Computers & Industrial Engineering 156 (2021) 107240

Contents lists available at ScienceDirect

Computers & Industrial Engineering

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

Truck scheduling optimization at a cold-chain cross-docking terminal with product perishability considerations

Oluwatosin Theophilus [a], Maxim A. Dulebenets [b,*], Junayed Pasha [a], Yui-yip Lau [c], Amir M. Fathollahi-Fard [d], Arash Mazaheri [e]

[a] *Department of Civil & Environmental Engineering, Florida A&M University-Florida State University (FAMU-FSU) College of Engineering, 2525 Pottsdamer Street, Building B, Suite B339, Tallahassee, FL 32310-6046, USA*

[b] *Department of Civil & Environmental Engineering, Florida A&M University-Florida State University (FAMU-FSU) College of Engineering, 2525 Pottsdamer Street, Building A, Suite A124, Tallahassee, FL 32310-6046, USA*

[c] *Division of Business and Hospitality Management, College of Professional and Continuing Education, The Hong Kong Polytechnic University, Hong Kong*

[d] *Department of Electrical Engineering, École de Technologie Supérieure, University of Quebec, 1100 Notre-Dame St. W., Montreal, Quebec, Canada*

[e] *Department of Building, Civil and Environmental Engineering, Concordia University, 1455 De Maisonneuve Blvd. W., Montreal, Quebec, Canada*

A R T I C L E I N F O

Keywords:
Cold supply chains
Cross-docking
Perishable products
Truck scheduling
Optimization
Metaheuristics

A B S T R A C T

An efficient product distribution is critical for proper supply chain operations. Many supply chains handle perishable products that decay over time. Due to mismanagement of supply chain operations, a significant portion of perishable products is wasted, resulting in substantial monetary losses. Cross-docking terminals (CDTs) have been widely used in cold supply chains for the product distribution but have not received adequate attention in the scientific literature. To improve the efficiency of perishable product distribution, this study introduces for the first time a novel mixed-integer mathematical formulation for the truck scheduling optimization at a cold-chain CDT. The model explicitly captures the decay of perishable products throughout the service of arriving trucks and accounts for the presence of temperature-controlled storage areas that are specifically designated for perishable products. The objective minimizes the total cost incurred during the truck service. Considering the complexity of the proposed model, a customized Evolutionary Algorithm is developed to solve it. The computational performance of the developed algorithm is assessed throughout the numerical experiments based on a detailed comparative analysis against the other metaheuristics. The developed Evolutionary Algorithm is found to be the most promising metaheuristic, considering both solution quality and CPU time perspectives. Furthermore, the proposed algorithm demonstrates an acceptable stability of the solution quality at termination. A set of additional sensitivity analyses are performed in order to draw some significant managerial implications, which would be of potential interest to the supply chain stakeholders that are involved in the distribution of perishable products in cold supply chains.

1. Introduction

Supply chain managers all over the world are constantly seeking for promising methods to improve the efficiency of product distribution and meet some important goals of supply chains, including the following: (1) cost minimization; (2) improvement of customer satisfaction; (3) efficient utilization of resources; (4) revenue/profit maximization; and (5) value creation (Dulebenets, 2018a; Felfel, Ayadi, & Masmoudi, 2016; Luo, Yang, & Wang, 2019; Nogueira, Coutinho, Ribeiro, & Ravetti, 2020). The various procedures used by supply chain stakeholders across the world have attracted the interest of researchers over the past two decades, aiming to accurately model these procedures and, ultimately, enhance the supply chain efficiency. This is primarily due to an increase in the volume of freight across various supply chains, alongside the complexity involved in the distribution of products (Dulebenets, 2018b; Ladier & Alpan, 2016). The supply chain stakeholders are faced with numerous challenges and tasks on a regular basis that have to be successfully addressed to achieve certain common objectives. These challenges include but are not limited to: (1) fierce competition in the industry; (2) operating cost reduction; (3) management of product

* Corresponding author.
E-mail addresses: opt16@my.fsu.edu (O. Theophilus), mdulebenets@eng.famu.fsu.edu (M.A. Dulebenets), jp17j@my.fsu.edu (J. Pasha), yuiyip.lau@cpce-polyu.edu.hk (Y.-y. Lau), amirmohammad.fathollahifard.1@ens.etsmtl.ca (A.M. Fathollahi-Fard), arashmaz@gmail.com (A. Mazaheri).

https://doi.org/10.1016/j.cie.2021.107240

Available online 16 March 2021
0360-8352/

perishability (in case of supply chains with perishable products); (4) maintenance of supplier and customer relationships; (5) management of the associated uncertainties; and (6) customer empowerment (Dulebenets & Ozguven, 2017; Ladier & Alpan, 2016; Margolis, Sullivan, Mason, & Magagnotti, 2018; Sreedevi & Saranga, 2017).

Improvement in the efficiency of product distribution in a supply chain invariably leads to a reduction in the operational cost, increased revenue, and generally lower price of products for the end customers. The cross-docking strategy has been relied on by many stakeholders in supply chains to facilitate the product distribution (Ladier & Alpan, 2016). A typical cross-docking terminal (CDT) has the following features and components (see Fig. 1): (1) a set of inbound doors where inbound trucks are served; (2) a set of outbound doors where outbound trucks are served; (3) a set of sorting/storage areas where products are deconsolidated, sorted, and consolidated (note that some storage areas can be temperature-controlled and designated specifically for perishable products); and (4) handling equipment, such as forklifts, conveyor belts, or combination of them in some cases. The cross-docking strategy can be described as follows. The incoming products are delivered by inbound trucks to the CDT from suppliers and manufacturers. The inbound trucks are unloaded by handling equipment after their assignment to the available inbound doors. The unloaded products are deconsolidated first, then sorted, and finally consolidated again in the designated storage areas. The products that were consolidated are loaded by handling equipment onto the outbound trucks, which deliver these products to the end customers (Ladier & Alpan, 2016).

Due to its proven effectiveness, the cross-docking strategy has been heavily deployed by the world largest retailers, such as Walmart, COSTCO, Target, and Office Depot. Walmart was the first retailer that started using CDTs. Furthermore, shipping companies, such as Federal Express (FedEx)[1], Express Mail Service (EMS), and United Parcel Service (UPS), have been using the cross-docking strategy in their supply chains for many years as well (Dulebenets, 2018a). The operations inside CDTs are usually planned and executed directly by the CDT operators (Ladier & Alpan, 2016). The CDT operators must address certain key decision problems to ensure the adequate functionality of CDTs, including the following: (1) the CDT shape determination; (2) determination of the required number of doors along with the door service mode; (3) the problem of truck scheduling (i.e., scheduling of arriving trucks for service); (4) allocation of the available handling equipment (deployment of forklifts or conveyor belts); and (5) storage area allocation. The CDT truck scheduling problem is considered as one of the main challenging decision problems that are faced by the CDT operators (Ladier & Alpan, 2016) and will be the main focus of the present study.

Although a significant number of previous studies addressed the CDT truck scheduling problem, cold-chain CDTs handling perishable products (i.e., the products that decay over time due to fluctuations in temperature, humidity, and pressure throughout the product distribution process) have not received adequate attention in the literature (Rahbari, Nasiri, Werner, Musavi, & Jolai, 2019). However, cold-chain CDTs have been widely used by different supply chain stakeholders for many years. For example, FedEx is heavily using cold-chain CDTs to handle healthcare products, specialty foods, flowers, seafood, and other perishable products (FedEx, 2016). Ineffective management of food supply chains, for example, can lead to a significant waste of food products. According to Mena, Terry, Williams, and Ellram (2014), approximately 20–30% of food products are wasted in supply chains. In the United States alone, more than 30% of perishable products, worth almost $50 billion, are thrown away every year (Environment, 2020). Such a significant waste of products occurs due to mismanagement of supply chain operations. To improve the efficiency of supply chains with perishable products, this study introduces for the first time a novel mixed-integer mathematical formulation for the truck scheduling optimization at a CDT that explicitly captures the decay of perishable products throughout the service of arriving trucks and accounts for the presence of temperature-controlled storage areas that are specifically designated for perishable products.

The objective function of the formulation presented minimizes the total cost incurred during the truck service that incorporates the total truck waiting cost, the total truck service cost, the total product inventory cost, the total truck delayed departure cost, as well as the total product decay cost. Considering the computational complexity of the model proposed, a customized Evolutionary Algorithm is developed to solve it. The computational performance of the developed algorithm is assessed throughout the numerical experiments based on a detailed comparative analysis against the other metaheuristics. Some significant managerial implications, which would be of potential interest to the supply chain stakeholders that are involved in the distribution of perishable products, are drawn as well. The contributions of this work to the CDT truck scheduling literature and the state-of-the-art can be outlined as follows:

- A novel mixed-integer mathematical formulation is proposed for the truck scheduling optimization at a cold-chain CDT;
- Unlike the previously conducted CDT truck scheduling studies, the proposed model explicitly captures the decay of perishable products using an exponential function throughout the service of arriving trucks;
- This study models the temperature-controlled storage areas that are designated specifically for perishable products and critical for proper operations of cold-chain CDTs;
- Considering the computational complexity of the proposed model, a novel customized metaheuristic is presented to solve the model;
- A detailed comparative analysis is conducted to assess the computational performance of the metaheuristic developed against the alternative exact and approximate optimization methods.

The remaining sections of this manuscript are further organized in the following order. Section 2 provides a concise review of the recent studies that are relevant to the problem of truck scheduling at CDTs. Section 3 provides a detailed description of the operations of the CDT to be modeled in this study. Section 4 presents the proposed mixed-integer mathematical formulation for the CDT truck scheduling problem with product perishability considerations. Section 5 contains a thorough description of the customized Evolutionary Algorithm that was developed as a part of this study to solve the mathematical model proposed. Section 6 evaluates the solution algorithm developed in terms of different performance metrics and provides some managerial implications. Section 7 concludes with the main findings of the present study and proposes some areas to be considered in the future research.

2. Literature review

Several previous studies have conducted a detailed review of the relevant efforts on the CDT operations. These studies aimed to analyze different CDT mathematical models, the effects of the CDT shape selection, and various CDT planning levels that ranged from strategic to tactical and operational (Agustina, Lee, & Piplani, 2010; Ladier & Alpan, 2016; Shuib & Fatthi, 2012; Theophilus, Dulebenets, Pasha, Abioye, & Kavoosi, 2019; Van Belle, Valckenaers, & Cattrysse, 2012). The focus of this section of the manuscript is to present a review of the most recent studies that are relevant to the CDT truck scheduling. The studies collected were further classified as the general CDT truck scheduling studies and the studies specifically focusing on the cold-chain CDT operations as well as the product perishability considerations.

[1] Note: the full list of abbreviations that were used in this manuscript is provided in Appendix A.

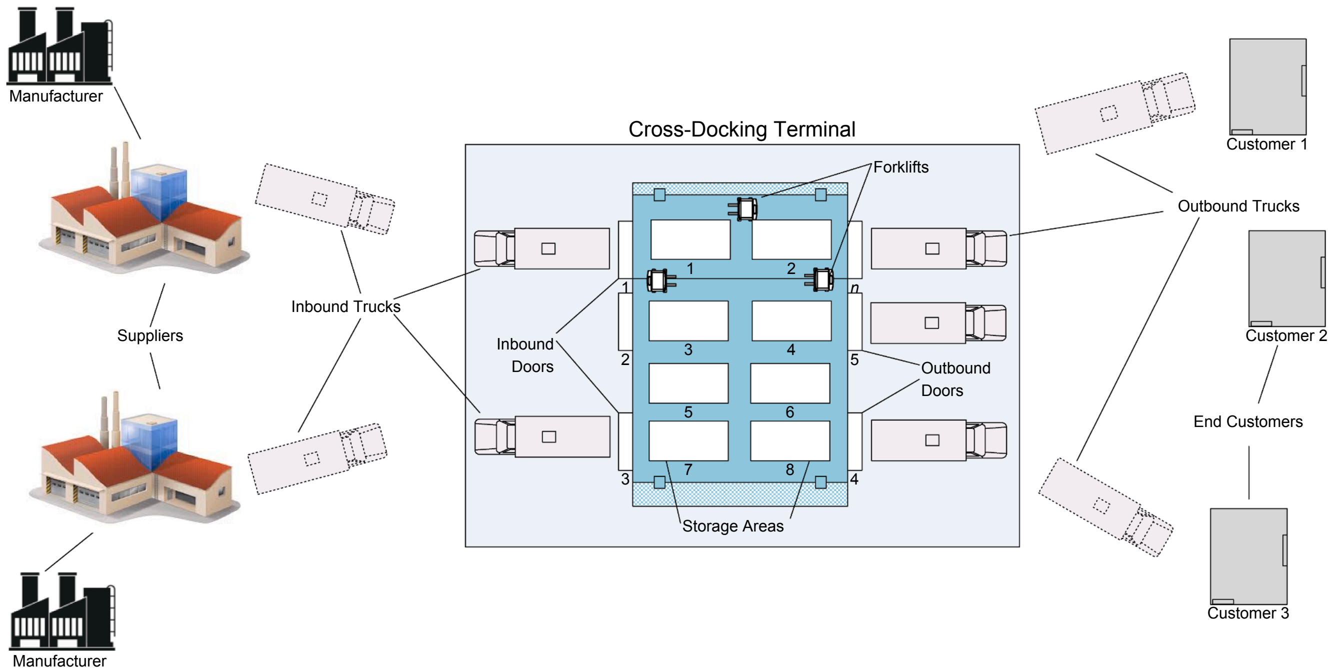

Fig. 1. A supply chain with a CDT.

2.1. Truck-scheduling at CDTs

Ahkamiraad and Wang (2018) studied the problem of capacitated vehicle routing with multiple CDTs, considering pickups, deliveries, and time windows. The objective function of the proposed model minimized the transportation and fixed costs. The problem was solved using a hybrid algorithm that was inspired by the Evolutionary Algorithm (EA) and Particle Swarm Optimization (PSO). Baniamerian, Bashiri, and Zabihi (2018) introduced the problem of vehicle routing and scheduling at a CDT where customer satisfaction was directly accounted for. A mixed-integer linear mathematical formulation was proposed for the problem, minimizing the transport cost and the cost of customer time window violation. The authors designed a two-phase EA to solve the model presented. The obtained results showed that the developed mathematical model achieved a customer satisfaction of at least 86.6%, unlike the classic model that achieved the maximum satisfaction level of 33.3%. Heidari, Zegordi, and Tavakkoli-Moghaddam (2018) formulated the truck scheduling problem at a CDT that incorporated uncertainty in truck arrival times. A bi-level optimization framework was used for the problem. Two metaheuristics, Non-Dominated Sorting Genetic Algorithm II (NSGA-II) along with Multi-Objective Differential Evolution (MODE), were deployed to solve the problem. The NSGA-II and MODE were compared against an alternative algorithm (called “GASH”). The conducted numerical experiments confirmed the superiority of MODE.

Molavi, Shahmardan, and Sajadieh (2018) examined the truck scheduling problem at a CDT with the scheduled departure times of outbound trucks as hard constraints. The authors developed a mixed-integer programming model, minimizing the delivery cost and the penalty of delayed shipments. The problem was solved using DE, EA, PSO, and hybrid metaheuristic. The computational experiments that were conducted as a part of that study showed that the hybrid metaheuristic was superior to the alternative algorithms. Nasiri, Rahbari, Werner, and Karimi (2018) presented a mixed-integer linear formulation, which incorporated the selection of suppliers and the allocation of orders into the vehicle routing problem with CDTs. The objective minimized the costs associated with purchasing, cross-docking, transportation, and early/tardy delivery. A two-stage solution algorithm was deployed to tackle large-scale instances. The experiments showed that the transportation and earliness/tardiness costs were the highest cost components. Abad, Vahdani, Sharifi, and Etebari (2019) proposed a multi-objective optimization model for the split pollution vehicle routing problem with a CDT and fuzzy probabilistic time window constraints. The first objective function focused on minimization of the total cost, whereas the second one focused on minimization of the total fuel consumption. The third objective maximized the supplier and customer satisfaction. Multi-Objective Grey Wolf Optimizer (MOGWO) and Multi-Objective Imperialist Competitive Algorithm (MOICA) were used to solve the presented mathematical model.

Corsten, Becker, and Salewski (2019) proposed a CDT optimization model that integrated the scheduling of trucks and workforce for a single working day. The objective was to minimize the cost of engaging temporary workers throughout the service of trucks. CPLEX was used as a solution method. The study considered some workforce coordination policies as well as daily working hour regulations. Fathollahi-Fard, Ranjbar-Bourani, Cheikhrouhou, and Hajiaghaei-Keshteli (2019) applied the Social Engineering Optimizer (SEO) approach to tackle the truck scheduling problem at a CDT. The SEO was inspired by social engineering theory, which involves attackers targeting defenders using certain important information. The results from the performed numerical experiments showed that the adopted approach delivered competitive results when compared with the existing algorithms. Khorshidian, Shirazi, and Ghomi (2019) presented a bi-objective model for the integrated transportation planning and truck scheduling problem in a CDT system. The objective functions minimized the total cost and the makespan. The study proposed a hybrid solution approach that was inspired by the augmented e-constraint method (which is generally referred to as “AUGMECON2”) and TOPSIS to solve the model. A real-life case study was presented to showcase the potential of the developed methodology.

Rijal, Bijvank, and de Koster (2019) studied the integrated problem of the CDT truck scheduling and door assignment, considering a mixed door service mode. The objective function of the presented mathematical model aimed to minimize the total cost that was incurred during the truck service. The problem was solved by an adaptive large neighborhood search algorithm that was developed as a part of that study. A set of extensive computational experiments demonstrated that simultaneous optimization of door assignment and truck scheduling reduced the average total cost by 12% as compared to the sequential approach.

Tadumadze, Boysen, Emde, and Weidinger (2019) focused on the integrated scheduling of workforce and trucks that were arriving in the considered planning horizon at a CDT. The objective function of the developed model focused on minimization of the total flow time of trucks. A number of heuristics were presented to tackle the model presented. The conducted numerical experiments revealed that the integrated scheduling of workforce and trucks could improve the CDT operations by reducing the truck flow times. Wisittipanich, Irohara, and Hengmeechai (2019) investigated the problem of truck scheduling in a network of CDTs. The objective was to simultaneously identify the schedule of all the inbound and outbound trucks in the network of CDTs, aiming to minimize the makespan. LINGO was used to solve the model. The numerical experiments demonstrated that optimizing the truck schedule at each CDT sequentially was less effective as compared to the integrated scheduling of trucks in the CDT network.

Ardakani, Fei, and Beldar (2020) considered the CDT truck scheduling problem, assuming that the pre-emption of inbound trucks was allowed (i.e., a given inbound truck might vacate the door for service of another truck and then return back to that door to complete its service). The objective function of the model that was proposed aimed to minimize the makespan. Exact and heuristic methods were deployed to tackle the problem. The conducted computational experiments clearly showed a competitive performance of the proposed heuristic algorithms. Shahmardan and Sajadieh (2020) addressed a special case of the truck scheduling problem at a CDT, where compound trucks were served. A compound truck could be used as an inbound truck and an outbound truck. The objective was to identify the best truck and destination assignment that minimized the makespan. The model was solved using a metaheuristic algorithm that was inspired by Simulated Annealing (SA). A set of extensive numerical experiments confirmed that partial unloading of compound trucks could be helpful in decreasing the makespan and improving the efficiency of the CDT operations.

2.2. Cold-chain CDT operations and product perishability considerations

Among the studies on CDT truck scheduling, only a few efforts were dedicated to modeling cold-chain CDT operations and product perishability. For example, Boysen (2010) addressed the truck scheduling problem at a "zero-inventory" CDT in a food supply chain. Based on a "zero-inventory" concept, the perishable products unloaded at the CDT were transferred directly to the refrigerated outbound trucks without any temporary storage. Different objectives were considered in the study, including the overall truck flow time minimization, the overall truck processing time minimization, and the overall truck tardiness minimization. Two methods were used to solve the proposed model, which included Dynamic Programming (DP) and SA. Agustina, Lee, and Piplani (2014) investigated the problem of truck scheduling and vehicle routing at a CDT for perishable product distribution. A mixed-integer programming formulation was presented for the problem, and the objective function minimized the total cost that included: (1) earliness cost; (2) tardiness cost; (3) storage cost; and (4) transportation cost. The concept of customer zoning was introduced to decrease the computational complexity of the presented model. The model was further solved with an exact optimization approach (i.e., CPLEX). The results from the computational experiments showed that the scheduling strategy proposed could aid the distribution of perishable products at a minimal cost and preserve the product quality.

Rahbari et al. (2019) studied the problem of vehicle routing and scheduling at a CDT with perishable products. A bi-objective mixed-integer linear formulation was presented, and the objective function minimized the total transportation cost and maximized the weighted freshness of transported products. A linear function was used to model the product decay over time. Furthermore, two robust optimization models were developed to capture uncertainty in product freshness and outbound truck travel time. A set of numerical experiments clearly showed that the developed methodology was able to drastically reduce the decay of perishable products without substantially affecting the total transportation cost. Fathollahi-Fard, Ahmadi, and Sajadieh (2020) proposed a Modified Red Deer Algorithm (MRDA) to tackle the problem of truck scheduling at a CDT with perishable products. A strict deadline was imposed for service completion of the outbound trucks that carried perishable products. The objective of the presented formulation minimized the makespan. The performance of the proposed algorithm was assessed using ten problem instances. The performed numerical experiments revealed that MRDA was superior to the other metaheuristics, including SA, EA, PSO, ICA, and RDA.

Moreover, there are a lot of studies that modeled perishability of products throughout different supply chain operations but did not explicitly capture the cross-docking operations. For example, Ahumada and Villalobos (2011) proposed an integrated tactical-level planning method for the production and distribution of perishable products. The objective maximized the total revenue. A linear function was adopted for modeling the product decay. The problem was solved to optimality using CPLEX. Bilgen and Çelebi (2013) developed a model for the integrated distribution planning and production scheduling in dairy supply chains. The objective aimed to maximize the total benefit, taking into account the main cost components of supply chain operations (e.g., processing, storage, setup, backlogging, overtime) and shelf life of perishable products. A hybrid solution approach, inspired by simulation and optimization, was presented to solve the problem. Grunow and Piramuthu (2013) modeled the use of the Radio-Frequency Identification (RFID) technology in highly perishable food supply chains. The product decay was emulated using an exponential function. Considering product expiry dates and remaining shelf life, certain conditions were developed under which RFID could benefit various supply chain stakeholders. Many other studies used exponential functions as well, aiming to accurately model the decay of perishable products due to different factors throughout the supply chain operations (Piramuthu & Zhou, 2013; Piramuthu, Farahani, & Grunow, 2013; Rong, Akkerman, & Grunow, 2011; Wang & Li, 2012; Yu & Nagurney, 2013).

2.3. Literature summary, existing gaps, and contributions of this study

The literature review conducted indicates that the number of studies on CDT truck scheduling increases every year. Different models have been proposed to improve the effectiveness of CDT operations and truck scheduling. However, most of the developed models cannot be applied in cold-chain CDTs, which handle perishable products that decay over time due to fluctuations in temperature, humidity, and pressure. Only Rahbari et al. (2019) explicitly modeled the decay of perishable products using a linear function throughout their distribution at the CDT and delivery to the end customers. On the other hand, there are a lot of studies that modeled perishability of products throughout different supply chain operations but did not explicitly capture the cross-docking operations. Many of these studies used an exponential function to accurately capture the decay of perishable products (Piramuthu & Zhou, 2013; Piramuthu et al., 2013; Rong et al., 2011; Wang & Li, 2012; Yu & Nagurney, 2013). In order to enhance the effectiveness of supply chains with perishable products, this study introduces for the first time a novel mixed-integer mathematical formulation for the truck scheduling optimization at a cold-chain CDT. Unlike the study by Rahbari et al. (2019), the proposed model explicitly captures the decay of perishable products using an exponential function throughout the service of arriving trucks. Moreover, this study models the temperature-controlled storage areas that are designated specifically for perishable products. Considering the computational complexity of the proposed model, a customized Evolutionary Algorithm is designed to tackle the model and demonstrate some significant managerial implications.

3. Problem description

This section of the manuscript describes in detail the operations of the cold-chain CDT modeled in this study. A description of the main notations that will be used throughout the problem description and mathematical model development is presented in Table 1. The shape of the CDT is assumed to be an I-shape. Several CDT truck scheduling studies reported that the CDTs with I-shape have been widely used in practice (Dulebenets, 2019a; Ladier & Alpan, 2016; Theophilus et al., 2019). However, the mathematical model to be presented in this study would be applicable to the CDTs of other shapes, such as E, H, L, T, U, X, etc. The geometric layout of the CDT is presented in Fig. 2. The rest of this section addresses the following aspects of the considered CDT truck scheduling problem: (1) truck arrivals; (2) CDT door assignment; (3) internal CDT operations; (4) modeling decay of perishable products; (5) objective of the CDT operator; and (6) an illustrative example of a truck service.

3.1. Truck arrivals

The CDT operations begin upon arrival of the assigned trucks. Let $T = \{1, \cdots, w^1\}$ be the set of trucks arriving for the service at the considered cold-chain CDT. The set of all the arriving trucks consists of the set of inbound trucks and the set of outbound trucks. Let $T^{in} = \{1, \cdots, w^2\}, T^{in} \subseteq T$ and $T^{out} = \{1, \cdots, w^3\}, T^{out} \subseteq T$ be the set of inbound trucks and the set of outbound trucks, respectively. Therefore, any given truck is either an inbound truck or an outbound truck: $T^{in} \cup T^{out} = T$ and $T^{in} \cap T^{out} = \emptyset$. The truck-to-door assignment is performed by the CDT operator. The inbound trucks may arrive either full or partially full. On the other hand, the outbound trucks may arrive either empty or partially empty. In this study, the arrival time is assumed to follow a predetermined arrival schedule agreed upon by both the CDT operator and the corresponding logistics company. Such an assumption has been commonly adopted by many of the previous CDT truck scheduling studies (Boloori Arabani, Zandieh, & Ghomi, 2012; Dulebenets, 2019a; Liao, Egbelu, & Chang, 2013; Miao, Lim, & Ma, 2009).

Table 1
The main notations used in this study.

Model Component		Description
Type	Notation	
Sets	$T = \{1, \cdots, w^1\}$	set of all the trucks arriving at the CDT
	$T^{in} = \{1, \cdots, w^2\}, T^{in} \subseteq T$	set of all the arriving inbound trucks
	$T^{out} = \{1, \cdots, w^3\}, T^{out} \subseteq T$	set of all the arriving outbound trucks
	$D = \{1, \ldots, w^4\}$	set of all the available CDT doors
	$D^{in} = \{1, \ldots, w^5\}, D^{in} \subseteq D$	set of all the available inbound doors
	$D^{out} = \{1, \ldots, w^6\}, D^{out} \subseteq D$	set of all the available outbound doors
	$O = \{1, \ldots, w^7\}$	set of all the available service orders
	$N = \{1, \ldots, w^8\}$	set of all the available temporary storage areas
	$P = \{1, \ldots, w^9\}$	set of all the product types to be handled at the CDT
Decision variables	$x_{tdo} \in \mathbb{B} \forall t \in T, d \in D, o \in O$	=1 if truck t is assigned to be served at CDT door d in the o^{th} service order (=0 otherwise)
Auxiliary variables	$TC \in \mathbb{R}^+$	the total cost to be incurred by the CDT operator throughout service of the arriving trucks (USD)
	$y_{tdo} \in \mathbb{R}^+ \forall t \in T, d \in D, o \in O$	idle time of CDT door d between service of truck t and its preceding truck that was served in the $(o-1)^{th}$ service order (hours)
	$\tau^{wt}_t \in \mathbb{R}^+ \forall t \in T$	waiting time for truck t (hours)
	$\tau^{st}_t \in \mathbb{R}^+ \forall t \in T$	service start time for truck t (hours)
	$\tau^{tst}_{tdo} \in \mathbb{R}^+ \forall t \in T, d \in D, o \in O$	total service time for truck t at CDT door d served in the o^{th} service order (hours)
	$\tau^{ft}_t \in \mathbb{R}^+ \forall t \in T$	service finish time for truck t (hours)
	$\tau^{dt}_t \in \mathbb{R}^+ \forall t \in T$	delayed departure time for truck t (hours)
	$\tau^{ts}_{\underline{t}\bar{t}p} \in \mathbb{R}^+ \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	temporary storage time for the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (hours)
	$\tau^{tht}_{\underline{t}\bar{t}p} \in \mathbb{R}^+ \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	total handling and transfer time for the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (hours)
	$Q^{\tau}_{\underline{t}\bar{t}p} \in \mathbb{R}^+ \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	quality of the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time τ (%)
	$\Delta Q_{\underline{t}\bar{t}p} \in \mathbb{R}^+ \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	change in quality of the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (%)
Parameters	$\tau^{at}_t \in \mathbb{R}^+ \forall t \in T$	arrival time of truck t at the CDT (hours)
	$\tau^{ht}_t \in \mathbb{R}^+ \forall t \in T$	handing time of truck t (hours)
	$\tau^{sd}_t \in \mathbb{R}^+ \forall t \in T$	scheduled departure time for truck t from the CDT (hours)
	$\tau^{tr}_{dn} \in \mathbb{R}^+ \forall d \in D, n \in N$	transfer time from CDT door d to storage area n (hours)
	$z_{tn} \in \mathbb{B} \forall t \in T, n \in N$	=1 if the products carried by truck t are assigned to temporary storage area n (=0 otherwise)
	$\varphi_{\underline{t}\bar{t}p} \in \mathbb{B} \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	=1 if inbound truck $\underline{t}$ carries the product of type p for outbound truck $\bar{t}$ (=0 otherwise)
	$q_{\underline{t}\bar{t}p} \in \mathbb{N} \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	quantity of the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (product units)
	$Q^0_{\underline{t}\bar{t}p} \in \mathbb{R}^+ \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	quality of the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time "0" (%)
	$\lambda_p \in \mathbb{R}^+ \forall p \in P$	decay rate of the product of type p (hour^{-1})
	$c^{wt}_t \in \mathbb{R}^+ \forall t \in T$	unit waiting cost for truck t (USD/hour)
	$c^{tst}_t \in \mathbb{R}^+ \forall t \in T$	unit service cost for truck t (USD/hour)
	$c^{ts}_p \in \mathbb{R}^+ \forall p \in P$	unit temporary storage cost for the product of type p (USD/hour)
	$c^{dt}_t \in \mathbb{R}^+ \forall t \in T$	unit delayed departure cost for truck t (USD/hour)
	$c^{dc}_p \in \mathbb{R}^+ \forall p \in P$	unit decay cost for the product of type p (USD/% decay)
	M	sufficiently large positive number

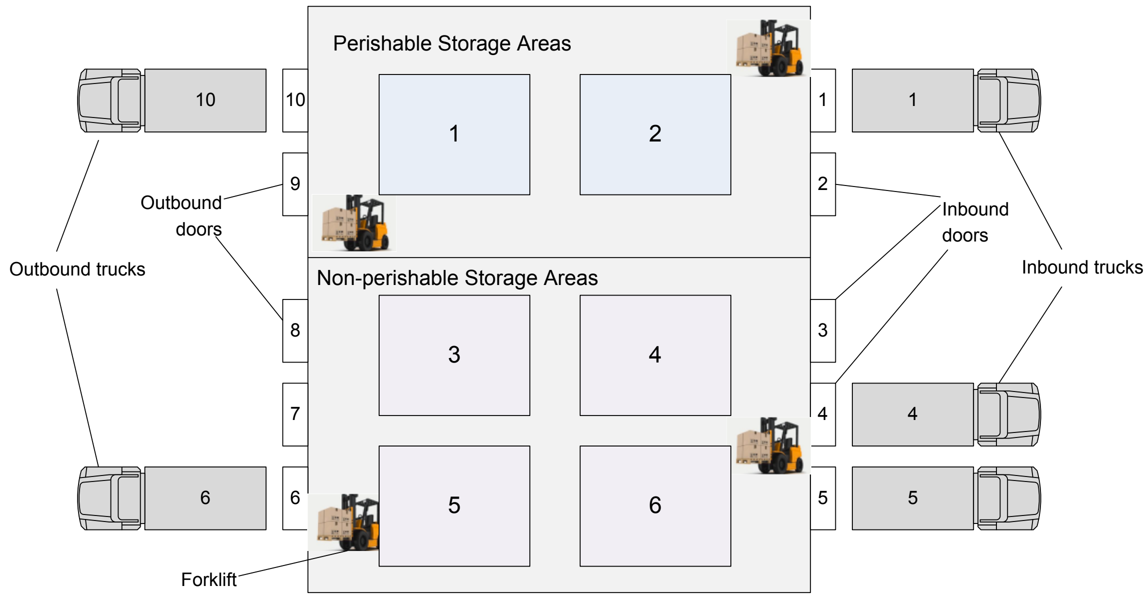

Fig. 2. The handling processes within a cold-chain CDT.

3.2. CDT door assignment

There are two door assignment modes in the CDT truck scheduling literature (Ladier & Alpan, 2016; Theophilus et al., 2019): (1) the mixed door service mode, based on which the same door can either serve inbound or outbound trucks (i.e., any arriving truck can be served at any available door); (2) the segregated door service mode, based on which a set of doors are dedicated to serve inbound trucks only (i.e., these doors are referred to as "inbound doors"), while the remaining doors are dedicated to serve outbound trucks only (i.e., these doors are referred to as "outbound doors"). The cold-chain CDT considered in this study is assumed to operate in a segregated door service mode (see Fig. 2). Let $D = \{1, ..., w^4\}$ be the set of the CDT doors available for service of the arriving trucks. Let $D^{in} = \{1, ..., w^5\}, D^{in} \subseteq D$ and $D^{out} = \{1, ..., w^6\}, D^{out} \subseteq D$ be the set of inbound doors and the set of outbound doors, respectively. Each arriving inbound truck has to be scheduled for service at one of the available inbound doors in one of the service orders. On the other hand, each arriving outbound truck has to be scheduled for service at one of the available outbound doors in one of the service orders. Let $O = \{1, ..., w^7\}$ be the set of all the available service orders.

3.3. Internal CDT operations

The canonical CDT operations include the following: (1) unloading; (2) product transfer; (3) deconsolidation; (4) sorting; (5) storage; (6) consolidation; and (7) loading. The unloading operation begins after an inbound truck is docked for service. On the contrary, the loading operation begins for an outbound truck after the corresponding inbound truck starts unloading the products or directly from the designated storage area (in case the assigned products have already been transferred to the designated storage area). Once any given truck (whether inbound or outbound) is docked for service, it will remain docked until the service is complete. Therefore, pre-emption is not allowed throughout the service of trucks. Usually, the products that are transported by the inbound trucks come in standard packaging units (for example, standard boxes or pallets). The time required to unload a particular inbound truck or load a particular outbound truck is referred to as "handling time" and is denoted as $\tau_t^{ht}, t \in T$ (hours). The forklift operators are used at the considered cold-chain CDT for the product handling and transfer.

The unloaded products are transferred to the temporary storage areas that are located between the inbound and outbound doors (see Fig. 2). Let $N = \{1, ..., w^8\}$ be the set of available temporary storage areas. In addition to the storage of products, each temporary storage area also serves as a point for deconsolidation, sorting, and consolidation of the products based on the customer preferences. Some products can be transferred directly from the inbound trucks to the corresponding outbound trucks without temporary storage if these trucks are already docked at the cold-chain CDT. The allocation of the available temporary storage areas depends on the product type. Let $P = \{1, ..., w^9\}$ be the set of products delivered by the inbound trucks. Unlike the previous CDT truck scheduling studies, this study categorizes the temporary storage areas of the considered CDT into two groups (see Fig. 2): (1) perishable storage areas – specifically allocated for the storage of perishable products; (2) non-perishable storage areas – specifically allocated for the storage of non-perishable products. The temperature-controlled perishable storage areas are critical for cold-chain CDTs, as in certain cases the arrival times of the inbound trucks and the corresponding outbound trucks may substantially vary, and temporary storage of perishable products would be unavoidable. Without temperature-controlled storage areas, the decay rate of perishable products would significantly increase.

The CDT operator generally allocates a sufficient capacity for the available storage areas based on the expected amount of trucks that will be arriving and the quantity of products to be delivered (Ladier & Alpan, 2016). Denote $\tau^{ts}_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ (hours) as the temporary storage time for a given product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$. Let $c_p^{ts}, p \in P$ (USD/hour) be the unit temporary storage cost for the product of type p. It is assumed that the temporary storage cost varies based on the product type. Perishable products will require temperature-controlled storage areas that are more expensive to operate when comparing to the storage areas for non-perishable products that do not maintain any specific temperature. The service time for a given truck t ($\tau_{tdo}^{st}, t \in T, d \in D, o \in O$ – hours) includes the following two components: (1) truck handing time ($\tau_t^{ht}, t \in T$ – hours), which is the time required to unload/load the products; and (2) transfer time from the assigned CDT door to the designated storage area ($\tau_{dn}^{tr}, d \in D, n \in N$ – hours). Let $z_{tn} = 1, t \in T, n \in N$ if the products carried by truck t are assigned to temporary storage area n (=0 otherwise). The service time of trucks is affected by the following factors: (1) the number of product types; (2) the quantity of each product type; (3) the weight and size of the product packaging units; and (4) the distance between the designated storage area and the assigned CDT door. In fact, the service time of the trucks docked at the doors farther away from the designated storage area is expected to be higher as compared to the trucks docked at the doors closer to the designated storage area.

In this study, it is assumed that the service times of trucks are deterministic in nature. Such an assumption has been commonly adopted by many of the previous CDT truck scheduling studies (Boloori Arabani et al., 2012; Dulebenets, 2019a; Liao et al., 2013; Miao et al., 2009). The service of a given truck incurs the unit service cost calculated per hour ($c_t^{tst}, t \in T$ – USD/hour). Furthermore, each inbound truck and each outbound truck are expected to complete their service at the considered cold-chain CDT according to the scheduled departure time ($\tau_t^{sd}, t \in T$ – hours). Whenever the service completion of a truck extends beyond the deadline, a unit delayed departure cost ($c_t^{dt}, t \in T$ – USD/hour) is incurred by the CDT operator. Several previous CDT truck scheduling studies discussed the importance of timely truck service and application of penalties in any case of the scheduled departure time violation (Dulebenets, 2019a; Ladier & Alpan, 2016; Theophilus et al., 2019).

3.4. Modeling decay of perishable products

As discussed earlier (see Section 3.3), once perishable products are unloaded from the inbound trucks, they will be transferred to the designated temporary storage areas or directly to the corresponding outbound trucks if these trucks are already docked at the cold-chain CDT. Different types of perishable products will require different types of storage areas. For example, certain food products require refrigeration and have to be stored under freezing temperatures (−25 °C to −10 °C), while certain types of pharmaceuticals have to be stored under cold temperatures (2–8 °C). It is assumed that a given perishable product deteriorates due to temperature fluctuations that are likely to occur throughout the unloading process from the inbound trucks, the transfer process between the CDT doors and the designated storage areas, and the loading process on the outbound trucks. However, the product deterioration substantially slows down when the products are placed in the temperature-controlled storage areas and is assumed to be insignificant. Furthermore, the product deterioration is assumed to be zero while the products are being inside the refrigerated trucks (a.k.a., "reefers") before the truck doors are opened and the loading/unloading process begins. Such an assumption can be justified by the fact that the refrigerated trucks are designed to maintain a specific temperature required for the perishable products they carry.

An additional cost is associated with the decay of perishable products throughout the product handling and transfer. Denote $c_p^{dc}, p \in P$ (USD/% decay) as the unit decay cost for the product of type p. As discovered during the literature review, an exponential function has been widely used to model the decay of perishable products (Piramuthu & Zhou, 2013; Piramuthu et al., 2013; Rong et al., 2011; Wang & Li, 2012; Yu & Nagurney, 2013). Hence, this study adopts the following relationship to estimate the quality of the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time τ based on the notations that were defined in Table 1:

$$\boldsymbol{Q}_{\underline{t}\bar{t}p}^{\tau} = Q_{\underline{t}\bar{t}p}^{0} e^{-\lambda_p \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}} \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P \tag{1}$$

The following relationship can be used to calculate the total handling and transfer time for the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ based on the notations that were defined in Table 1:

$$\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht} \geq \left(\tau_{\underline{t}}^{ht} + \sum_{d \in D} \sum_{o \in O} \sum_{n \in N} \tau_{dn}^{tr} z_{\underline{t}n} \boldsymbol{x}_{\underline{t}do} + \tau_{\bar{t}}^{ht} + \sum_{d \in D} \sum_{o \in O} \sum_{n \in N} \tau_{dn}^{tr} z_{\bar{t}n} \boldsymbol{x}_{\bar{t}do} \right) \varphi_{\underline{t}\bar{t}p} \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P \tag{2}$$

The following relationship can be used to calculate the change in quality of the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ based on the notations that were defined in Table 1:

$$\boldsymbol{\Delta Q}_{\underline{t}\bar{t}p} = Q_{\underline{t}\bar{t}p}^{0} - \boldsymbol{Q}_{\underline{t}\bar{t}p}^{\tau} \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P \tag{3}$$

Tracking changes in the perishable product quality throughout the cold supply chain operations is critical, as these product quality changes will determine the product "shelf life". The term "shelf life" is widely used in the studies dealing with perishable products and represents the total number of days remaining for a particular perishable product to be of an adequate quality for a given customer (Dulebenets & Ozguven, 2017; Grunow & Piramuthu, 2013). A proper management of cold-chain CDTs will prevent substantial product quality changes, which will ultimately increase the product shelf life.

3.5. Objective of the CDT operator

In the CDT truck scheduling problem, the cold-chain CDT operator has to make certain decisions, considering a number of important tradeoffs. The arriving inbound trucks must be scheduled for service at the available inbound doors, whereas the outbound trucks must be scheduled for service at the available outbound doors. The truck-to-door assignment directly affects the product transfer time between the CDT doors and the designated temporary storage areas. Increasing product transfer time is not desirable as it will further increase the total decay of perishable products throughout the service of arriving trucks. Along with the product perishability considerations, the cold-chain CDT operator has to prevent excessive waiting times of the arriving trucks before they could be docked at the CDT doors and reduce potential truck delayed departures. In the meantime, the temporary storage time of products has to be optimized as well to avoid excessive product inventory costs.

The main objective of the cold-chain CDT operator is to develop such a schedule for the arriving trucks that will yield the least total cost to be incurred throughout the service of arriving trucks that incorporates the following cost components: (i) total truck waiting cost; (ii) total truck service cost; (iii) total product inventory cost; (iv) total truck delayed departure cost; and (v) total product decay cost.

3.6. An illustrative example of a truck service

This section presents an example of the service of inbound and outbound trucks at the considered cold-chain CDT (see Fig. 2). This example assumes that inbound trucks "1" and "4" deliver perishable products that will be loaded on outbound truck "10". Inbound trucks "1" and "4" are assigned for service at inbound doors "1" and "4", respectively. On the other hand, outbound truck "10" is assigned for service at outbound door "10". The CDT operator allocates temperature-controlled storage area "1" for the products delivered by inbound trucks "1" and "4" (since they are perishable in nature). Storage area "1" is located close to outbound door "10" in order to ensure timely loading of outbound truck "10". Selection of the alternative storage area (e.g., storage area "2" instead of storage area "1") may increase the service time of outbound truck "10" due to increased travel distance for the forklift operators. The provided example exclusively focuses on the interaction between inbound trucks "1" and "4" and outbound truck "10" for simplicity. However, without loss of generality, more complex relationships between trucks can be modeled (e.g., inbound trucks "1", "4", and "5" deliver perishable products for outbound truck "10"; inbound truck "1" delivers perishable products not only for outbound truck "10" but also for outbound truck "6").

4. Mathematical model

This section of the manuscript provides a mathematical formulation for the CDT truck scheduling problem with product perishability considerations (**TSPCDT**), the adopted linearization techniques, and the linearized formulation of the **TSPCDT** mathematical model. A detailed description of the main notations used in the **TSPCDT** mathematical model is presented in Table 1.

4.1. Model formulation

The proposed mixed-integer nonlinear programming model for the CDT truck scheduling problem with product perishability considerations (**TSPCDT**) can be formulated as follows.

TSPCDT: CDT Truck Scheduling Problem with Product Perishability Considerations

$$minTC = \Bigg[\left(\sum_{t\in T}\tau_t^{wt}c_t^{wt}\right) + \left(\sum_{t\in T}\sum_{d\in D}\sum_{o\in O}\tau_{tdo}^{tst}c_t^{tst}\right) + \left(\sum_{\underline{t}\in T^{in}}\sum_{\bar{t}\in T^{out}}\sum_{p\in P}\tau_{\underline{t}\bar{t}p}^{ts}q_{\underline{t}\bar{t}p}c_p^{ts}\right) + \left(\sum_{t\in T}\tau_t^{dt}c_t^{dt}\right) + \left(\sum_{\underline{t}\in T^{in}}\sum_{\bar{t}\in T^{out}}\sum_{p\in P}\Delta Q_{\underline{t}\bar{t}p}q_{\underline{t}\bar{t}p}c_p^{dc}\right)\Bigg] \quad (4)$$

Subject to:

$$\sum_{d\in D}\sum_{o\in O}x_{tdo} = 1 \forall t\in T \quad (5)$$

$$\sum_{t\in T}x_{tdo} \leq 1 \forall d\in D, o\in O \quad (6)$$

$$\sum_{d\in D^{out}}\sum_{o\in O}x_{tdo} = 0 \forall t\in T^{in} \quad (7)$$

$$\sum_{d\in D^{in}}\sum_{o\in O}x_{tdo} = 0 \forall t\in T^{out} \quad (8)$$

$$\sum_{t^*\in T:t^*\neq t}\sum_{o^*\in O:o^*<o}\left(\tau_{t^*do^*}^{tst} + y_{t^*do^*}\right) + y_{tdo} \geq \tau_t^{at}x_{tdo} \forall t\in T, d\in D, o\in O \quad (9)$$

$$\tau_t^{st} \geq \sum_{t^*\in T:t^*\neq t}\sum_{o^*\in O:o^*<o}\left(\tau_{t^*do^*}^{tst} + y_{t^*do^*}\right) + y_{tdo} - M(1 - x_{tdo}) \forall t\in T, d\in D, o\in O \quad (10)$$

$$\tau_{\bar{t}}^{st} \geq \tau_{\underline{t}}^{st}\varphi_{\underline{t}\bar{t}p} \forall \underline{t}\in T^{in}, \bar{t}\in T^{out}, \underline{t}\neq\bar{t}, p\in P \quad (11)$$

$$\tau_{tdo}^{tst} = \left[\tau_t^{ht} + \sum_{n\in N}\left(\tau_{dn}^{tr}z_{tn}\right)\right]x_{tdo} \forall t\in T, d\in D, o\in O \quad (12)$$

$$\tau_t^{ft} = \tau_t^{st} + \sum_{d\in D}\sum_{o\in O}\tau_{tdo}^{tst} \forall t\in T \quad (13)$$

$$\tau_t^{wt} \geq \tau_t^{st} - \tau_t^{at} \forall t\in T \quad (14)$$

$$\tau_t^{dt} \geq \tau_t^{ft} - \tau_t^{sd} \forall t\in T \quad (15)$$

$$\tau_{\underline{t}\bar{t}p}^{tht} \geq \left(\tau_{\underline{t}}^{ht} + \sum_{d\in D}\sum_{o\in O}\sum_{n\in N}\tau_{dn}^{tr}z_{\underline{t}n}x_{\underline{t}do} + \tau_{\bar{t}}^{ht} + \sum_{d\in D}\sum_{o\in O}\sum_{n\in N}\tau_{dn}^{tr}z_{\bar{t}n}x_{\bar{t}do}\right)\varphi_{\underline{t}\bar{t}p} \forall \underline{t}\in T^{in}, \bar{t}\in T^{out}, \underline{t}\neq\bar{t}, p\in P \quad (16)$$

$$\tau_{\underline{t}\bar{t}p}^{ts} \geq \left(\tau_{\bar{t}}^{ft} - \tau_{\underline{t}}^{st}\right)\varphi_{\underline{t}\bar{t}p} \forall \underline{t}\in T^{in}, \bar{t}\in T^{out}, \underline{t}\neq\bar{t}, p\in P \quad (17)$$

$$Q_{\underline{t}\bar{t}p}^{\tau} = Q_{\underline{t}\bar{t}p}^{0}e^{-\lambda_p\tau_{\underline{t}\bar{t}p}^{tht}} \forall \underline{t}\in T^{in}, \bar{t}\in T^{out}, \underline{t}\neq\bar{t}, p\in P \quad (18)$$

$$\Delta Q_{\underline{t}\bar{t}p} = \left(Q_{\underline{t}\bar{t}p}^{0} - Q_{\underline{t}\bar{t}p}^{\tau}\right)\varphi_{\underline{t}\bar{t}p} \forall \underline{t}\in T^{in}, \bar{t}\in T^{out}, \underline{t}\neq\bar{t}, p\in P \quad (19)$$

$$x_{tdo}, z_{tn}, \varphi_{\underline{t}\bar{t}p} \in \mathbb{B} \forall t\in T, \underline{t}\in T^{in}, \bar{t}\in T^{out}, \underline{t}\neq\bar{t}, d\in D, o\in O, p\in P, n\in N \quad (20)$$

$$TC, y_{tdo}, \tau_t^{wt}, \tau_t^{st}, \tau_{tdo}^{tst}, \tau_t^{ft}, \tau_t^{dt}, \tau_{\underline{t}\bar{t}p}^{ts}, \tau_{\underline{t}\bar{t}p}^{tht}, Q_{\underline{t}\bar{t}p}^{\tau}, \Delta Q_{\underline{t}\bar{t}p}, \tau_t^{at}, \tau_t^{ht}, \tau_t^{sd}, \tau_{dn}^{tr}, Q_{\underline{t}\bar{t}p}^{0}, \lambda_p, c_t^{wt}, c_t^{tst}, c_p^{ts}, c_t^{dt}, c_p^{dc}, M \in \mathbb{R}^{+} \forall t\in T, \underline{t}\in T^{in}, \bar{t}\in T^{out}, \underline{t}\neq\bar{t}, d\in D, o\in O, p\in P, n\in N \quad (21)$$

$$q_{\underline{t}\bar{t}p} \in \mathbb{N} \forall \underline{t}\in T^{in}, \bar{t}\in T^{out}, \underline{t}\neq\bar{t}, p\in P \quad (22)$$

The objective function (4) of the **TSPCDT** mathematical model focuses on minimization of the total cost (***TC***), associated with the service of all the arriving trucks at the cold-chain CDT. The main cost components are incorporated within the model, including: (i) the total waiting cost; (ii) the total service cost; (iii) the total product inventory cost; (iv) the total delayed departure cost; and (v) the total product decay cost. Constraint set (5) assures that each arriving truck will be assigned for service in one of the service orders at one of the available doors. Constraint set (6) assures that no more than one truck can be assigned to an available door in a given service order. Constraint set (7) guarantees that no inbound truck will be assigned for service at an outbound door, while constraint set (8) assures that no outbound truck will be assigned for service at an inbound door. Constraint set (9) ensures that the service of a truck starts only after the truck arrives at the cold-chain CDT. Constraint set (10) indicates that the service of a given truck can start at a given door only after the completion of service of all the preceding trucks at the door.

Constraint set (11) guarantees that the service of a given outbound truck may begin after the beginning of service of any inbound truck that delivers the products for that outbound truck. Constraint set (12) estimates the total service time for each arriving truck based on the truck handling time (i.e., unloading or loading) and the product transfer time to the assigned temporary storage area. Constraint sets (13) through (15) compute the service finish time, waiting time, and delayed departure time of each arriving truck at the cold-chain CDT. Constraint set (16) computes the total handling and transfer time for each product type delivered by a given inbound truck for a certain outbound truck. Constraint set (17) computes the temporary storage time for each product type delivered by a given inbound truck for a certain outbound truck. Constraint set (18) estimates the quality of each product type at time τ. Constraint set (19) computes the change in quality of each product type delivered by a given inbound truck for a certain outbound truck. Constraint sets (20) through (22) show the nature of the model parameters and variables.

4.2. Linearization techniques

TSPCDT is a nonlinear mathematical model due to constraint set (18) that estimates the quality of the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time τ. The exponential function is expected to improve the accuracy of the product decay modeling but introduces a high degree of nonlinearity in the model. The linearization of the nonlinear model components is expected to reduce its computational complexity (Pasha et al., 2020). There are a number of approaches used in the literature to approximate nonlinear functions that include the following (Dulebenets, 2019b; Wang, Meng, & Liu, 2013): (1) static outer approximation method; (2) dynamic outer approximation method; (3) static secant approximation method; (4) dynamic secant approximation method; (5) enumeration method; (6) discretization method; and others. This study will rely on the static secant approximation method for linearizing the product decay function due to its reported effectiveness (Wang et al., 2013).

Let $A_s^0, s \in S$ be the piecewise static secant approximation of the product decay function, where $S = \{1, \ldots, w^{10}\}$ is a set of linear secant-based segments in the approximation. Denote $A_p^n\left(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}\right)$

$$= \frac{\boldsymbol{\Delta Q}_{\underline{t}\bar{t}p}}{Q_{\underline{t}\bar{t}p}^0} = \frac{Q_{\underline{t}\bar{t}p}^0 - \boldsymbol{Q}_{\underline{t}\bar{t}p}^{\tau}}{Q_{\underline{t}\bar{t}p}^0} = \frac{Q_{\underline{t}\bar{t}p}^0 - Q_{\underline{t}\bar{t}p}^0 e^{-\lambda_p \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}}}{Q_{\underline{t}\bar{t}p}^0} = 1 - e^{-\lambda_p \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}} \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$$

as a nonlinear decay function for the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$. Then, $A_p^n(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht})$ can be linearized for each product type p using its piecewise linear secant approximation $A_{ps}^0(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht})$. Some illustrative examples of the linear approximations for a given product decay function are presented in Fig. 3. In the considered examples, the product decay rate was assumed to be $\lambda = 0.029$ hour^{-1}, whereas the total handling and transfer time was assumed to vary from 0 h to ≈350 h. It can be noticed that an increase in the number of linear secant segments enhances the approximation accuracy but may also increase the CPU time due to an increase in the number of variables in the model.

4.3. Linearized model

Let $\boldsymbol{d}_{\underline{t}\bar{t}ps}^0 = 1, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P, s \in S$ if linear secant segment s is selected to approximate the function of decay for the product of type p delivered by inbound truck $\underline{t}$ for outbound truck $\bar{t}$. Denote $str_{ps}^0, p \in P, s \in S$ (hours) and $en_{ps}^0, p \in P, s \in S$ (hours) as the handling and transfer time values for the product of type p at the beginning and at the end of linear secant segment s, respectively. Let $Slp_{ps}^0, p \in P, s \in S$ (% decay/hour) and $Inc_{ps}^0, p \in P, s \in S$ (% decay) be the slope and the intercept of linear secant segment s for the product of type p, respectively. Then, the original nonlinear **TSPCDT** mathematical model can be reformulated as a linear problem (**TSPCDTL**) as follows.

TSPCDTL: Linearized CDT Truck Scheduling Problem with Product Perishability Considerations

$$\min \boldsymbol{TC} = \left[\left(\sum_{t \in T} \boldsymbol{\tau}_t^{wt} c_t^{wt}\right) + \left(\sum_{t \in T}\sum_{d \in D}\sum_{o \in O} \boldsymbol{\tau}_{tdo}^{tst} c_t^{tst}\right) + \left(\sum_{\underline{t} \in T^{in}}\sum_{\bar{t} \in T^{out}}\sum_{p \in P} \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{ts} q_{\underline{t}\bar{t}p} c_p^{ts}\right) + \left(\sum_{t \in T} \boldsymbol{\tau}_t^{dt} c_t^{dt}\right) + \left(\sum_{\underline{t} \in T^{in}}\sum_{\bar{t} \in T^{out}}\sum_{p \in P}\sum_{s \in S} A_{ps}^0(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}) q_{\underline{t}\bar{t}p} c_p^{dc}\right)\right] \tag{23}$$

Subject to:
Constraint sets (5)-(17), (20)-(22)

$$\sum_{s \in S} \boldsymbol{d}_{\underline{t}\bar{t}ps}^0 = 1 \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P \tag{24}$$

$$str_{ps}^0 \boldsymbol{d}_{\underline{t}\bar{t}ps}^0 \leq \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht} \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P, s \in S \tag{25}$$

$$en_{ps}^0 + M(1 - \boldsymbol{d}_{\underline{t}\bar{t}ps}^0) \geq \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht} \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P, s \in S \tag{26}$$

$$A_{ps}^0\left(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}\right) \geq Slp_{ps}^0 \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht} + Inc_{ps}^0 - M(1 - \boldsymbol{d}_{\underline{t}\bar{t}ps}^0) \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P, s \in S \tag{27}$$

In the reformulated **TSPCDTL** mathematical model, the objective function (23) focuses on minimization of the total cost ($\boldsymbol{TC}$), associated

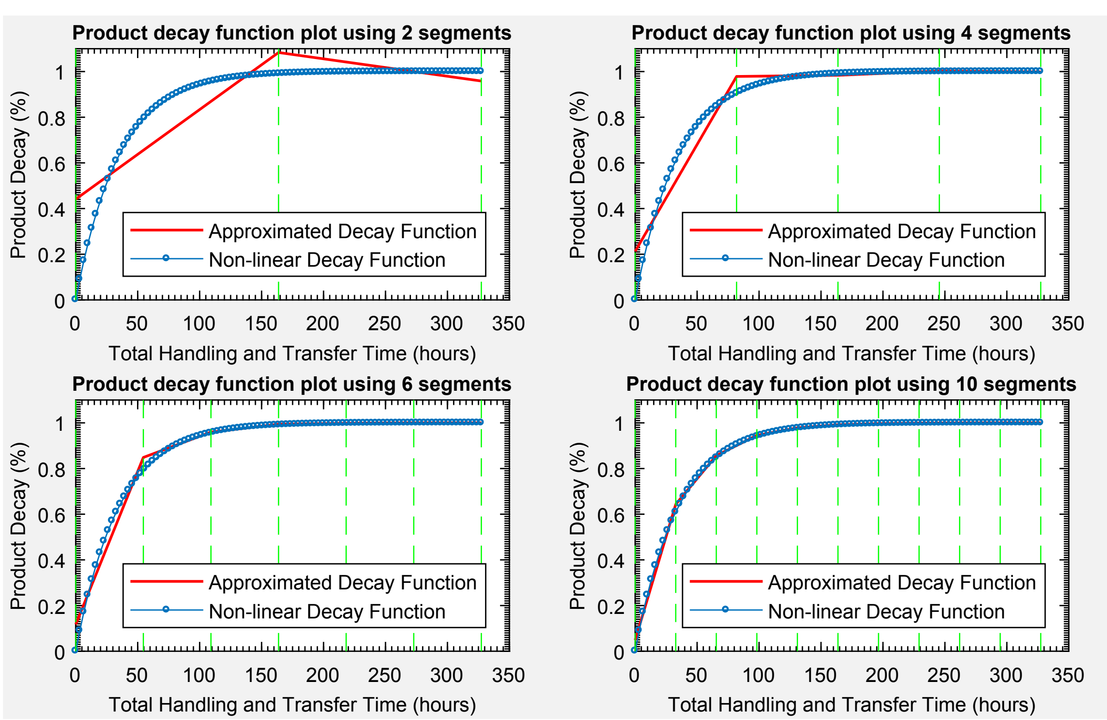

Fig. 3. Examples of the product decay function linearization.

with the service of all the arriving trucks at the cold-chain CDT. Constraint set (24) assures that only one linear secant segment will be chosen to approximate the function of decay for the product of type p delivered by inbound truck t for outbound truck $\bar{t}$. Constraint sets (25) and (26) define the range of the total handling and transfer time values when linear secant segment s is chosen to approximate the function of decay for the product of type p. Constraint set (27) estimates the approximated decay value for the product of type p. After implementation of the adopted linearization techniques, the **TSPCDTL** mathematical model can be solved using the existing mixed-integer programming optimization approaches (e.g., CPLEX, MOSEK, GUROBI). However, exact optimization approaches may require a substantial amount of CPU time to handle large-scale problem instances, and the development of approximate solution approaches will be necessary.

5. Solution algorithm

The mathematical formulation of the CDT truck scheduling problem with product perishability considerations (**TSPCDT**) can be viewed as a special case of the unrelated machine scheduling problem. In the unrelated machine scheduling problem, the decision maker has to assign the arriving jobs for processing on the available machines that have various speeds, and the processing time of a given job is affected by the job properties as well as the machine assigned (Pinedo, 2016). Similarly, the **TSPCDT** mathematical model assigns the arriving trucks (that can be viewed as "jobs") for service at the available doors (that can be viewed as "machines") and accounts for the changes in the service time of inbound and outbound trucks at the considered cold-chain CDT depending on the door assignment (see Section 3.3 for more details). The unrelated machine scheduling problems are recognized as NP-hard problems in a strong sense (Pinedo, 2016). The non-linear term represented by constraint set (18) that is used for the product decay estimations within the **TSPCDT** mathematical model is expected to increase the computational complexity even further.

Small-scale instances of **TSPCDT** and **TSPCDTL** can be solved to global optimality; however, approximate solution approaches would be necessary to tackle large-scale instances. This study will use a customized EA-based algorithm as the main solution approach for **TSPCDT**, since the EA-based algorithms were found to be effective for optimizing the cross-docking operations (Ladier & Alpan, 2016; Theophilus et al., 2019). However, the other metaheuristics will be considered as well throughout the numerical experiments to evaluate the computational efficiency of the developed EA. Note that metaheuristics can solve **TSPCDT** directly without application of linearization techniques. However, the exact mixed-integer linear programming optimization approaches can be applied for **TSPCDTL** only.

EAs can be considered as an adaptation of the theory of genetics and natural selection initially proposed by Charles Darwin (Darwin, 1859). EAs rely on the principle of "survival of the fittest" to search for promising solutions (i.e., the solutions with higher fitness values are given preference throughout the search process). Algorithm 1 presents the main steps used in the proposed EA. In step 0, the required data structures are initialized. The population of chromosomes (i.e., candidate solutions) is initialized in step 1 using the First Come First Served policy with truck sequence considerations (FCFS-TSC). The fitness values of the initial chromosomes are computed in step 2. Then, the algorithm checks the termination criterion and moves to steps 3–7 in case the criterion is not met. In step 3, the parent selection is conducted. In steps 4 and 5, the EA algorithm applies the crossover operator and mutation operator in order to generate and mutate the offspring, respectively. In step 6, the fitness values of the offspring chromosomes are evaluated. In step 7, the survivors for the next generation are selected. Once the termination criterion is achieved, the algorithm returns the final population that contains the best solution discovered for the **TSPCDT** mathematical model. Unlike the canonical EAs that rely on the binary chromosome representation, the proposed EA adopts the integer chromosome representation, which would be more suitable for the considered decision problem. Furthermore, unlike the canonical EAs that rely on the random chromosome initialization, the proposed EA deploys a problem-specific local search heuristic for the chromosome initialization (i.e., the FCFS-TSC heuristic). The next sections of this manuscript will provide more details regarding the aforementioned EA procedures.

Algorithm 1. (*Evolutionary Algorithm (EA)*)

$EA\left(Psize^{EA}, Tsize, Tsel, p^{c}, p^{m}, InputData\right)$
in: $Psize^{EA}$ – EA population size; $Tsize$ – tournament size;
$Tsel$ – number of individuals selected in each tournament; p^{c} – crossover probability;
p^{m} – mutation probability; $InputData$ – **TSPCDT** input data
out: $EAPop$ – final EA population
0: Initialization of data structures
1: $EAPop \leftarrow \boldsymbol{FCFS-TSC}\left(Psize^{EA}, InputData\right)$ ◁Initialization of chromosomes/population
2: $FitVals \leftarrow \boldsymbol{Fitness}(EAPop, InputData)$ ◁Computation of fitness values
While termination criterion is not met → go to steps 3-7, **else** → terminate
3: $Parents \leftarrow \boldsymbol{Tournament}(EAPop, FitVals, Tsize, Tsel)$ ◁Parent selection
4: $Offspring \leftarrow \boldsymbol{CrossoverPMX}(Parents, p^{c})$ ◁Crossover operations
5: $EAPop \leftarrow \boldsymbol{MutationINV}(Offspring, p^{m})$ ◁Mutation operations
6: $FitVals \leftarrow \boldsymbol{Fitness}(EAPop, InputData)$ ◁Computation of fitness values
7: $EAPop \leftarrow \boldsymbol{RWS}(EAPop, FitVals)$◁Survivor selection
8: **return** $EAPop$

5.1. Solution representation

The candidate solutions to the **TSPCDT** mathematical model are encoded into the chromosomes in the proposed EA algorithm. The chromosomes are assumed to have the integer representation. The chromosomes contain the information regarding the assignment of trucks to the cold-chain CDT doors along with the service order of trucks at each cold-chain CDT door. Fig. 4 provides an illustrative example of a chromosome, where the considered cold-chain CDT has three doors. Door "1" is assumed to be inbound, while doors "2" and "3" serve as outbound doors. Based on the illustrative example, inbound trucks "3", "1", "2", and "5" will be served at door "1". Truck "3" will be served first, followed by trucks "1" and "2". On the other hand, truck "5" will be served last at inbound door "1". Similarly, outbound trucks "7", "9", and "5" will be served at outbound door "2", whereas outbound trucks "4" and "6" will be served at outbound door "3". The term "gene" will be used to represent distinct components of every chromosome, while gene locations will be referred to as "loci" (singular – "locus"). In the example presented (see Fig. 4), the genes with inbound truck "3" and inbound door "1" are placed in locus "1".

5.2. Initialization of the EA population

The canonical FCFS strategy and its variations have been widely deployed in the EA literature and the freight terminal operations studies (Dulebenets, 2019a; Kavoosi et al., 2020). This study uses a modified FCFS strategy to initialize the chromosomes and population, named as

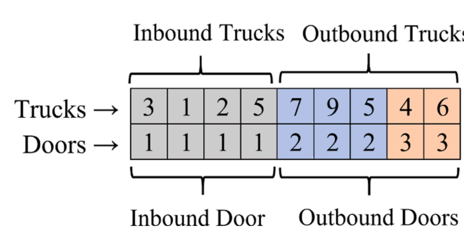

Fig. 4. Solution representation.

the First Come First Served policy with truck sequence considerations (FCFS-TSC). Unlike the canonical FCFS strategy, FCFS-TSC assures that a given outbound truck will be served after the inbound truck delivering the products for that outbound truck (i.e., truck sequence requirements are directly accounted for). The main steps used in FCFS-TSC are presented in Algorithm 2.

Algorithm 2. (*First Come First Served Policy with Truck Sequence Considerations (FCFS-TSC)*)

$\boldsymbol{FCFS-TSC}(T, T^{in}, T^{out}, D, O, N, \tau^{at}, \tau^{ht}, \tau^{tr}, z)$

in: $T = \{1, .., w^1\}$ – set of trucks; $T^{in} = \{1, .., w^2\}$ – set of inbound trucks; $T^{out} = \{1, .., w^3\}$ – set of outbound trucks;
$D = \{1, .., w^4\}$ – set of doors; $O = \{1, .., w^7\}$ – set of service orders; $N = \{1, ..., w^8\}$ – set of storage areas;
τ^{at} – arrival time of trucks; τ^{ht} – handing time of trucks; τ^{tr} – transfer time between doors and storage areas;
z – product-to-storage area assignment

out: $\boldsymbol{x}$ – initial truck-to-door-to-service order assignment

0: $|T^{in}| \leftarrow w^2; |T^{out}| \leftarrow w^3; |\boldsymbol{\tau}^{AV}| \leftarrow w^4; |\boldsymbol{x}| \leftarrow w^1.w^4.w^7; |\boldsymbol{\tau}^{st}| \leftarrow w^1; |\boldsymbol{\tau}^{ft}| \leftarrow w^1$ ◁Initialization
1: $T^{IS} \leftarrow \boldsymbol{Order}(T^{in}, \tau^{at})$ ◁Sorting inbound trucks in the order of their arrival
2: $T^{OS} \leftarrow \boldsymbol{Order}(T^{out}, \tau^{at})$ ◁Sorting outbound trucks in the order of their arrival
3: $T^S \leftarrow T^{IS} \cup T^{OS}$ ◁Combining the sorted inbound and outbound trucks
4: **for** $t = 1 : |T^S|$ **do**
5: $d \leftarrow argmin(\boldsymbol{\tau}_d^{AV})$ ◁Identification of the first available door
6: $o \leftarrow argmin(\boldsymbol{x}_{tdo})$ ◁Selection of the earliest truck service order
7: $\boldsymbol{x}_{tdo} \leftarrow 1$ ◁Assignment of the truck in the earliest service order
8: $\boldsymbol{\tau}_t^{st} \leftarrow max(\tau_t^{at}, \boldsymbol{\tau}_d^{AV})$ ◁Computing the truck service start time
9: $\boldsymbol{\tau}_{tdo}^{tst} \leftarrow \tau_t^{ht} + \sum_{n \in N}(\tau_{dn}^{tr} z_{tn})$ ◁Computing the truck total service time
10: $\boldsymbol{\tau}_t^{ft} \leftarrow \boldsymbol{\tau}_t^{st} + \boldsymbol{\tau}_{tdo}^{tst}$ ◁Computing the truck service finish time
11: $\boldsymbol{\tau}_d^{AV} \leftarrow \boldsymbol{\tau}_t^{ft}$ ◁Updating the door availability
12: $t \leftarrow t + 1$
13: **end for**
14: **return** $\boldsymbol{x}$

In step 0, the required data structures are initialized. In steps 1–3, the inbound trucks and outbound trucks are sorted based on their times of arrival, and the sorted sets of trucks are combined. The main loop of FCFS-TSC starts in step 4. In step 5, the first available CDT door is selected (either inbound or outbound door depending on the truck type). In step 6, the earliest order of truck service is identified. In step 7, the next truck in the combined set of all the trucks is selected for service at the first door available in the earliest order of service. Then, the start time of truck service is computed in step 8. In step 9, the total service time of the truck is computed as the summation of the total handling time and the total product transfer time. In step 10, the finish time of truck service is computed. In step 11, FCFS-TSC updates the CDT door availability. FCFS-TSC exits the loop after the last truck has been assigned for service at the cold-chain CDT. In step 14, FCFS-TSC returns the truck-to-door-to-service order assignment. Note that half of the EA population will be created using FCFS-TSC, while the remaining half will be created randomly to increase the EA population diversity.

5.3. Solution fitness estimations

Once the initial population is generated, the developed EA will start estimating fitness of each population chromosome. The mathematical function presented in equation (28) is used to compute the fitness values of the available chromosomes in the EA population. The function contains a penalty term (Ψ) to reduce the presence of infeasible chromosomes. The infeasibility may be caused by the crossover and mutation operators (see Section 5.5 for more details) due to violation of the truck sequence requirements (e.g., a given outbound truck will be served before the inbound truck delivering the products for that outbound truck, thereby contradicting realistic practices of cross-docking). The value of the penalty term will be set during the parameter tuning analysis (see Section 6.1 for more details).

$$min\boldsymbol{TC} = \Psi\Big[\Big(\sum_{t \in T} \boldsymbol{\tau}_t^{wt} c_t^{wt}\Big) + \Big(\sum_{t \in T}\sum_{d \in D}\sum_{o \in O} \boldsymbol{\tau}_{tdo}^{tst} c_t^{tst}\Big) + \Big(\sum_{t\_ \in T^{in}}\sum_{\bar{t} \in T^{out}}\sum_{p \in P} \boldsymbol{\tau}_{t\_\bar{t}p}^{ts} q_{t\_\bar{t}p} c_p^{ts}\Big) + \Big(\sum_{t \in T} \boldsymbol{\tau}_t^{dt} c_t^{dt}\Big) + \Big(\sum_{t\_ \in T^{in}}\sum_{\bar{t} \in T^{out}}\sum_{p \in P} \boldsymbol{\Delta Q}_{t\_\bar{t}p} q_{t\_\bar{t}p} c_p^{dc}\Big)\Big] \tag{28}$$

5.4. Parent selection

After entering its main loop, the developed EA implements the parent selection mechanism to choose the chromosomes that will undergo crossover and mutation. The tournament selection operator is used in this study for the selection of parents. The main steps performed by the tournament selection operator are presented in Algorithm 3. In step 0, the tournament selection operator starts the process by initializing the required data structure. The main loop executed by the tournament selection operator starts in step 1. In step 2, the chromosomes are randomly selected to participate in the tournament. In steps 3 through 8, a total of *Tsel* chromosomes that have the highest fitness are selected from the tournament and appended to the data structure that contains the parent chromosomes. The tournaments are continuously launched until the required number of parents are selected. In step 11, the tournament selection operator returns the data structure with the selected parent chromosomes.

Algorithm 3. (*Tournament Selection (Tournament)*)

$\boldsymbol{Tournament}(EAPop, FitVals, Tsize, Tsel)$

in: $EAPop$ – EA population in a given generation; $FitVals$ – fitness of EA chromosomes in a given generation;
$Tsize$ – tournament size; $Tsel$ – number of individuals selected in each tournament

out: $Parents$ – parent chromosomes in a given generation

0: $Parents \leftarrow \emptyset$ ◁Initialization
1: **for** $i = 1 : (|EAPop|/Tsel)$
2: $[Tour, FitVals^{Tour}] \leftarrow \boldsymbol{Rand}(EAPop, FitVals, Tsize)$ ◁Select chromosomes for the tournament
3: **for** $j = 1 : Tsel$
4: $k^* \leftarrow argmin(FitVals^{Tour})$ ◁Fittest chromosome search
5: $Parents \leftarrow Parents \cup \{Tour_{k^*}\}$ ◁Fittest chromosome selection as a parent
6: $Tour \leftarrow Tour - \{Tour_{k^*}\}$ ◁Fittest chromosome removal from the tournament
7: $j \leftarrow j + 1$
8: **end**
9: $i \leftarrow i + 1$
10: **end**
11: **return** $Parents$

5.5. Crossover and mutation operations

After selecting the parent chromosomes, the proposed EA deploys the crossover operator and mutation operator in order to generate and mutate the offspring, respectively. The crossover operator enables the EA algorithm with exploration of the promising search space domains. Selection of the appropriate crossover operator depends on the adopted chromosome representation (Eiben & Smith, 2015). The proposed EA deploys the Partially Mapped Crossover (PMX) that has been widely used in the studies that deployed EAs for the chromosomes with an integer representation (as the one that was adopted in this study – see Fig. 4). Fig. 5 provides an illustrative example of applying the PMX operator on two parent chromosomes.

In the first step, the PMX operator selects a random segment from parent chromosome "1", and the genes are directly copied to offspring "1" in the corresponding loci. In the example presented (see Fig. 5), the genes with trucks "2", "3", "6", "5", and "8" are copied from parent

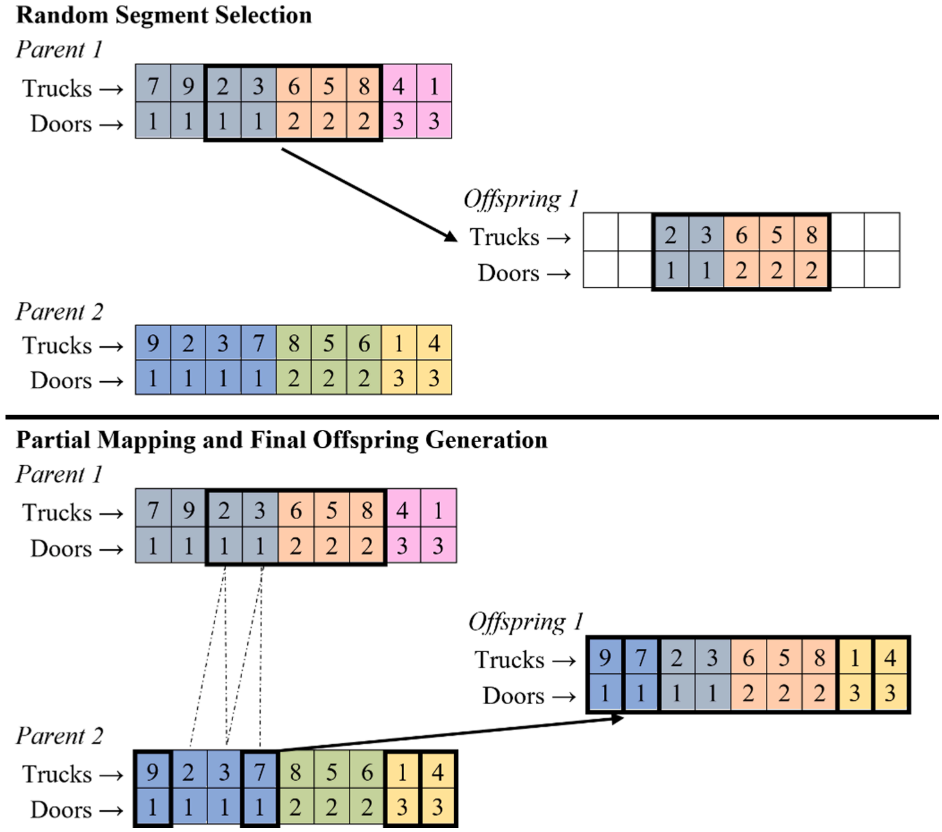

Fig. 5. Partially-mapped crossover example.

chromosome "1" to offspring "1". In the second step, the loci in parent chromosome "2" that correspond to the selected loci in parent chromosome "1" (i.e., loci "3" through "7") are checked for the genes that have not been copied to offspring "1". The example presented shows that the gene with truck "7" has not been copied. Therefore, the PMX operator performs partial mapping for the gene with truck "7" to determine the appropriate locus for that gene in offspring "1". Truck "7" occupies locus "4" in parent chromosome "2", while the same locus is occupied by truck "3" in parent chromosome "1". Truck "3" can be found in locus "3" of parent chromosome "2". Therefore, locus "3" of parent chromosome "1" is checked. Truck "2" occupies locus "3" in parent chromosome "1" and can be found in locus "2" of parent chromosome "2". Since locus "2" in offspring "1" was not occupied by any gene before parting mapping, the gene with truck "7" will be moved to locus "2". In the last step, the missing genes (i.e., the genes that contain trucks "9", "1", and "4") will be copied to offspring "1" in their respective loci.

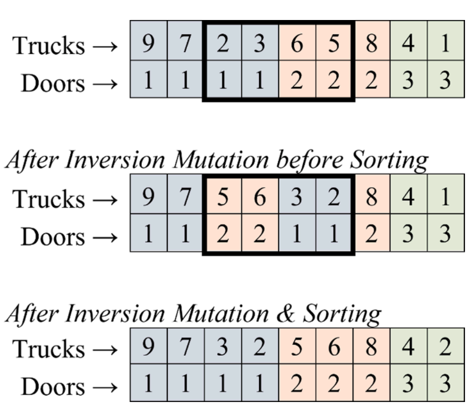

Fig. 6. Inversion mutation example.

The mutation operator enables the EA algorithm with exploitation of the promising search space domains. The proposed EA deploys the inversion mutation operator to perform mutation of the produced offspring chromosomes. Fig. 6 provides an illustrative example of applying the inversion mutation on the offspring chromosome. A portion of the chromosome is randomly selected, and the corresponding genes in the selected segment are inverted. In the example presented (see Fig. 6), the genes with trucks "2" and "5" exchange their positions, whereas the genes with trucks "3" and "6" exchange their positions as well. After performing the inversion mutation, the genes that represent the trucks will be sorted by their assigned doors to prevent disruption in the truck service orders (i.e., the genes with trucks "3" and "2" should be placed next to the genes with trucks "9" and "7" as all of these trucks are assigned for service at door "1").

5.6. Survivor selection

After generating and mutating the offspring chromosomes, the developed EA evaluates their fitness and applies the survivor selection operator to choose the offspring chromosomes that will be further assigned to the following generation. The roulette wheel selection operator is used in this study for the selection of survivors. The main steps performed by the roulette wheel selection operator are presented in Algorithm 4. In step 0, the roulette wheel selection operator initializes the required data structures. In steps 1–4, the fitness function value is adjusted for each chromosome in the EA population (since **TSPCDT** has a minimization objective). In step 5, the adjusted fitness values are normalized such that the summation of fitness values of the available population chromosomes is equal to one. Another loop executed by the roulette wheel selection operator starts in step 6. In step 7, a random number (*Val*) between 0.00 and 1.00 is generated. Then, the chromosome with a normalized fitness value, which is closer to the random number generated, is selected to become the next generation chromosome in step 9. This procedure continues until the required number of offspring are selected. In step 12, the roulette wheel selection operator returns the data structure with the selected offspring chromosomes that will be further assigned to the following generation.

Algorithm 4. (*Roulette Wheel Selection (RWS)*)

RWS$(EAPop, FitVals)$
in: *EAPop* – EA population in a given generation; *FitVals* – fitness of EA chromosomes in a given generation
out: *Offspring*– offspring chromosomes to be transferred to the next generation
0: $Offspring \leftarrow \emptyset; |FitVals^{aux}| \leftarrow |FitVals|$ ⊲Initialization
1: **for** $i = 1 : |EAPop|$ **do**
2: $FitVals_i^{aux} \leftarrow 1/FitVals_i$ ⊲Adjusting the chromosome fitness
3: $i \leftarrow i + 1$
4: **end for**
5: $FitVals^{aux} \leftarrow \boldsymbol{Normalize}(FitVals^{aux})$ ⊲Normalizing the adjusted fitness values
6: **for** $i = 1 : |EAPop|$ **do**
7: $Val \leftarrow \boldsymbol{Rand}(0.00; 1.00)$ ⊲Generate a random number between 0.00 and 1.00
8: $j \leftarrow \boldsymbol{find}(FitVals^{aux} - Val > 0)$ ⊲Select the chromosome based on step 7
9: $Offspring \leftarrow Offspring \cup \{EAPop_j\}$ ⊲Append the selected offspring
10: $i \leftarrow i + 1$
11: **end for**
12: **return***Offspring*

The proposed EA also uses the elitism strategy. The elitism strategy involves the storage and transfer of the fittest individual to the next generation before the parent selection, crossover, mutation, and survivor selection are applied to the chromosomes. Such a strategy allows preserving the fittest individual from one generation to another and preventing solution quality retrogression due to application of stochastic operators throughout the EA evolution (i.e., parent selection, crossover, mutation, and survivor selection).

5.7. EA termination

The developed EA terminates when one of the following conditions is satisfied: (a) no improvements in the solution quality have been identified after a specific number of generations; (b) the maximum number of generations has been reached. Such termination conditions were found to be popular in the EA literature (Eiben & Smith, 2015).

6. Numerical experiments

In this section of the manuscript, the conducted numerical experiments are described in detail. The main objective of the numerical experiments was to evaluate the computational performance of the developed EA metaheuristic against the other metaheuristics that have been often used for optimizing different cross-docking operations. The EA metaheuristic was compared against the following alternative algorithms: (a) Variable Neighborhood Search (i.e., VNS); (b) Tabu Search (i.e., TS); and (c) Simulated Annealing (i.e., SA). A detailed description of the VNS metaheuristic can be found in Hansen and Mladenović (2001), while Liao et al. (2013) provides a thorough description of the TS and SA metaheuristics. Unlike the canonical SA that works with a current solution and its one neighbor in each iteration, the developed SA was designed to generate multiple neighbors of a current solution (the number of neighbors was determined during the SA parameter tuning) in each iteration to improve the SA explorative capabilities. The EA, VNS, TS, and SA metaheuristics were encoded using MATLAB version 2016a. On the other hand, the **TSPCDTL** mathematical model was encoded using the General Algebraic Modeling System (GAMS) version 24.8. CPLEX was set as a mixed-integer linear programming optimization solver in GAMS. All the numerical experiments were executed on a DELL workstation that has an Intel Core i7-7700 k processor, 32 GB RAM, and Microsoft Windows 10 Operating System.

The scope of the numerical experiments includes the following steps: (1) selection of parameters for the **TSPCDT** mathematical model and the considered metaheuristics; (2) comparative analysis of the considered metaheuristics against exact optimization; (3) detailed evaluation of the considered metaheuristics in terms of various performance metrics; and (4) analysis of managerial implications using the most effective metaheuristic algorithm (identified in step 3). The next sections of this manuscript elaborate on each one of the analysis steps.

Table 2
The **TSPCDT** parameter data.

Parameter	Selected Value
Number of trucks: $\|T\|$ (trucks)	Value varies depending on the problem instance
Number of doors: $\|D\|$ (doors)	Value varies depending on the problem instance
Number of available service orders: $\|O\|$ (orders)	$\|O\| = \|T\|$
Number of available temporary storage areas: $\|N\|$ (storage areas)	5
Number of product types to be handled: $\|P\|$ (product types)	10
Average inter-arrival time of trucks: $\Delta\tau^{at}$ (hours)	$\Delta\tau^{at} = ED[0.083]$
Arrival time of truck*t*: $\tau_t^{at}, t \in T$ (hours)	$\tau_{t+1}^{at} = \tau_t^{at} + \Delta\tau^{at} \forall t \in T$
Handing time of truck *t*: $\tau_t^{ht}, t \in T$ (hours)	$\tau_t^{ht} = UD[0.167; 0.444] \forall t \in T$
Scheduled departure time for truck *t*: $\tau_t^{sd}, t \in T$ (hours)	$\tau_t^{sd} = \tau_t^{at} + min(\boldsymbol{\tau}_{tdo}^{tst}) \cdot UD[1.000; 1.200] \forall t \in T$
Transfer time from door *d* to storage area *n*: $\tau_{dn}^{tr}, d \in D, n \in N$ (hours)	$\tau_{dn}^{tr} = UD[1.250; 2.333] \forall d \in D, n \in N$
Temporary storage area assignment:$z_{tn}, t \in T, n \in N$	$z_{tn} = round(UD[0; 1]) \forall t \in T, n \in N$
Product-to-truck assignment:$\varphi_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	$\varphi_{\underline{t}\bar{t}p} = round(UD[0; 1]) \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, p \in P$
Product quantity: $q_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ (product units)	$q_{\underline{t}\bar{t}p} = round(UD[30; 40]) \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, p \in P$
Product quality at time “0”: $Q_{\underline{t}\bar{t}p}^0, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ (%)	$Q_{\underline{t}\bar{t}p}^0 = 100 \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, p \in P$
Decay rate of product type *p*: $\lambda_p, p \in P$ (hour^{-1})	$\lambda_p = UD[0.010; 0.030] \forall p \in P$
Unit waiting cost for truck *t*: $c_t^{wt}, t \in T$ (USD/hour)	$c_t^{wt} = UD[100; 200] \forall t \in T$
Unit service cost for truck *t*: $c_t^{tst}, t \in T$ (USD/hour)	$c_t^{tst} = UD[150; 300] \forall t \in T$
Unit temporary storage cost for product type *p*: $c_p^{ts}, p \in P$ (USD/hour)	$c_p^{ts} = UD[50; 100] \forall p \in P$
Unit delayed departure cost for truck *t*: $c_t^{dt}, t \in T$ (USD/hour)	$c_t^{dt} = UD[350; 700] \forall t \in T$
Unit decay cost for product type *p*: $c_p^{dc}, p \in P$ (USD/% decay)	$c_p^{dc} = UD[100; 150] \forall p \in P$
Sufficiently large positive number:M	10,000

6.1. Parameter selection

6.1.1. TCPCDT mathematical model

The parameter values for the **TSPCDT** were set based on the previous CDT truck scheduling studies (Boloori Arabani et al., 2012; Dulebenets, 2019a; Liao et al., 2013; Theophilus et al., 2019) and are presented in Table 2. The arrival times of trucks at the considered cold-chain CDT, previously negotiated between the corresponding logistics companies and the CDT operator, were modeled based on an exponential distribution with an average inter-arrival time of 5 min or ≈0.083 h (i.e., $ED[0.083]$). The handling time of trucks, which refers to either unloading of inbound trucks or loading of outbound trucks, was assumed to vary from 10 min (or ≈0.167 h) to 26.667 min (or ≈0.444 h): $\tau_t^{ht} = UD[0.167; 0.444] \forall t \in T$ (hours). Note that term $UD[Var_1; Var_2]$ represents the generation of pseudorandom numbers uniformly distributed between Var_1 and Var_2. The scheduled departure time of each truck was set based on its time of arrival and the time required completing its service: $\tau_t^{sd} = \tau_t^{at} + min(\tau_{tdo}^{tst}) \cdot UD[1.000; 1.200] \forall t \in T$ (hours). The time required to transfer the products that were delivered by a given inbound truck from a given CDT door to one of the storage areas ($\tau_{dn}^{tr}, d \in D, n \in N$) was assumed to vary from 75 min (or ≈1.250 h) to 140 min (or ≈2.333 h). The considered cold-chain CDT was assumed to have 5 temporary storage areas ($|N| = 5$). The quantity of a given product type transported by a given inbound truck for a given outbound truck ($q_{t\_\bar{t}p}, t\_ \in T^{in}, \bar{t} \in T^{out}, p \in P$) was assumed to vary from 30 product units to 40 product units.

The inbound trucks were assumed to deliver ten different product types ($|P| = 10$). The quality of all the products delivered to the considered cold-chain CDT at time "0" ($Q^0_{t\_\bar{t}p}, t\_ \in T^{in}, \bar{t} \in T^{out}, t\_ \neq \bar{t}, p \in P$) was assumed to be 100%. Furthermore, the decay rate of a given product type ($\lambda_p, p \in P$) ranged between 0.010 h^{-1} and 0.030 h^{-1}. The unit truck waiting cost ($c_t^{wt}, t \in T$) varied from 100 USD/hour to 200 USD/hour, whereas the unit truck service cost ($c_t^{tst}, t \in T$) varied from 150 USD/hour to 300 USD/hour. On the other hand, the unit temporary product storage cost ($c_p^{ts}, p \in P$) ranged between 50 USD/hour and 100 USD/hour, while the unit truck delayed departure cost ($c_t^{dt}, t \in T$) ranged between 350 USD/hour and 700 USD/hour. As for the unit product decay cost ($c_p^{dc}, p \in P$), it was assumed to vary from 100 USD/% decay to 150 USD/% decay. This study considered two groups of problem instances based on their scale. The first group included 30 small-scale problem instances with the number of CDT doors ranging between 2 and 4 and the number of trucks ranging between 6 and 15. The second group included 30 large-scale problem instances with the number of CDT doors ranging between 4 and 8 and the number of trucks ranging between 84 and 120.

6.1.2. Metaheuristics

In this study, the parameter tuning analysis had to be performed for a total of four metaheuristics (EA, VNS, TS, and SA). The main objective of the parameter tuning was to determine the most appropriate combination of parameter values to enhance the performance of each metaheuristic. The proposed EA has a total of six parameters: (1) population size; (2) penalty for infeasible solutions; (3) tournament size; (4) number of individuals selected in each tournament; (5) crossover probability; and (6) mutation probability. The selection of EA parameter values was conducted based on the evaluation of each parameter combination with respect to the tradeoff between the objective function value and the CPU time (e.g., increasing population size may enhance the explorative capabilities of the algorithm and improve the objective function values at termination but will require more CPU time). The EA parameter tuning was conducted for the three large-scale problem instances selected at random, assuming that each parameter has three candidate values (see Table 3). Furthermore, ten EA replications were executed throughout the course of the conducted analysis for each parameter combination to accurately estimate the average values of objective function as well as the average CPU time. Therefore, EA was launched for a total of (3 candidate values)$^{(6\ parameters)}$ $\cdot$ (3 problem instances) $\cdot$ (10 replications for each combination) = 21,870 times throughout the analysis. The results from the performed EA parameter tuning are reported in Table 3.

A similar analysis was also conducted for the VNS, TS, and SA metaheuristics in order to identify the appropriate parameter values for these algorithms. Based on preliminary EA runs, the maximum number of generations was set to 3000 generations, while the maximum number of generations without objective function improvements was set to 2500 generations. Similarly, the maximum number of iterations was set to 3000 iterations for VNS, TS, and SA, while the maximum number of iterations without objective function improvements was set to 2500 iterations for all the other metaheuristics considered.

Table 3
Selected values of the parameters of metaheuristics.

Algorithm	Parameter	Candidate values	Selected value
EA	Population size ($Psize^{EA}$)	[30; 40; 50]	50
EA	Penalty (Ψ)	[4.00; 6.00; 8.00]	4.00
EA	Tournament size ($Tsize$)	[8; 10; 13]	13
EA	Number of individuals selected in each tournament ($Tsel$)	[4; 5; 7]	7
EA	Crossover probability (p^c)	[0.30; 0.50; 0.70]	0.30
EA	Mutation probability (p^m)	[0.01; 0.02; 0.05]	0.01
VNS	Neighborhood size ($Psize^{VNS}$)	[30; 40; 50]	50
VNS	Penalty (Ψ)	[4.00; 6.00; 8.00]	4.00
VNS	Exchange probability (p^{ex})*	[0.01; 0.02; 0.05]	0.01
TS	Number of solutions evaluated during local search ($Psize^{TS}$)	[30; 40; 50]	40
TS	Penalty (Ψ)	[4.00; 6.00; 8.00]	4.00
TS	Exchange probability (p^{ex})*	[0.01; 0.02; 0.05]	0.02
TS	Size of the Tabu list ($Tabu^{max}$) – % of $Psize^{TS}$	[0.10; 0.15; 0.20]	0.20
SA	Number of solutions evaluated during local search ($Psize^{SA}$)	[30; 40; 50]	50
SA	Penalty (Ψ)	[4.00; 6.00; 8.00]	4.00
SA	Exchange probability (p^{ex})*	[0.01; 0.02; 0.05]	0.01
SA	Initial temperature (τ^0)	[1,502; 1,602; 1,702]	1,602
SA	Temperature interval ($d\tau$)	[0.3; 0.4; 0.5]	0.5

* Exchange probability refers to the probability of inverting a set of consecutive trucks in the current solution when generating the neighbor solutions during local search.

6.2. Comparative analysis against exact optimization

As a part of the numerical experiments, a supplementary computational analysis was performed to examine the accuracy of the solutions of the considered metaheuristics. The solutions returned by the metaheuristics were compared with the solutions obtained by CPLEX for all the generated small-scale problem instances, where the number of CDT doors ranged between 2 and 4 (an increment of 1 CDT door was adopted) and the number of trucks ranged between 6 and 15 (an increment of 1 truck was adopted). Note that CPLEX was applied to the linearized version of the **TSPCDT** mathematical model (i.e., **TSPCDTL**) to overcome the increasing complexity of the model due to additional nonlinear terms (see Section 4.3). A total of ten linear secant segments were used to approximate the function of decay for each product type, as piecewise functions with ten linear secant segments demonstrated high approximation accuracy (see Fig. 3) and required a reasonable CPU time. The target optimality gap of CPLEX was set to 0.5%, and its CPU time limit was restricted to 1 h (i.e., 3600 sec). The results from the detailed comparative analysis of the considered metaheuristics against CPLEX are reported in Table 4, which contains the following information for every small-scale problem instance: (i) the problem instance number; (ii) the number of doors – $|D|$; (iii) the number of trucks – $|T|$; (iv) the $\boldsymbol{TC}$ value of the optimal truck schedule obtained by CPLEX; (v) the CPU time for CPLEX; (vi) the $\boldsymbol{TC}$ value of the truck schedule obtained by each metaheuristic (average over ten replications); (vii) the CPU time for each metaheuristic (average over ten replications).

The analysis results demonstrated that CPLEX was very sensitive to increasing problem size even after linearizing the original **TSPCDT** mathematical model. In particular, the small-scale problem instances

Table 4
CPLEX vs. considered metaheuristics for small-scale problem instances.

Instance	|D|	|T|	CPLEX		EA		VNS		TS		SA	
			***TC*(USD)**	CPU (sec)	***TC*(USD)**	CPU (sec)	***TC*(USD)**	CPU (sec)	***TC*(USD)**	CPU (sec)	***TC*(USD)**	CPU (sec)
1	2	6	**17,792.8**	7.34	**17,792.8**	15.86	**17,792.8**	10.71	**17,792.8**	9.48	**17,792.8**	12.27
2	2	7	**21,691.9**	80.39	**21,691.9**	15.95	**21,691.9**	12.16	**21,691.9**	10.21	**21,691.9**	14.12
3	2	8	**25,150.1**	326.19	**25,150.1**	17.83	**25,150.1**	12.54	**25,150.1**	11.18	**25,150.1**	13.50
4	2	9	33,977.1	3,604.30	**33,640.7**	16.90	**33,640.7**	13.28	**33,640.7**	10.64	**33,640.7**	14.17
5	2	10	35,100.1	3,602.50	**34,482.9**	17.02	**34,482.9**	14.64	**34,482.9**	11.38	**34,482.9**	14.76
6	2	11	45,833.5	3,602.52	**44,287.9**	18.10	**44,287.9**	15.15	**44,287.9**	11.57	**44,287.9**	15.08
7	2	12	53,645.1	3,603.44	**51,586.8**	20.47	53,006.0	17.23	52,837.9	14.00	53,242.4	17.30
8	2	13	64,710.0	3,602.49	**61,752.0**	22.25	62,274.1	17.66	62,127.7	14.60	64,475.0	19.43
9	2	14	77,202.2	3,602.91	**71,782.6**	21.09	74,443.0	18.53	72,116.6	15.59	74,724.9	20.07
10	2	15	98,720.3	3,602.45	**90,304.0**	23.39	92,598.1	19.61	90,457.7	15.44	92,692.0	21.02
11	3	6	**13,248.7**	14.58	**13,248.7**	14.57	**13,248.7**	11.36	**13,248.7**	11.12	**13,248.7**	12.49
12	3	7	**16,082.8**	55.34	**16,082.8**	16.00	**16,082.8**	11.55	**16,082.8**	10.69	**16,082.8**	13.27
13	3	8	**19,063.7**	3,009.34	**19,063.7**	16.23	**19,063.7**	12.08	**19,063.7**	10.76	**19,063.7**	14.20
14	3	9	22,825.4	3,602.42	**22,563.7**	17.04	**22,563.7**	13.41	**22,563.7**	11.46	**22,563.7**	14.44
15	3	10	26,976.1	3,602.53	**26,410.9**	15.59	**26,410.9**	13.58	**26,410.9**	11.66	**26,410.9**	15.37
16	3	11	32,104.3	3,604.24	**30,937.9**	17.22	31,241.9	14.90	30,957.9	12.39	31,565.1	15.70
17	3	12	44,327.3	3,603.56	**42,516.1**	19.83	44,112.4	17.15	42,644.0	14.41	44,162.8	18.21
18	3	13	50,911.4	3,604.26	**48,097.7**	20.76	48,575.9	16.76	48,312.3	15.05	48,851.2	19.02
19	3	14	57,099.5	3,603.09	**53,022.1**	21.35	53,577.8	17.48	53,099.1	15.03	54,522.1	19.72
20	3	15	73,675.0	3,602.69	**66,421.7**	22.49	68,558.9	19.15	68,359.0	16.57	68,921.7	20.99
21	4	6	**12,025.6**	43.20	**12,025.6**	13.32	**12,025.6**	10.77	**12,025.6**	11.34	**12,025.6**	11.53
22	4	7	**14,359.9**	370.89	**14,359.9**	14.21	**14,359.9**	13.04	**14,359.9**	11.56	**14,359.9**	12.42
23	4	8	**16,494.3**	3,161.14	**16,494.3**	15.02	**16,494.3**	13.37	**16,494.3**	12.55	**16,494.3**	13.88
24	4	9	19,636.0	3,604.63	**19,326.8**	16.46	**19,326.8**	13.89	**19,326.8**	11.70	**19,326.8**	13.76
25	4	10	20,877.1	3,604.78	**20,393.7**	15.94	**20,393.7**	14.14	**20,393.7**	11.94	**20,393.7**	14.44
26	4	11	25,760.9	3,604.44	24,815.4	16.60	25,018.5	13.95	**24,735.4**	12.40	25,253.8	15.27
27	4	12	34,076.6	3,603.32	**32,665.4**	18.55	33,391.2	16.44	32,983.6	13.62	33,855.8	18.11
28	4	13	41,191.8	3,603.11	**38,378.6**	19.17	38,596.6	17.44	**38,378.6**	14.19	40,356.4	18.84
29	4	14	47,862.6	3,602.82	43,906.6	19.88	43,996.8	17.12	**43,473.7**	15.48	44,628.2	19.39
30	4	15	60,974.2	3,602.75	**54,783.6**	21.62	56,632.6	19.48	56,083.9	17.44	56,929.5	21.15
Average:			37,446.5	2,757.92	**35,599.6**	18.02	36,101.3	14.95	35,786.1	12.85	36,373.3	16.13

*Bold font is used for the best objective function values achieved.

Table 5
Comparative analysis of the considered metaheuristics for large-scale problem instances.

Instance	|D|	|T|	EA		VNS		TS		SA	
			TC (10^3 USD)	CPU (sec)	***TC*** (10^3 USD)	CPU (sec)	***TC*** (10^3 USD)	CPU (sec)	***TC*** (10^3 USD)	CPU (sec)
31	4	84	**1,268.34**	98.03	1,603.14	96.31	1,553.21	80.44	2,251.24	99.98
32	4	88	**1,435.39**	108.18	1,687.45	101.50	1,676.15	87.12	2,644.48	105.09
33	4	92	**1,582.43**	117.81	2,259.87	107.86	2,008.82	92.01	3,078.55	110.47
34	4	96	**1,718.32**	126.29	2,696.11	118.95	2,221.14	97.49	3,364.65	114.31
35	4	100	**1,898.07**	137.66	3,090.78	124.12	2,775.18	99.87	3,831.24	122.62
36	4	104	**2,253.20**	156.52	3,433.42	135.01	2,395.98	103.19	4,644.62	137.02
37	4	108	**2,448.96**	147.43	3,716.47	139.07	2,663.28	109.05	5,030.33	144.92
38	4	112	**2,667.63**	151.99	4,335.81	151.27	2,755.23	114.46	5,504.95	151.38
39	4	116	**2,919.21**	152.70	4,975.67	161.70	3,053.66	121.97	5,987.11	158.87
40	4	120	**3,108.67**	160.36	5,331.87	166.81	3,464.04	131.60	6,593.46	166.59
41	6	84	**880.66**	97.90	1,040.02	97.69	1,023.11	81.67	1,620.83	101.44
42	6	88	**1,022.12**	105.39	1,343.31	104.68	1,153.38	86.37	1,912.59	110.78
43	6	92	**1,142.79**	111.60	1,526.22	110.50	1,351.95	92.29	2,196.78	112.60
44	6	96	**1,214.00**	117.86	1,645.67	118.39	1,405.21	97.37	2,460.04	118.91
45	6	100	**1,334.10**	129.62	1,964.91	132.00	1,508.63	101.32	2,787.06	126.50
46	6	104	**1,662.72**	131.82	2,337.82	139.14	1,678.72	103.54	3,227.21	133.63
47	6	108	**1,831.01**	139.34	2,375.70	145.21	1,891.55	109.18	3,503.90	149.24
48	6	112	**1,934.07**	146.19	2,648.32	147.88	1,962.68	114.69	3,750.66	156.42
49	6	116	**2,151.45**	152.67	3,162.15	158.72	2,258.25	120.53	4,133.25	164.27
50	6	120	**2,407.47**	160.49	3,486.78	162.40	2,410.51	127.98	4,586.20	172.42
51	8	84	**707.44**	100.70	806.08	96.23	782.38	77.01	1,308.67	100.37
52	8	88	**801.59**	105.92	994.87	109.97	878.31	82.66	1,536.22	110.74
53	8	92	**913.88**	111.66	1,115.40	121.52	1,058.02	87.37	1,732.87	119.17
54	8	96	**963.86**	124.40	1,259.69	134.66	1,102.81	95.07	1,940.31	121.34
55	8	100	**1,090.57**	132.20	1,590.59	140.93	1,323.82	98.36	2,171.83	131.98
56	8	104	**1,396.17**	140.02	1,672.85	145.87	1,397.78	103.88	2,478.61	136.33
57	8	108	1,527.76	144.06	1,689.21	153.05	**1,469.99**	109.81	2,709.95	137.39
58	8	112	1,643.97	151.14	2,077.30	162.90	**1,633.94**	115.14	2,926.57	154.69
59	8	116	**1,816.76**	158.00	2,362.89	173.51	1,836.52	121.58	3,217.98	152.02
60	8	120	1,972.21	165.50	2,517.34	173.36	**1,936.66**	127.45	3,485.67	159.63
Average:			**1,657.16**	132.78	2,358.26	134.37	1,821.03	103.02	3,220.59	132.70

*Bold font is used for the best objective function values achieved.

with 9 trucks and more could not be solved to global optimality within the imposed CPU time limit. Such a finding proves a high computational complexity of the studied CDT truck scheduling problem and highlights the need for application of metaheuristics. Furthermore, it can be observed that all the considered metaheuristics were able to achieve the optimal solutions provided by CPLEX for problem instances 1, 2, 3, 11, 12, 13, 21, 22, and 23 and required much smaller CPU time as compared to CPLEX. In particular, the EA, VNS, TS, and SA metaheuristics required on average 18.02 sec, 14.95 sec, 12.85 sec, and 16.13 sec, respectively, over the generated small-scale problem instances. The average values of the objective function comprised 35,599.6 USD, 36,101.3 USD, 35,786.1 USD, and 36,373.3 USD for the EA, VNS, TS, and SA metaheuristics, respectively. Therefore, the developed EA typically yielded higher quality solutions for the generated small-scale problem instances. However, the difference among the objective function values that were returned by the considered metaheuristics was not substantial as the analyzed problem instances were small-scale (the number of CDT doors did not exceed 4, while the number of trucks did not exceed 15).

6.3. Detailed evaluation of the metaheuristics

As a part of the numerical experiments, a set of supplementary computational analyses were performed to evaluate the considered metaheuristics in terms of different performance metrics for all the generated large-scale problem instances, where the number of CDT doors ranged between 4 and 8 (an increment of 2 CDT doors was adopted) and the number of trucks ranged between 84 and 120 (an increment of 4 trucks was adopted). First, the objective function and CPU time values that were returned by the considered metaheuristics were compared. Second, the stability of the objective function at termination along with the required CPU time was assessed for each metaheuristic considered. Third, a detailed investigation of the convergence patterns was performed for each metaheuristic considered. The next sections of this manuscript elaborate more on each one of the conducted analyses.

6.3.1. Objective function and CPU time values

The analysis of the objective function and CPU time values is a critical step in evaluating the computational performance of each metaheuristic. Ideally, an efficient metaheuristic algorithm returns good-quality solutions that have desirable values of the objective function within a reasonable CPU time (unlike CPLEX that can produce good-quality or even optimal solutions but will incur a prohibitively large CPU time). The results from the detailed comparative analysis of the considered metaheuristics in terms of the objective function and CPU time values are reported in Table 5, which contains the following information for every large-scale problem instance: (i) the problem instance number; (ii) the number of doors – $|D|$; (iii) the number of trucks – $|T|$; (iv) the $\boldsymbol{TC}$ value of the truck schedule obtained by each metaheuristic (average over ten replications); (v) the CPU time for each metaheuristic (average over ten replications).

The analysis results demonstrated that the average values of the objective function comprised $1657.16 \cdot 10^3$ USD, $2358.26 \cdot 10^3$ USD, $1821.03 \cdot 10^3$ USD, and $3220.59 \cdot 10^3$ USD for the EA, VNS, TS, and SA metaheuristics, respectively. Therefore, EA was typically able to obtain the best objective function values. Such a performance of EA can be explained by the fact that EA is recognized as a population-based metaheuristic that has a high capability of exploring the solution search space for promising domains and good-quality solutions. On the contrary, the VNS, TS and SA metaheuristics are recognized as single-solution-based metaheuristics that primarily rely on local search and are limited to a certain extent in their capabilities of exploring the solution search space for promising domains and good-quality solutions. Throughout the analysis, the statistical significance in the difference among the average objective function values that were returned by EA and the other considered metaheuristics was assessed. The assessment was performed by means of a paired z-test. In particular, a total of three types of a paired z-test were conducted for each large-scale problem instance: (a) "EA vs. VNS"; (b) "EA vs. TS"; and (c) "EA vs. SA". The null hypothesis (H_0) of each z-test assumed that there was no substantial difference among the average objective function values of EA and the other metaheuristics. The alternate hypothesis (H_a), on the other hand, assumed that there was a substantial difference among the average objective function values of metaheuristics.

The z-statistic was computed as follows for each paired z-test: $z = \frac{mean(\boldsymbol{TC}_m) - mean(\boldsymbol{TC}_{EA})}{\sqrt{\frac{(std(\boldsymbol{TC}_m))^2}{n} + \frac{(std(\boldsymbol{TC}_{EA}))^2}{n}}}$, where $mean(\boldsymbol{TC}_m)$ is the average of the objective function values obtained by metaheuristic m; $mean(\boldsymbol{TC}_{EA})$ is the average of the objective function values obtained by EA; $std(\boldsymbol{TC}_m)$ is the objective function standard deviation for metaheuristic m; $std(\boldsymbol{TC}_{EA})$ is the objective function standard deviation for EA; and $n = 2$ as two algorithms with ten replications each were compared during each z-test. Based on the outcomes from the performed paired z-tests, the average values of the z-statistic over all the generated large-scale problem instances comprised 30.61, 5.36, and 46.08 for the "EA vs. VNS", "EA vs. TS", and "EA vs. SA" tests, respectively. Therefore, the null hypothesis can be rejected at 0.01% significance level with a critical z-value of 3.819, and the objective function values of EA are statistically superior as compared to the ones obtained by the other metaheuristics. In terms of the CPU time values, the EA, VNS, TS, and SA metaheuristics required on average 132.78 sec, 134.37 sec, 103.02 sec, and 132.70 sec, respectively, over the generated large-scale problem instances. The maximum CPU time required by any of the considered metaheuristics did not exceed 180 sec (or 3 min) over all the generated large-scale problem instances. Such a CPU time can be viewed as reasonable, taking into account the fact that the large-scale problem instances with up to 8 CDT doors and 120 trucks were evaluated throughout the numerical experiments.

6.3.2. Objective function and CPU time variations

The considered metaheuristics rely on a variety of stochastic operators (e.g., the developed EA deploys the crossover operator and mutation operator in order to generate and mutate the offspring). The stochastic operators allow metaheuristics effectively changing the current solutions in order to identify more promising solutions. In the meantime, application of stochastic operators also leads to the variations in the values of objective function at termination. Furthermore, execution of the same metaheuristic for the same problem instance on a given CPU may require different computational times. The latter phenomenon can be caused by the additional procedures a given CPU may run during the metaheuristic execution (e.g., basic software updates, antivirus scan, hardware scan). Significant variations in the objective function values and the CPU time are not desirable.

As a part of the numerical experiments, a supplementary computational analysis was performed to examine the variations in the values of objective function and the CPU time for the considered metaheuristics. The coefficients of variation values for the objective function and the CPU time were computed for every metaheuristic and every large-scale problem instance over ten replications performed. The results from the conducted analysis are reported in Fig. 7. The analysis results demonstrated that the coefficient of variation of the objective function values that were returned by the considered metaheuristics did not exceed 2.82% for the generated large-scale problem instances. Moreover, the coefficient of variation of the CPU time required by the considered metaheuristics did not exceed 5.91% for the generated large-scale problem instances. Hence, the considered metaheuristics can be recognized as reliable algorithms in terms of both objective function and CPU time values.

6.3.3. Algorithmic convergence patterns

The analysis of the algorithmic convergence patterns is another

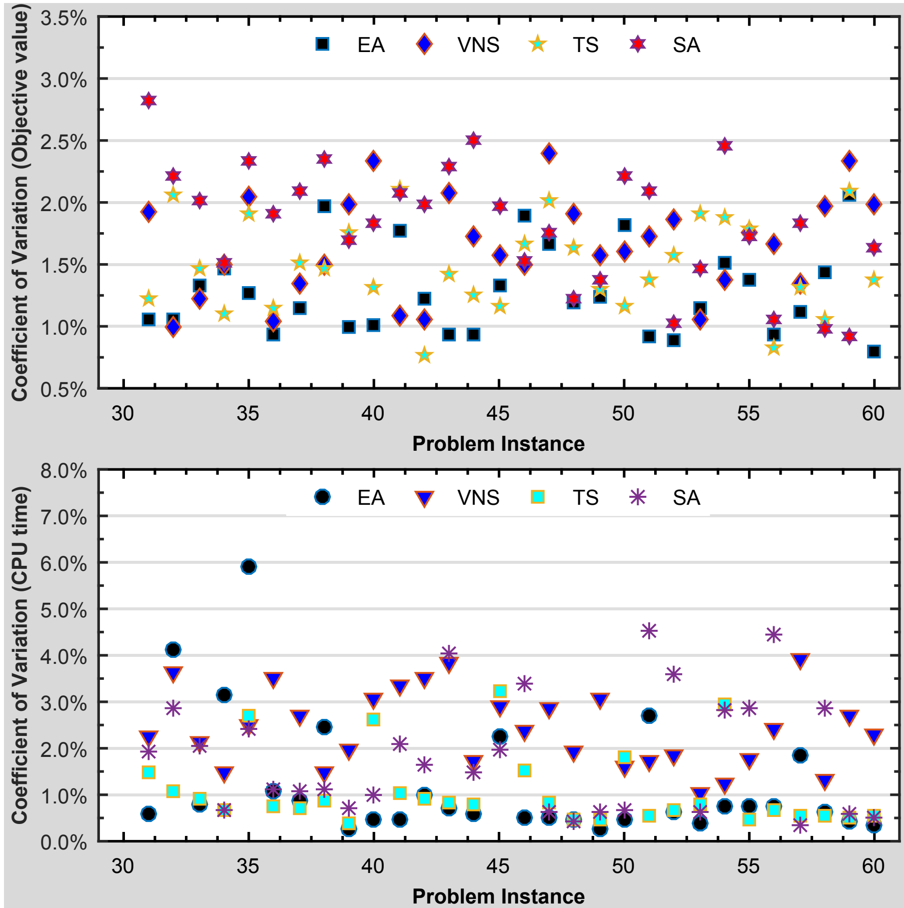

Fig. 7. The coefficient of variation of the objective function and CPU time values for the considered metaheuristics.

critical step in evaluating the efficiency of each metaheuristic. In particular, the convergence pattern analysis allows determining how efficiently each metaheuristic discovers the promising search space domains and superior solutions from one generation (or iteration) to another. As a part of the numerical experiments, the convergence patterns were thoroughly analyzed for every metaheuristic considered and every large-scale problem instance generated. The results from the conducted analysis are reported in Fig. 8 for large-scale problem instances 51 through 60. The selected problem instances represent a portion of the large-scale problem instances with the largest number of CDT doors and trucks. However, similar tendencies were noticed for the remainder of the problem instances.

The analysis results demonstrated that EA was able to achieve good-quality solutions more quickly when comparing to VNS, TS, and SA. Such a performance of EA can be explicated by the fact that EA is a population-based metaheuristic and has a high capability of exploring the solution search space for promising domains and good-quality solutions. On the contrary, the VNS, TS and SA metaheuristics are single-solution-based metaheuristics that primarily rely on local search and are limited to a certain extent in their explorative and exploitative capabilities. Nevertheless, TS was able to show quite a competitive performance for a number of large-scale problem instances (i.e., problem instances 56 through 60). Hence, introduction of the Tabu list in TS was found to be favorable for the search process, as the additional restrictions were imposed for revisiting the same solutions. Such restrictions prompted TS discovering new promising search space domains rather than evaluating the same solutions near local optima.

6.4. Managerial implications

As a part of the numerical experiments, a set of supplementary sensitivity analyses were performed to draw some managerial implications using the proposed **TSPCDT** mathematical model and the most promising metaheuristic. Based on a set of detailed comparative analyses that were performed in Sections 6.3.1, 6.3.2 and 6.3.3 of the manuscript, the developed EA was found to be the most promising metaheuristic as it demonstrated the best tradeoff between the solution quality at convergence and the required CPU time. Hence, the developed EA will be further deployed to conduct the sensitivity analyses and draw some managerial implications. In particular, the sensitivity of the truck scheduling decisions to the following attributes will be analyzed: (a) the product decay rate; (b) the unit product temporary storage cost; and (c) the CDT door availability and truck arrival patterns. The next sections of this manuscript elaborate more on each one of the conducted sensitivity analyses.

6.4.1. Sensitivity of the truck scheduling decisions to the product decay rate

A total of ten product decay rate scenarios were created by increasing the base product decay rate value, which was estimated using uniform distribution $UD[0.010; 0.030]$, by 20% from one scenario to another (i.e., the product decay rate in scenario “1” was generated as $UD[0.010; 0.030]$, while the product decay rate in scenario “10” was generated as $UD[0.052; 0.155]$). The remaining **TSPCDT** parameters were assumed to be the same as specified in Section 6.1.1. The results from the conducted sensitivity analysis are reported in Fig. 9, which shows the changes in the total truck waiting time, the total product storage time, as well as the total truck delayed departure time for the considered product decay rate scenarios.

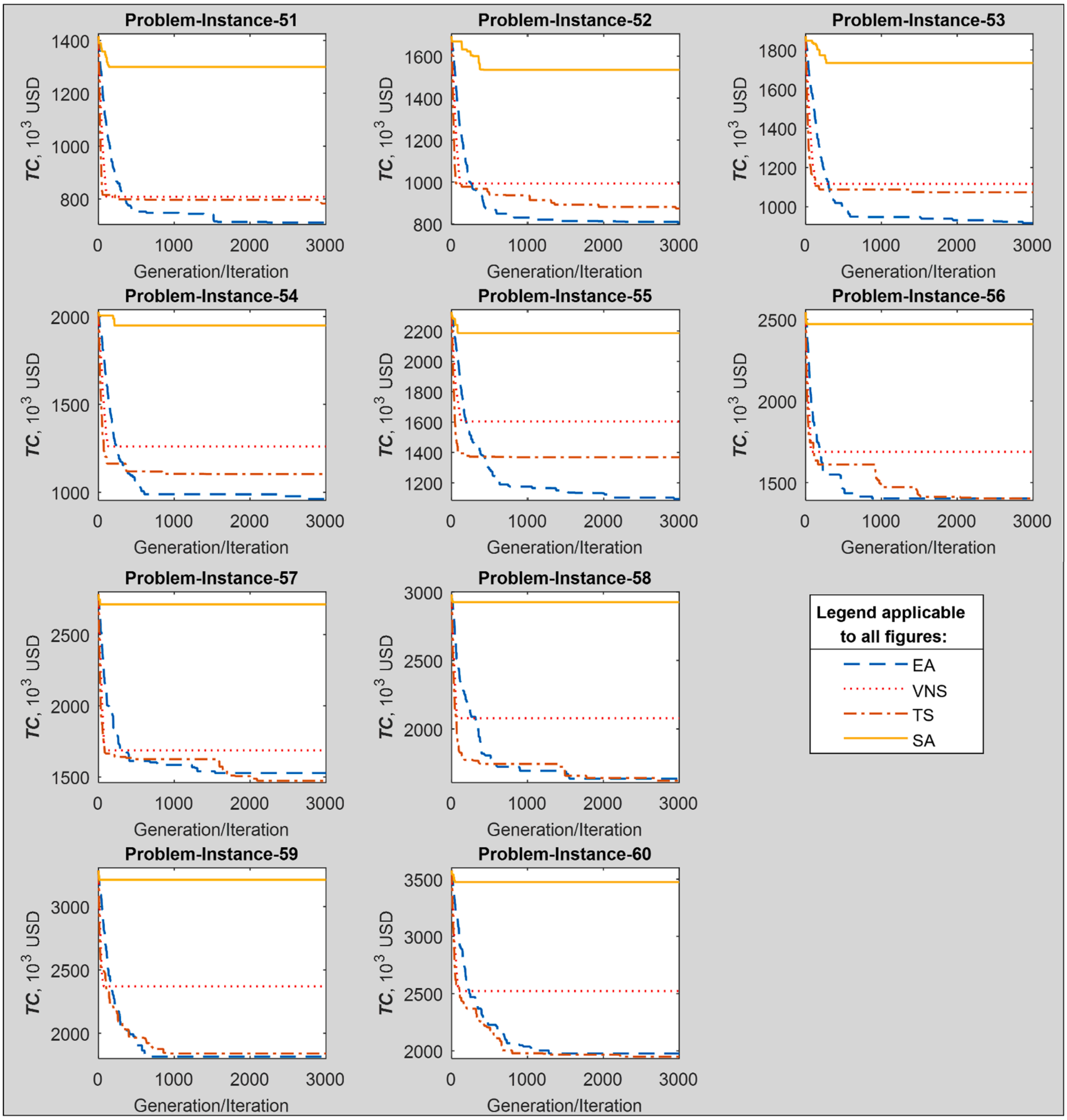

Fig. 8. The convergence patterns for the considered metaheuristics (problem instances 51–60).

It can be observed that an increase in the product decay rate caused a reduction in the total truck waiting time as well as the total truck delayed departure time, aiming to load the arriving outbound trucks with perishable products in a timely manner, so that they could deliver these products to the designated end customers. However, in certain cases, the perishable products had to be stored in temporary storage areas for a longer time period (see Fig. 9) due to the differences in the service start times of the inbound trucks and the corresponding outbound trucks. Nevertheless, since this study modeled temperature-controlled temporary storage areas for perishable products, the total product decay was not substantially affected by increasing the total product storage time. Furthermore, an increase in the product decay rate caused an increase in the total cost, associated with the service of all the arriving trucks at the cold-chain CDT (i.e., the **TSPCDT** objective), from 1972.21 $\cdot$ 10^3 USD in scenario “1” to 2085.51 $\cdot$ 10^3 USD in scenario “10”.

6.4.2. *Sensitivity of the truck scheduling decisions to the unit product temporary storage cost*

A total of ten unit product temporary storage cost scenarios were created by increasing the base unit product temporary storage cost value, which was estimated using uniform distribution $UD[25.66; 51.32]$, by 10% from one scenario to another (i.e., the unit product temporary storage cost in scenario “1” was generated as $UD[25.66; 51.32]$, while the unit product temporary storage cost in scenario “10” was generated as $UD[60.50; 121.00]$). The remaining **TSPCDT** parameters were assumed to be the same as specified in Section 6.1.1. The results from the conducted sensitivity analysis are reported in Fig. 10, which shows the changes in the total truck waiting time, the total product storage time, as well as the total truck delayed departure time for the considered unit product temporary storage cost scenarios.

It can be observed that an increase in the unit product temporary storage cost caused a reduction in the total product storage time, aiming to decrease the total product inventory cost. In the meantime, the arriving outbound trucks had to wait for the service start of the

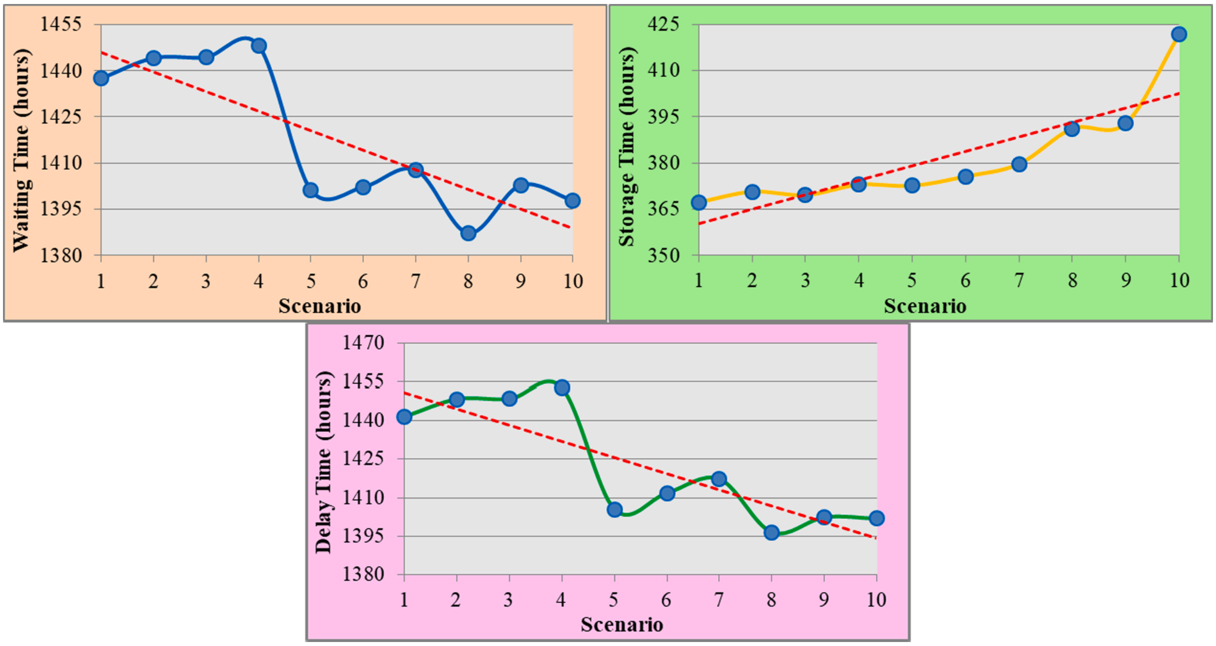

Fig. 9. Sensitivity of the total truck waiting time, total product storage time, and total truck delayed departure time to the product decay rate.

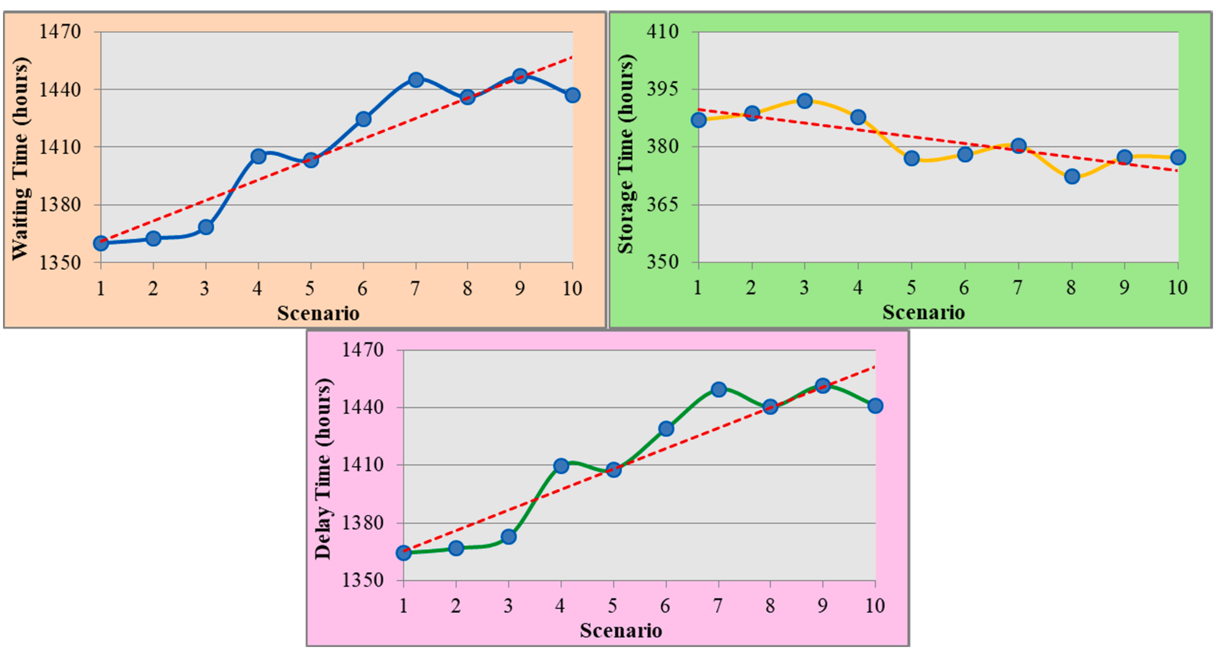

Fig. 10. Sensitivity of the total truck waiting time, total product storage time, and total truck delayed departure time to the unit product temporary storage cost.

corresponding inbound trucks, so that the delivered perishable products could be directly transferred from the inbound doors to the outbound doors with no (or minimum) temporary storage. The latter generally caused an increase in the total truck waiting time along with the total truck delayed departure time (see Fig. 10) from one scenario to another. Furthermore, an increase in the unit product temporary storage cost caused an increase in the total cost, associated with the service of all the arriving trucks at the cold-chain CDT (i.e., the **TSPCDT** objective), from $1484.41 \cdot 10^3$ USD in scenario "1" to $2204.59 \cdot 10^3$ USD in scenario "10".

6.4.3. Sensitivity of the truck scheduling decisions to the CDT door availability and truck arrival patterns

As a part of the numerical experiments, the developed EA was executed for different large-scale problem instances, where the number of CDT doors ranged between 4 and 8 (an increment of 2 CDT doors was adopted) and the number of trucks ranged between 84 and 120 (an increment of 4 trucks was adopted). The CDT door availability (i.e., capacity of the considered cold-chain CDT) and the truck arrival patterns directly influence the truck scheduling decisions. Fig. 11 shows the changes in the total truck waiting time, the total product storage time, as well as the total truck delayed departure time for the considered CDT door availability and truck arrival pattern scenarios. It can be observed that the total truck waiting time could be reduced from 3074.6 h to

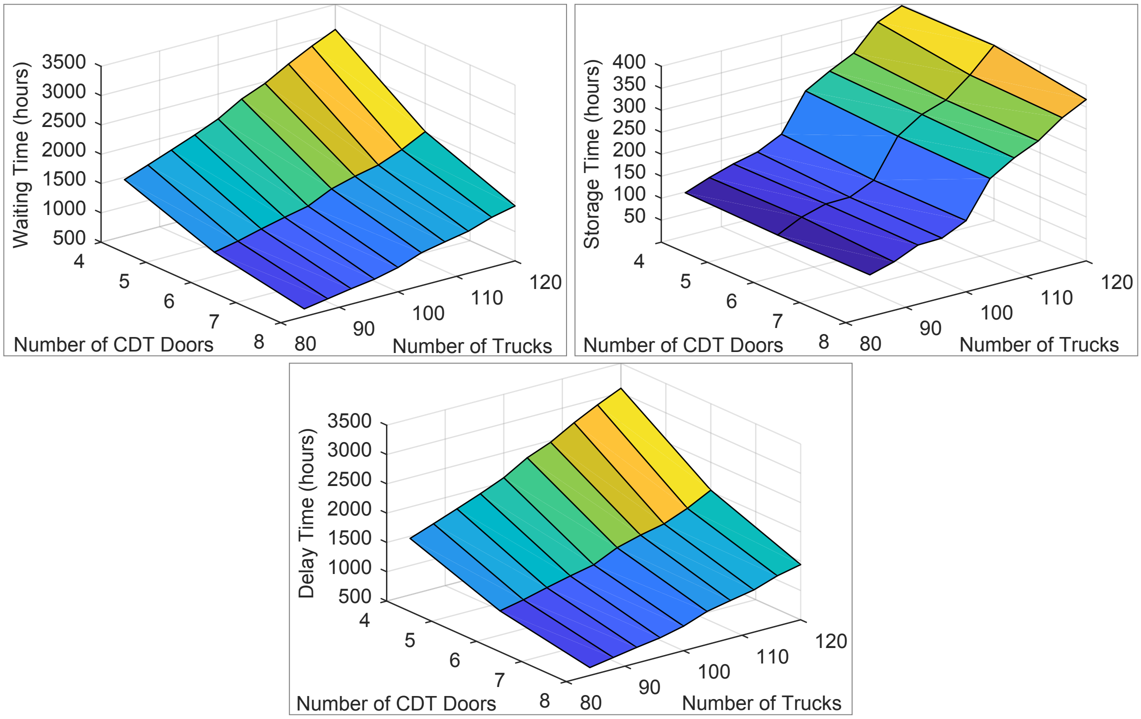

Fig. 11. Sensitivity of the total truck waiting time, total product storage time, and total truck delayed departure time to the CDT door availability and truck arrival patterns.

1437.4 h by increasing the CDT door availability from 4 doors to 8 doors for the case with 120 trucks arriving for service at the considered cold-chain CDT. Similarly, the total truck delayed departure time could be reduced by more than 45% after increasing the CDT door availability from 4 doors to 8 doors for the case with 120 trucks arriving for service at the considered cold-chain CDT. The total product storage time was mostly affected by the number of arriving trucks (see Fig. 11). Furthermore, an increase in the CDT door availability from 4 doors to 8 doors caused a reduction in the total cost, associated with the service of 120 arriving trucks at the cold-chain CDT (i.e., the **TSPCDT** objective), from $3108.67 \cdot 10^3$ USD to $1972.21 \cdot 10^3$ USD.

7. Conclusions and future research needs

The supply chain operations directly depend on the efficiency of the product distribution. Many supply chains handle perishable products that decay over time due to fluctuations in temperature, humidity, and pressure throughout the product distribution process. As a result of mismanagement of supply chain operations, a significant portion of perishable products is wasted, resulting in substantial monetary losses. Cross-docking terminals (CDTs) have been widely used in cold supply chains for the product distribution but have not received adequate attention in the scientific literature. In order to improve the effectiveness of the perishable product distribution, this study proposed a novel mixed-integer mathematical formulation for the truck scheduling problem at a cold-chain CDT. The model explicitly captured the decay of perishable products throughout the service of arriving trucks and accounted for the presence of temperature-controlled storage areas that were specifically designated for perishable products. The objective of the presented model focused on minimization of the total cost incurred during the truck service. Considering the computational complexity of the proposed model, a customized Evolutionary Algorithm (EA) was designed to solve it.

The computational performance of EA was assessed throughout the numerical experiments based on a detailed comparative analysis against the other metaheuristics, including the following: (1) Variable Neighborhood Search (i.e., VNS); (2) Tabu Search (i.e., TS); and (3) Simulated Annealing (i.e., SA). The developed EA was found to be the most promising metaheuristic, considering both solution quality and CPU time perspectives. Furthermore, the proposed EA algorithm demonstrated an acceptable level of stability of the solution quality at convergence and the required CPU time as well. A set of sensitivity analyses, conducted throughout the numerical experiments, revealed that the truck scheduling decisions could be substantially affected by the decay rate of perishable products, the unit temporary storage cost of perishable products, the CDT door availability, and the truck arrival patterns. The proposed mixed-integer programming model, the developed metaheuristic, and the conducted sensitivity analyses would be of potential interest to the supply chain stakeholders that are heavily involved in the distribution of perishable products in cold supply chains and aim to improve their operations.

Throughout this study, a number of simplifying assumptions were made that could be further addressed as a part of the future research. First, uncertainty in the arrival times as well as the handling times of the inbound and outbound trucks could be incorporated in the developed mathematical model. Second, the limitations in the capacity of the temporary storage areas at the considered cold-chain CDT could be accounted for. Third, a multi-objective framework capturing conflicting objectives of the cold-chain CDT operator could be developed (e.g., reduce the total truck delayed departure time vs. reduce the number of forklift operators deployed at the considered cold-chain CDT). Fourth, the developed EA could be compared against some other metaheuristics that were previously used in the CDT truck scheduling literature and other studies (e.g., Differential Evolution, Imperialist Competitive Algorithm, Grey Wolf Optimizer, Particle Swarm Optimization, and Red Deer Algorithm) (Ayough, Zandieh, & Farhadi, 2020; Hussain, Salleh, Cheng, & Shi, 2019). Finally, some additional hybridization techniques could be incorporated within the developed EA (e.g., application of custom local search heuristics after performing the crossover and mutation operations in order to enhance the fitness of the produced and mutated offspring).

CRediT authorship contribution statement

Oluwatosin Theophilus: Conceptualization, Methodology, Data curation, Investigation, Writing - original draft. **Maxim A. Dulebenets:** Conceptualization, Methodology, Data curation, Visualization, Investigation, Writing - original draft, Writing - review & editing. **Junayed Pasha:** Methodology, Data curation, Investigation, Writing - original draft. **Yui-yip Lau:** Methodology, Data curation, Investigation, Writing - review & editing. **Amir M. Fathollahi-Fard:** Methodology, Data curation, Investigation, Writing - review & editing. **Arash Mazaheri:** Methodology, Data curation, Investigation, Writing - review & editing.

Acknowledgment

This study has been supported by the grant CMMI-1901109 administered by the National Science Foundation of the United States. The findings, opinions, and conclusions that can be found in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Appendix A. Full list of abbreviations that were used in the manuscript

AUGMECON2	Augmented Epsilon-Constraint Method
CDT	Cross-Docking Terminal
DE	Differential Evolution
DP	Dynamic Programming
EA	Evolutionary Algorithm
EMS	Express Mail Service
FedEx	Federal Express
ICA	Imperialist Competitive Algorithm
MODE	Multi-Objective Differential Evolution
MOGWO	Multi-Objective Grey Wolf Optimizer
MOICA	Multi-Objective Imperialist Competitive Algorithm
MRDA	Modified Red Deer Algorithm
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
PSO	Particle Swarm Optimization
RDA	Red Deer Algorithm
RFID	Radio-Frequency Identification
SA	Simulated Annealing
SEO	Social Engineering Optimizer
TOPSIS	Technique of Order Preference Similarity to the Ideal Solution
TSPCDT	CDT Truck Scheduling Problem with Product Perishability Considerations
TSPCDTL	Linearized CDT Truck Scheduling Problem with Product Perishability Considerations
UPS	United Parcel Service

References

Abad, H., Vahdani, B., Sharifi, M., & Etebari, F. (2019). A multi-objective optimization model for split pollution routing problem with controlled indoor activities in cross docking under uncertainty. *Journal of Quality Engineering and Production Optimization, 4*(1), 99–126.

Agustina, D., Lee, C. K. M., & Piplani, R. (2010). A review: Mathematical modles for cross docking planning. *International Journal of Engineering Business Management, 2*(2), 47–54.

Agustina, D., Lee, C. K. M., & Piplani, R. (2014). Vehicle scheduling and routing at a cross docking center for food supply chains. *International Journal of Production Economics, 152*, 29–41.

Ahkamiraad, A., & Wang, Y. (2018). Capacitated and multiple cross-docked vehicle routing problem with pickup, delivery, and time windows. *Computers & Industrial Engineering, 119*, 76–84.

Ahumada, O., & Villalobos, J. R. (2011). A tactical model for planning the production and distribution of fresh produce. *Annals of Operations Research, 190*(1), 339–358.

Ardakani, A., Fei, J., & Beldar, P. (2020). Truck-to-door sequencing in multi-door cross-docking system with dock repeat truck holding pattern. *International Journal of Industrial Engineering Computations, 11*, 201–220.

Ayough, A., Zandieh, M., & Farhadi, F. (2020). Balancing, sequencing, and job rotation scheduling of a U-shaped lean cell with dynamic operator performance. *Computers & Industrial Engineering, 143*, Article 106363.

Baniamerian, A., Bashiri, M., & Zabihi, F. (2018). Two phase genetic algorithm for vehicle routing and scheduling problem with cross-docking and time windows considering customer satisfaction. *Journal of Industrial Engineering International, 14*(1), 15–30.

Bilgen, B., & Çelebi, Y. (2013). Integrated production scheduling and distribution planning in dairy supply chain by hybrid modelling. *Annals of Operations Research, 211*(1), 55–82.

Boloori Arabani, A. B., Zandieh, M., & Ghomi, S. F. (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.

Corsten, H., Becker, F., & Salewski, H. (2019). Integrating truck and workforce scheduling in a cross-dock: Analysis of different workforce coordination policies. *Journal of Business Economics*, 1–31.

Darwin, C. (1859). *On the origin of species by means of natural selection or the preservation of favoured races in the struggle for life*. H. Milford: Oxford University Press.

Dulebenets, M. A., & Ozguven, E. E. (2017). Vessel scheduling in liner shipping: Modeling transport of perishable assets. *International Journal of Production Economics, 184*, 141–156.

Dulebenets, M. A. (2018a). A comprehensive evaluation of weak and strong mutation mechanisms in evolutionary algorithms for truck scheduling at cross-docking terminals. *IEEE Access, 6*, 65635–65650.

Dulebenets, M. A. (2018b). A diploid evolutionary algorithm for sustainable truck scheduling at a cross-docking facility. *Sustainability, 10*(5), 1333.

Dulebenets, M. A. (2019a). 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.

Dulebenets, M. A. (2019b). Minimizing the total liner shipping route service costs via application of an efficient collaborative agreement. *IEEE Transactions on Intelligent Transportation Systems, 20*(1), 123–136.

Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing*. Springer International Publishing.

Fathollahi-Fard, A. M., Ahmadi, A., & Sajadieh, M. S. (2020). An efficient modified red deer algorithm to solve a truck scheduling problem considering time windows and deadline for trucks' departure. *Evolutionary Computation in Scheduling*, 137–167.

Fathollahi-Fard, A. M., Ranjbar-Bourani, M., Cheikhrouhou, N., & Hajiaghaei-Keshteli, M. (2019). Novel modifications of social engineering optimizer to solve a truck scheduling problem in a cross-docking system. *Computers & Industrial Engineering, 137*, Article 106103.

FedEx (2016). FedEx cold chain center opens in Memphis. Accessed on 12/20/2020 from https://newsroom.fedex.com/newsroom/fedex-cold-chain-center-opens-memphis/.

Felfel, H., Ayadi, O., & Masmoudi, F. (2016). Multi-objective stochastic multi-site supply chain planning under demand uncertainty considering downside risk. *Computers & Industrial Engineering, 102*, 268–279.

Grunow, M., & Piramuthu, S. (2013). RFID in highly perishable food supply chains–Remaining shelf life to supplant expiry date? *International Journal of Production Economics, 146*(2), 717–727.

Hansen, P., & Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European Journal of Operational Research, 130*(3), 449–467.

Heidari, F., Zegordi, S. H., & Tavakkoli-Moghaddam, R. (2018). Modeling truck scheduling problem at a cross-dock facility through a bi-objective bi-level optimization approach. *Journal of Intelligent Manufacturing, 29*(5), 1155–1170.

Hussain, K., Salleh, M. N. M., Cheng, S., & Shi, Y. (2019). Metaheuristic research: A comprehensive survey. *Artificial Intelligence Review, 52*(4), 2191–2233.

Kavoosi, M., Dulebenets, M. A., Abioye, O., Pasha, J., Theophilus, O., Wang, H., … Mikijeljević, M. (2020). Berth scheduling at marine container terminals: A universal island-based metaheuristic approach. *Maritime Business Review, 5*(1), 30–66.

Khorshidian, H., Shirazi, M. A., & Ghomi, S. F. (2019). An intelligent truck scheduling and transportation planning optimization model for product portfolio in a cross-dock. *Journal of Intelligent Manufacturing, 30*(1), 163–184.

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

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.

Luo, H., Yang, X., & Wang, K. (2019). Synchronized scheduling of make to order plant and cross-docking warehouse. *Computers & Industrial Engineering, 138*, Article 106108.

Margolis, J. T., Sullivan, K. M., Mason, S. J., & Magagnotti, M. (2018). A multi-objective optimization model for designing resilient supply chain networks. *International Journal of Production Economics, 204*, 174–185.

Mena, C., Terry, L. A., Williams, A., & Ellram, L. (2014). Causes of waste across multi-tier supply networks: Cases in the UK food sector. *International Journal of Production Economics, 152*, 144–158.

Miao, Z., Lim, A., & Ma, H. (2009). Truck dock assignment problem with operational time constraint within cross-docks. *European Journal of Operational Research, 192*(1), 105–115.

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.

Nasiri, M. M., Rahbari, A., Werner, F., & Karimi, R. (2018). Incorporating supplier selection and order allocation into the vehicle routing and multi-cross-dock scheduling problem. *International Journal of Production Research, 56*(19), 6527–6552.

Nogueira, T. H., Coutinho, F. P., Ribeiro, R. P., & Ravetti, M. G. (2020). Parallel-machine scheduling methodology for a multi-dock truck sequencing problem in a cross-docking center. *Computers & Industrial Engineering*, Article 106391.

Pasha, J., Dulebenets, M. A., Kavoosi, M., Abioye, O. F., Theophilus, O., Wang, H., … Guo, W. (2020). Holistic tactical-level planning in liner shipping: An exact optimization approach. *Journal of Shipping and Trade, 5*, 1–35.

Pinedo, M. (2016). *Scheduling: Theory, algorithms, and systems*. Springer International Publishing.

Piramuthu, S., & Zhou, W. (2013). RFID and perishable inventory management with shelf-space and freshness dependent demand. *International Journal of Production Economics, 144*(2), 635–640.

Piramuthu, S., Farahani, P., & Grunow, M. (2013). RFID-generated traceability for contaminated product recall in perishable food supply networks. *European Journal of Operational Research, 225*(2), 253–262.

Rahbari, A., Nasiri, M. M., Werner, F., Musavi, M., & Jolai, F. (2019). The vehicle routing and scheduling problem with cross-docking for perishable products under uncertainty: Two robust bi-objective models. *Applied Mathematical Modelling, 70*, 605–625.

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, 278*(3), 752–771.

Rong, A., Akkerman, R., & Grunow, M. (2011). An optimization approach for managing fresh food quality throughout the supply chain. *International Journal of Production Economics, 131*(1), 421–429.

Shahmardan, A., & Sajadieh, M. S. (2020). Truck scheduling in a multi-door cross-docking center with partial unloading–Reinforcement learning-based simulated annealing approaches. *Computers & Industrial Engineering, 139*, Article 106134.

Shuib, A., & Fatthi, W. N. A. W. A. (2012). A review on quantitative approaches for dock door assignment in cross-docking. *International Journal on Advanced Science, Engineering and Information Technology, 2*(5), 370–374.

Sreedevi, R., & Saranga, H. (2017). Uncertainty and supply chain risk: The moderating role of supply chain flexibility in risk mitigation. *International Journal of Production Economics, 193*, 332–342.

Tadumadze, G., Boysen, N., Emde, S., & Weidinger, F. (2019). Integrated truck and workforce scheduling to accelerate the unloading of trucks. *European Journal of Operational Research, 278*(1), 343–362.

Theophilus, O., Dulebenets, M. A., Pasha, J., Abioye, O. F., & Kavoosi, M. (2019). Truck scheduling at cross-docking terminals: A follow-up state-of-the-art review. *Sustainability, 11*(19), 5245.

UN Environment (2020). Worldwide food waste. Accessed on 05/22/2020 from https://www.unenvironment.org/thinkeatsave/get-informed/worldwide-food-waste.

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

Wang, S., Meng, Q., & Liu, Z. (2013). Bunker consumption optimization methods in shipping: A critical review and extensions. *Transportation Research Part E: Logistics and Transportation Review, 53*, 49–62.

Wang, X., & Li, D. (2012). A dynamic product quality evaluation based pricing model for perishable food supply chains. *Omega, 40*(6), 906–917.

Wisittipanich, W., Irohara, T., & Hengmeechai, P. (2019). Truck scheduling problems in the cross docking network. *International Journal of Logistics Systems and Management, 33*(3), 420–439.

Yu, M., & Nagurney, A. (2013). Competitive food supply chain networks with application to fresh produce. *European Journal of Operational Research, 224*(2), 273–282.