6135.20	6096.50	6097.55	34.41	44.02	18.87	29.42	28.60
7	(10,13,10)	2740.83	2736.83	2747.40	2732.25	2729.75	35.89	54.76	26.40	29.28	31.14
8	(10,13,10)	5668.95	5645.33	5659.38	5612.08	5618.05	58.34	58.23	23.21	30.08	29.75
9	(10,13,10)	6108.35	6068.40	6101.40	6062.00	6058.95	40.25	56.84	23.72	23.35	21.66
10	(10,15,11)	3264.75	3249.40	3241.90	3191.40	3191.75	59.27	72.05	58.41	46.61	43.34
11	(10,15,11)	5764.35	5752.25	5759.25	5700.05	5702.10	66.30	73.06	39.92	55.74	54.25
12	(10,15,11)	3809.95	3803.15	3810.20	3783.95	3781.95	46.84	68.18	39.34	36.51	39.60
13	(11,15,12)	4891.95	4862.05	4878.18	4853.50	4853.50	47.83	67.92	19.96	25.04	23.90
14	(11,15,12)	4984.85	4956.15	4961.40	4891.90	4890.90	66.07	78.63	42.56	41.04	42.49
15	(11,15,12)	5231.20	5163.05	5192.60	5144.45	5135.95	45.62	78.77	33.11	33.37	34.20
16	(12,16,12)	3588.10	3565.85	3585.10	3540.15	3534.60	75.00	86.30	42.42	42.05	41.87
17	(12,16,12)	4934.75	4921.25	4910.85	4851.25	4851.55	73.27	88.91	30.65	51.86	49.45

(*continued on next page*)

Table A2 (*continued*)

#	Size	Objective Function					Best found solution (s)				
		DE	GA	SA-RL5	SA-RL6	PSO	DE	GA	SA-RL5	SA-RL6	
18	(12,16,12)	3616.95	3610.85	3610.60	3568.45	3564.25	75.40	85.59	39.15	49.81	48.88
19	(12,18,14)	3897.05	3844.00	3792.05	3651.95	3649.00	95.27	116.26	75.83	55.01	60.48
20	(12,18,14)	3153.40	3134.15	3119.75	3050.20	3053.45	106.02	112.86	76.42	45.34	47.14
21	(12,18,14)	4253.55	4231.25	4214.45	4134.50	4134.05	95.51	120.79	52.94	52.79	44.08
22	(14,19,14)	5590.55	5587.90	5503.10	5372.45	5364.90	98.63	119.03	63.13	47.79	52.06
23	(14,19,14)	5086.30	5062.10	5045.45	4972.50	4972.60	102.57	118.56	68.74	68.04	58.66
24	(14,19,14)	5355.60	5337.65	5319.85	5223.65	5223.50	97.21	115.08	72.68	66.50	53.74
25	(14,22,16)	5842.25	5791.23	5738.38	5640.70	5641.05	137.23	165.44	109.73	69.40	65.24
26	(14,22,16)	5782.40	5747.15	5678.35	5580.25	5577.60	154.27	151.95	94.19	84.28	66.67
27	(14,22,16)	5532.30	5495.35	5429.60	5314.40	5315.20	127.47	161.49	106.89	66.53	73.05
28	(15,23,15)	7427.55	7328.80	7147.55	6987.95	6994.10	141.52	153.13	121.71	85.99	80.26
29	(15,23,15)	6149.45	6121.05	6061.85	5977.15	5976.90	148.95	148.52	94.30	84.65	70.21
30	(15,23,15)	5904.05	5859.85	5831.00	5765.70	5764.10	123.55	156.20	88.56	69.47	66.05
31	(17,25,17)	5988.63	5960.45	5867.30	5728.88	5744.38	168.76	190.90	126.48	81.42	69.99
32	(17,25,17)	5045.55	5024.55	4971.15	4875.55	4872.35	163.72	185.29	150.10	120.94	134.17
33	(17,25,17)	6306.95	6293.80	6212.45	6104.45	6106.20	181.96	170.85	169.92	145.96	131.00
34	(18,25,18)	4362.65	4325.50	4264.15	4166.50	4165.40	168.16	196.58	148.31	98.68	112.07
35	(18,25,18)	4687.20	4703.30	4647.30	4533.90	4534.85	177.68	202.56	136.79	123.85	131.98
36	(18,25,18)	4803.40	4820.80	4747.55	4639.15	4641.20	184.56	191.08	166.52	119.13	112.39
37	(18,25,20)	5373.10	5272.10	5221.55	5057.85	5053.60	225.39	217.16	168.59	83.44	66.54
38	(20,27,20)	7184.25	7125.45	7009.40	6855.65	6850.00	231.46	243.98	198.44	107.81	90.97
39	(20,27,20)	7774.20	7751.80	7679.55	7509.80	7506.20	203.89	241.48	159.82	96.30	110.58
40	(20,30,20)	4048.50	4039.70	3916.75	3708.60	3708.40	252.54	192.02	190.81	159.09	153.44
41	(20,30,20)	6897.25	6884.50	6819.85	6692.90	6693.05	258.04	281.35	181.40	215.03	151.85
42	(20,30,20)	5600.70	5584.30	5500.85	5372.90	5371.65	271.73	251.83	169.23	225.18	177.65
	Average	5204.37	5178.05	5147.23	5063.46	5062.86	113.97	126.46	82.93	71.51	67.19

References

Ahmadi, E., Goldengorin, B., Süer, G. A., & Mosadegh, H. (2018). A hybrid method of 2-tsp and novel learning-based ga for job sequencing and tool switching problem. *Applied Soft Computing, 65*, 214–229.

Alpan, G., Ladier, A.-L., Larbi, R., & Penz, B. (2011). Heuristic solutions for transshipment problems in a multiple door cross docking warehouse. *Computers & Industrial Engineering, 61*(2), 402–408.

Amini, A., & Tavakkoli-Moghaddam, R. (2016). A bi-objective truck scheduling problem in a cross-docking center with probability of breakdown for trucks. *Computers & Industrial Engineering, 96*, 180–191.

Assadi, M. T., & Bagheri, M. (2016). Differential evolution and population-based simulated annealing for truck scheduling problem in multiple door cross-docking systems. *Computers & Industrial Engineering, 96*, 149–161.

Berghman, L., Briand, C., Leus, R., & Lopez, P. (2015). The truck scheduling problem at crossdocking terminals-exclusive versus mixed mode. In *4th International Conference on Operations Research and Enterprise Systems (ICORES 2015)* (pp. pp–247).

Bodnar, P., de Koster, R., & Azadeh, K. (2015). Scheduling trucks in a cross-dock with mixed service mode dock doors. *Transportation Science, 51*(1), 112–131.

Boloori Arabani, A. R., Fatemi Ghomi, S. M. T., & Zandieh, M. (2010). A multi-criteria cross-docking scheduling with just-in-time approach. *The International Journal of Advanced Manufacturing Technology, 49*(5–8), 741–756.

Boloori Arabani, A. R., Fatemi Ghomi, S. M. T., & Zandieh, M. (2011). Meta-heuristics implementation for scheduling of trucks in a cross-docking system with temporary storage. *Expert Systems with Applications, 38*(3), 1964–1979.

Boloori Arabani, A., Zandieh, M., & Fatemi Ghomi, S. M. T. (2011). Multi-objective genetic-based algorithms for a cross-docking scheduling problem. *Applied Soft Computing, 11*(8), 4954–4970.

Boloori Arabani, A., Zandieh, M., & Fatemi Ghomi, S. M. T. (2012). A cross-docking scheduling problem with sub-population multi-objective algorithms. *The International Journal of Advanced Manufacturing Technology, 58*(5–8), 741–761.

Boysen, N. (2010). Truck scheduling at zero-inventory cross docking terminals. *Computers & Operations Research, 37*(1), 32–41.

Boysen, N., & Fliedner, M. (2010). Cross dock scheduling: Classification, literature review and research agenda. *Omega, 38*(6), 413–422.

Buijs, P., Vis, I. F. A., & Carlo, H. J. (2014). Synchronization in cross-docking networks: A research classification and framework. *European Journal of Operational Research, 239*(3), 593–608.

Chen, M.-C., Hsiao, Y.-H., Reddy, R. H., & Tiwari, M. K. (2016). The self-learning particle swarm optimization approach for routing pickup and delivery of multiple products with material handling in multiple cross-docks. *Transportation Research Part E: Logistics and Transportation Review, 91*, 208–226.

Chen, F., & Lee, C.-Y. (2009). Minimizing the makespan in a two-machine cross-docking flow shop problem. *European Journal of Operational Research, 193*(1), 59–72.

Chen, F., & Song, K. (2009). Minimizing makespan in two-stage hybrid cross docking scheduling problem. *Computers & Operations Research, 36*(6), 2066–2073.

Chiarello, A., Gaudioso, M., & Sammarra, M. (2018). Truck synchronization at single door cross-docking terminals. *OR Spectrum, 40*(2), 395–447.

Chmielewski, A., Naujoks, B., Janas, M., & Clausen, U. (2009). Optimizing the door assignment in ltl-terminals. *Transportation Science, 43*(2), 198–210.

Dulebenets, M. A. (2019). A delayed start parallel evolutionary algorithm for just-in-time truck scheduling at a cross-docking facility. *International Journal of Production Economics, 212*, 236–258.

Ertek, G. (2012). Crossdocking insights from a third party logistics firm in turkey. *Managing supply chains on the Silk Road: Strategy, performance, and risk, C. Haksöz, S. Seshadri, and AV Iyer, Eds*, 202.

Fazel Zarandi, M. H., Khorshidian, H., & Akbarpour Shirazi, M. (2016). A constraint programming model for the scheduling of jit cross-docking systems with preemption. *Journal of Intelligent Manufacturing, 27*(2), 297–313.

Hermel, D., Hasheminia, H., Adler, N., & Fry, M. J. (2016). A solution framework for the multi-mode resource-constrained cross-dock scheduling problem. *Omega, 59*, 157–170.

Joo, C. M., & Kim, B. S. (2013). Scheduling compound trucks in multi-door cross-docking terminals. *The International Journal of Advanced Manufacturing Technology, 64*(5-8), 977–988.

Keshtzari, M., Naderi, B., & Mehdizadeh, E. (2016). An improved mathematical model and a hybrid metaheuristic for truck scheduling in cross-dock problems. *Computers & Industrial Engineering, 91*, 197–204.

Konur, D., & Golias, M. M. (2013). Cost-stable truck scheduling at a cross-dock facility with unknown truck arrivals: A meta-heuristic approach. *Transportation Research Part E: Logistics and Transportation Review, 49*(1), 71–91.

Korukoğlu, S., & Ballı, S. (2011). A improved vogel's approximatio method for the transportation problem. *Mathematical and Computational Applications, 16*(2), 370–381.

Kuo, Y. (2013). Optimizing truck sequencing and truck dock assignment in a cross docking system. *Expert Systems with Applications, 40*(14), 5532–5541.

Ladier, A.-L., & Alpan, G. (2016). Cross-docking operations: Current research versus industry practice. *Omega, 62*, 145–162.

Ladier, A.-L., & Alpan, G. (2018). Crossdock truck scheduling with time windows: Earliness, tardiness and storage policies. *Journal of Intelligent Manufacturing, 29*(3), 569–583.

Larbi, R., Alpan, G., Baptiste, P., & Penz, B. (2011). Scheduling cross docking operations under full, partial and no information on inbound arrivals. *Computers & Operations Research, 38*(6), 889–900.

Larbi, R., Alpan, G., & Penz, B. (2009). Scheduling transshipment operations in a multiple inbound and outbound door crossdock. *2009 international conference on computers & industrial engineering* (pp. 227–232). IEEE.

Lee, K., Kim, B. S., & Joo, C. M. (2012). Genetic algorithms for door-assigning and sequencing of trucks at distribution centers for the improvement of operational performance. *Expert Systems with Applications, 39*(17), 12975–12983.

Liao, T. W., Egbelu, P. J., & Chang, P.-C. (2013). 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*(1), 212–229.

Li, Y., Lim, A., & Rodrigues, B. (2004). Crossdocking—jit scheduling with time windows. *Journal of the Operational Research Society, 55*(12), 1342–1351.

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

Molavi, D., Shahmardan, A., & Sajadieh, M. S. (2018). Truck scheduling in a cross docking systems with fixed due dates and shipment sorting. *Computers & Industrial Engineering, 117*, 29–40.

Morais, V. W. C., Mateus, G. R., & Noronha, T. F. (2014). Iterated local search heuristics for the vehicle routing problem with cross-docking. *Expert Systems with Applications, 41*(16), 7495–7506.

Rahmanzadeh Tootkaleh, S., Fatemi Ghomi, S. M. T., & Sheikh Sajadieh, M. (2016). Cross dock scheduling with fixed outbound trucks departure times under substitution condition. *Computers & Industrial Engineering, 92*, 50–56.

Rijal, A., Bijvank, M., & de Koster, R. (2019). Integrated scheduling and assignment of trucks at unit-load cross-dock terminals with mixed service mode dock doors. *European Journal of Operational Research*.

Serrano, C., Delorme, X., & Dolgui, A. (2017). Scheduling of truck arrivals, truck departures and shop-floor operation in a cross-dock platform, based on trucks loading plans. *International Journal of Production Economics, 194*, 102–112.

Shakeri, M., Low, M. Y. H., Turner, S. J., & Lee, E. W. (2012). A robust two-phase heuristic algorithm for the truck scheduling problem in a resource-constrained crossdock. *Computers & Operations Research, 39*(11), 2564–2577.

Sutton, R. S., & Barto, A. G. (1998). *Reinforcement learning: An introduction.* MIT Press.

Tarantilis, C. D. (2013). Adaptive multi-restart tabu search algorithm for the vehicle routing problem with cross-docking. *Optimization Letters, 7*(7), 1583–1596.

Terreri, A. (2001). Profiting from cross docking. *Warehousing Management, 8*(8), 29–32.

Vahdani, B. (2019). Assignment and scheduling trucks in cross-docking system with energy consumption consideration and trucks queuing. *Journal of Cleaner Production, 213*, 21–41.

Vahdani, B., & Zandieh, M. (2010). Scheduling trucks in cross-docking systems: Robust meta-heuristics. *Computers & Industrial Engineering, 58*(1), 12–24.

Van Belle, J., Valckenaers, P., Berghe, G. V., & Cattrysse, D. (2013). A tabu search approach to the truck scheduling problem with multiple docks and time windows. *Computers & Industrial Engineering, 66*(4), 818–826.

Van Belle, J., Valckenaers, P., & Cattrysse, D. (2012). Cross-docking: State of the art. *Omega, 40*(6), 827–846.

Warren Liao, T., Egbelu, P. J., & Chang, P.-c. (2012). Two hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross docking operations. *Applied Soft Computing, 12*(11), 3683–3697.

Wisittipanich, W., & Hengmeechai, P. (2017). Truck scheduling in multi-door cross docking terminal by modified particle swarm optimization. *Computers & Industrial Engineering, 113*, 793–802.

Wooyeon, Y., & Egbelu, P. J. (2008). Scheduling of inbound and outbound trucks in cross docking systems with temporary storage. *European Journal of Operational Research, 184*(1), 377–396.

Yazdani, M., Naderi, B., & Mousakhani, M. (2015). A model and metaheuristic for truck scheduling in multi-door cross-dock problems. *Intelligent Automation & Soft Computing, 21*(4), 633–644.

Ye, Y., Li, J., Li, K., & Hui, F. (2018). Cross-docking truck scheduling with product unloading/loading constraints based on an improved particle swarm optimisation algorithm. *International Journal of Production Research, 56*(16), 5365–5385.