FLORIDA STATE UNIVERSITY

FAMU-FSU COLLEGE OF ENGINEERING

MODELS AND SOLUTION ALGORITHMS FOR EFFICIENT TRUCK SCHEDULING AT CROSS-DOCKING TERMINALS WITH TIME-SENSITIVE PRODUCTS

By

OLUWATOSIN THEOPHILUS

A Dissertation submitted to the
Department of Civil and Environmental Engineering
in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

2021

Oluwatosin Theophilus defended this dissertation on February 10, 2021

The members of the supervisory committee were:

Maxim A. Dulebenets
Professor Directing Dissertation

John O. Sobanjo
Committee Member

Juyeong Choi
Committee Member

Yanshuo Sun
Committee Member

Samuel Awoniyi
University Representative

The Graduate School has verified and approved the above-named committee members and certifies that the thesis has been approved in accordance with University requirements.

To the Lord God Almighty and to my lovely family.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor, Dr. Maxim A. Dulebenets, for his kind support, instructions, timely feedbacks, guidance, corrections, and encouragements. Without his invaluable involvement, and tremendous contributions to this research, it would have been impossible to reach the set goals. Thank you again, Dr. Maxim.

I would like to appreciate the efforts of my supervisory committee members, Dr. John O. Sobanjo, Dr. Juyeong Choi, Dr. Yanshuo Sun, and Dr. Samuel Awoniyi, for the critical review and helpful suggestions they provided. The suggestions of each committee member have helped improve the dissertation in no small magnitude. Also, I am elated to have had the opportunity of working with a group of highly smart colleagues, Dr. Olumide F. Abioye, Dr. Masoud Kavoosi, and Dr. Junayed Pasha, in the course of completing this dissertation.

Lastly, I am sincerely grateful to my family for their love, encouragements, and emotional support, without which I would not have been able come this far.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

The product distribution strategy deployed in a given supply chain is highly important for the proper operations of that supply chain. A number of supply chain problems are associated with the mismanagement of supply chain operations (for example, the decay of products that are perishable in nature). Many supply chain problems eventually result in monetary losses. The cross-docking strategy (i.e., use of cross-docking terminals (CDTs)) has been found to improve product distribution in many supply chains. The cross-docking strategy refers to the logistics process, which consists of the reception of inbound products that have been transported by trucks/trailers, unloading and transferring the products across the dock to outbound doors (with or without temporary storage), where the outbound trucks/trailers are docked to pick up the consolidated products, which are ready to be shipped out. This process enhances the efficiency of the supply chain operations, reduces handling times and operational costs using various handling equipment types (such as sorters, combiners, conveyor belts, and others).

A detailed survey of previous studies on CDTs showed that cross-docking is gaining attention increasingly over the years. Several supply chains have also adopted the cross-docking strategy in their distribution chains. Specifically, in practice, the CDT strategy is found suitable for product distribution in cold supply chains. Most of the surveyed studies focused on the CDT operations without considering the possibility of the perishable products decaying during the handling process. Other studies were also reviewed which considered the decay of products during various supply chain processes but not within the context of cross-docking operations. Therefore, there are no studies conducted to date that explicitly capture the perishable products decay within the context of supply chains and explicitly model the cross-docking strategy.

This dissertation focuses on the scheduling of trucks (both in- and outbound trucks) at a CDT, where some of the delivered products are perishable in nature. The short lifespan of perishable products (i.e., foods and drugs) poses critical challenges to the CDT operations management. Perishable goods are time-sensitive products that require minimal handling time to preserve their quality and profitability. Cross-docking is expected to facilitate the distribution of perishable products within supply chains. There are many challenges involved in the management of the

cross-docking terminals with perishable products, including determination of the service order of the trucks (inbound and outbound) carrying perishable products, selection of preemption strategies for certain trucks (i.e., a given truck can leave the door, so another truck can be docked for service), allocation of suitable temporary storage space for products, quality loss due to late delivery or errors in temperature control.

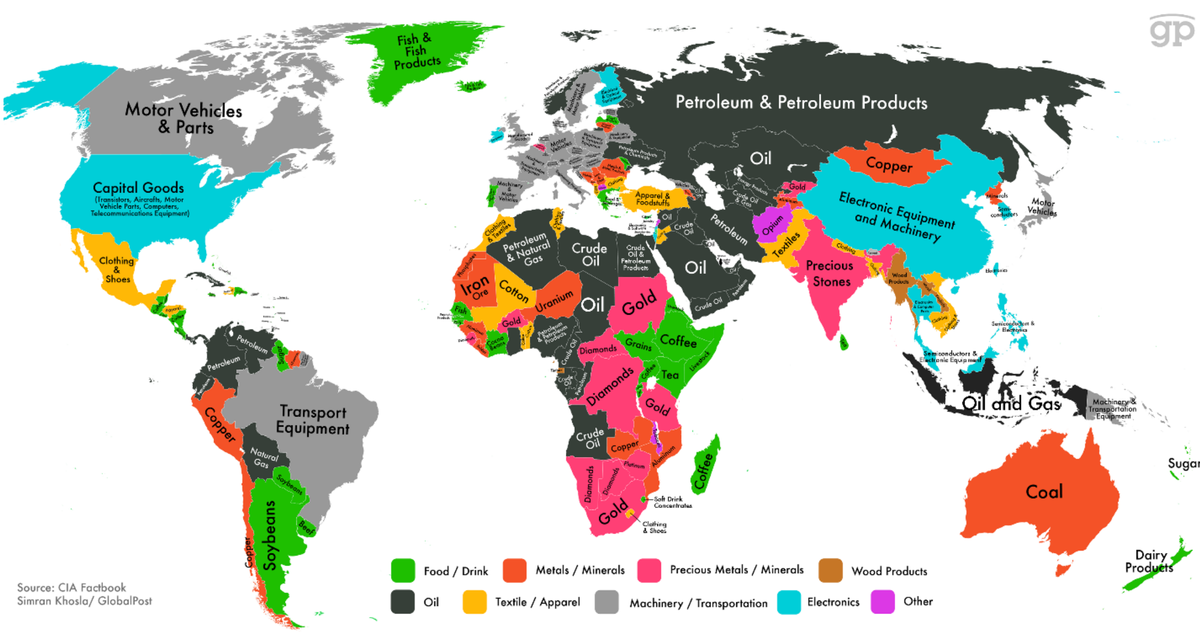
This dissertation aims to develop a mathematical model for scheduling the arriving trucks at a cross-dock terminal, taking product decay into consideration throughout the handling process. The objective of the mathematical model minimizes the total truck service cost, which includes (1) waiting cost; (2) service cost; (3) cost of product storage; (4) cost of delay in truck departure; and (5) the cost associated with the decay of products that are perishable in nature. A number of linearization techniques are discussed in order to linearize the original nonlinear mathematical model (where the nonlinearity is caused by the adopted product decay function). The complexity of the linearized model is evaluated in this dissertation. Moreover, the candidate solution approaches for the proposed mathematical model are described.

The developed model was solved using the exact optimization technique. In particular, the model was solved to optimality using CPLEX. However, it was observed that the computational time increased as the problem size increased due to the model complexity. Four alternative solution approaches namely: (1) Evolutionary Algorithm (EA); (2) Variable Neighborhood Search (VNS); (3) Tabu Search (TS); and (4) Simulated Annealing (SA), which are common metaheuristic algorithms, were developed and compared with CPLEX using small-size problem instances. These metaheuristics were able to achieve optimal solutions for the small-size problem instances and required relatively low computational times. The metaheuristic algorithms were further compared, and EA was found to outperform the others (VNS, TS, and SA) based on the balance between the objective function and computational time values. A set of analyses were carried out using EA, and managerial insights that could be of interest to supply chain stakeholders were drawn. The proposed mathematical model, the developed EA, and the managerial insights could assist the CDT manager in making efficient and timely truck scheduling decisions in any planning horizon.

CHAPTER ONE

1. INTRODUCTION

Logistic procedures, taking place in different supply chains across the world, have gained significant attention of researchers over the recent years. Different ways are being sought on how to improve effectiveness of the supply chain operations, minimize the associated costs, and maximize the total profit of relevant companies. Companies all over the world are pursuing application of the latest methodologies in their supply chains in order to reduce costs, increase customer satisfaction, improve utilization of the available assets, and build new revenues (Deveshwar and Rathee, 2010). A spatial distribution of raw materials, finished products, production firms, and consumers all over the world and within continents, countries, and localities is the reason why supply chains exist. The map in Figure 1 shows the location of different export products (e.g., gold, oil, coffee, tea, precious stones, electronic equipment, transport equipment, coal, etc.), which are distributed among different countries across the world. While certain products are readily available in certain parts of the world, the demand for these products may exist in other parts of the world.



Source: Khosla GlobalPost (2014)

Figure 1: World export map.

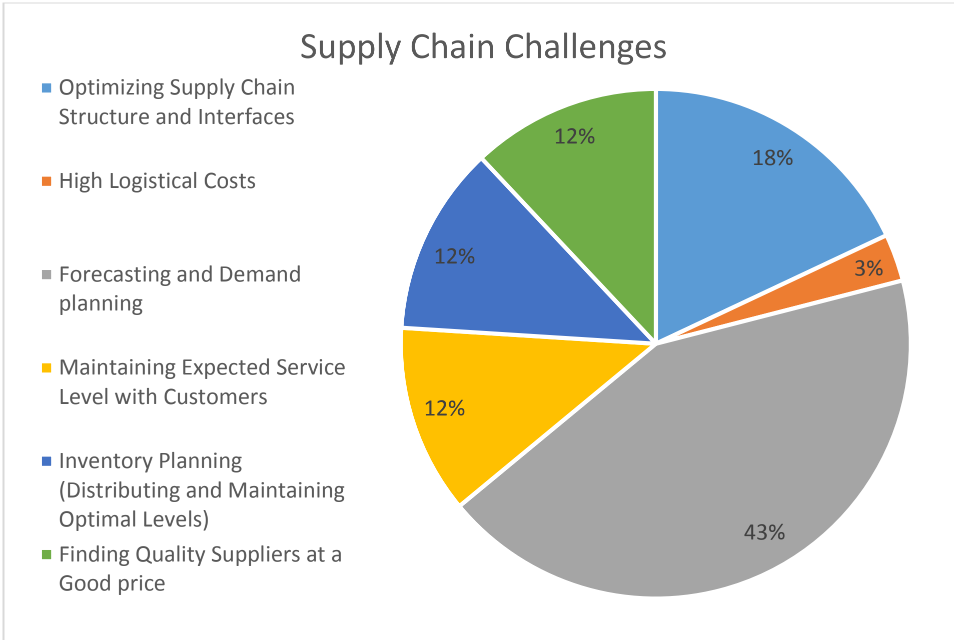
Table 1: Definitions of supply chain/supply chain management.

Author(s)	Year	Definition/Description
Jones and Riley	1985	Supply chain management deals with total flow of materials from suppliers through to the end users.
Cooper et al.	1997	Supply chain management is an integrative philosophy to manage the total flow of a distribution channel from supplier to ultimate users.
Mentzer et al.	2001	Supply chain is a set of organizations directly linked by one or more upstream and downstream flows of products, services, finances, or information from a source to a customer.
Bala	2014	Supply chain describes inter-organizational issues such as: an alternative organizational form to vertical integration; the relationship that a company develops with its suppliers and addressing the purchasing and supply perspective.
Patil	2015	Supply chain management is also defined as managing chain of events that strive to balance activities such as promotion, sales distribution, and production, the overall goal is how to achieve maximum customer satisfaction.
Kozlenkova et al.	2016	Supply chain is a system of organizations, people, activities, information and resources involved in moving a product or service from supplier to customer.
Jayant	2016	Supply chain is the collection of resources required to make and deliver a product to the final customer.

The term "supply chain management" was first coined by Oliver and Webber (1982), where they defined supply chain management as the process of planning, implementing, and controlling the operations within a given supply chain with the key objective of satisfying customer's requirements as efficiently as possible. Over the years, new definitions of the term "supply chain management" have been proposed. Table 1 shows various definitions of the term "supply chain management", as presented in publications by several authors between 1985 and 2016.

The Council of Supply Chain Management Professionals (CSCMP, 2013) defines supply-chain management in a more comprehensive manner as follows: “supply chain management encompasses the planning and management of all activities involved in sourcing and procurement, conversion, and all logistics management activities”. Importantly, supply chain management includes coordination and collaboration among various channel partners, which can include suppliers, third-party service providers, intermediaries, and final customers. Supply and demand management across different companies is integrated by supply chain management. Moreover, the primary responsibility of supply chains is to link major business processes and business functions across different companies and develop a high-performing cohesive business model (CSCMP, 2013). Supply chains are heavily dependent on all the logistics management activities, manufacturing operations, and a wide range of activities, associated with sales, marketing, finance, product design, and information technology. According to Jayant (2016), some of the reasons why supply chain management is currently important include: (1) heightened competition with respect to productivity, quality, and flexibility/responsiveness; (2) integration instead of managing individual pieces. Efficiency in a supply chain is crucial at both organizational levels and on a global scale.

Each supply chain creates the link between a supplier and the ultimate user of a commodity. The creation of a supply chain involves several decisions and planning, such as deciding the means of transporting raw materials from their original location to the production firm, and the distribution of finished products from the production firms to the consumers. These decisions become very complex when all the necessary factors involved are holistically considered. Generally, companies with well-run supply chains outperform the others. Therefore, supply chain management is a key determining factor of a company’s performance. There are several challenges that are faced by different organizations throughout supply chain management of organizations. The findings from the survey of supply chain challenges, conducted by Tefen and the Israeli Supply Chain Management Association (Barak and Gat, 2016), are summarized in Figure 2.

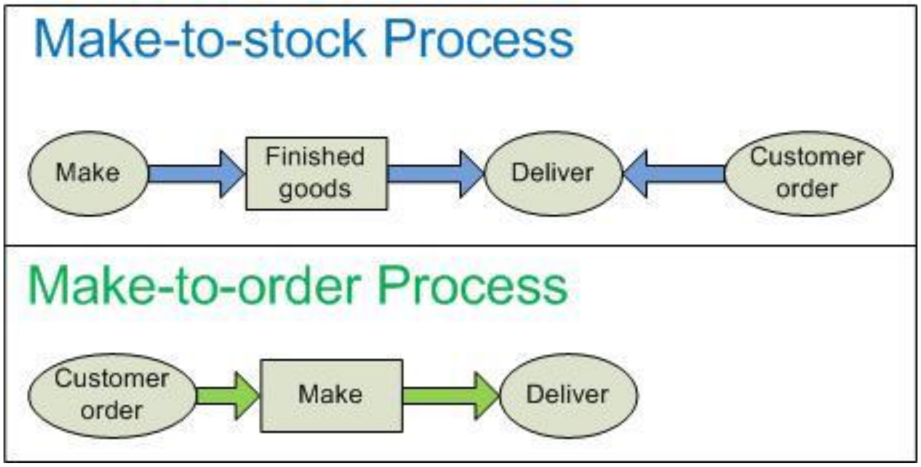


Note: This figures was prepared using the data from Barak and Gat (2016)

Figure 2: Supply chain challenges.

The survey identified six major challenges faced by supply chain management. The supply chain challenges, identified in the previously conducted survey (see Figure 2), include: (1) forecasting and demand planning – this involves the setting of reasonable goals based on the trends in historical data, recorded throughout supply chain operations; (2) optimizing supply chain structure and interfaces – this involves creating a balance between cost and profitability while ensuring seamless flow between the key components of a given supply chain; (3) finding quality suppliers at a the best possible price; (4) inventory planning – this involves effective product distribution and maintaining the optimal or close to the optimal inventory levels; (5) maintaining expected service level with customers; and (6) high logistical costs.

The current issues in supply chain management, as identified by Groznika and Trkman (2012), include: (1) strategic insights; (2) business redesign; (3) supply chain risk management; (4) supply chain framework and standards; (5) supplier/partner relationship management; (6) customer satisfaction; (7) cost minimization; and (8) performance measurement in supply chain management. Supply chain managers, researchers, and different stakeholders are constantly implementing promising alternatives and looking for new ideas to support supply chain operations, address the existing inefficiencies, and resolve a wide range of issues, including the ones that were highlighted above.



Note: This figures was prepared using the data from Skinner (2006)

Figure 3: Supply chain strategies: make-to-stock and make-to-order.

A major challenge in supply chain management is the decision on the strategy to use, i.e. strategic insight. In order to optimize various aspects of a given supply chain, such as operations, channel, outsourcing, and customer service, some approaches rely on a make-to-stock process (where the goods are produced and stored) and a make-to-order process (where the production directly depends on the customer's order). Figure 3 gives a schematic description of the difference between the make-to-stock process and the make-to-order process. An equally important decision is making a choice between push strategies (where each workstation pushes out a completed job to the subsequent station) and pull strategies (where a work station completes a job and then places a request for more jobs from the preceding station) at the production/manufacturing stage.

Business redesign consists in identification of possible inefficiencies in a given supply chain, including, but not limited to, lead times, product quality, service quality, and operational costs. There exist several solution approaches in order to accomplish the latter task. One approach, which is referred to as Business Process Reengineering (BPR), aims to assess, analyze, and restructure the business policies, practices, and procedures based on the findings for improved efficiency, revealed by different organizations. Another approach, which is referred to as

Continuous Process Improvement (CPI), relies on understanding of customer's requirements and process capacity, aiming to intermittently detect the causes of gaps in reliability, process cycle times, quality, and productivity. Some examples of the CPI techniques include six sigma (ensuring that products are delivered on time and in good quality from the beginning to the end of a supply chain) and Total Quality Management (the continuous improvement of internal practices).

The management of supply chains comes with various risks (or uncertainties), which pose a significant challenge. There are several uncertainties that may exist in a given supply chain. Examples include uncertainties in demand for products, supply of raw materials, transportation of products between various nodes of a supply chain, processing times of products at intermediate facilities, and others. The main solution approaches, used in risk management of supply chains, are: (1) determine which risk to take after a detailed risk analysis process; and (2) determine how to mitigate unavoidable risks either with dual or multiple sourcing, change of suppliers or other components of the supply chain. A disruption in the components of a supply chain, if not handled well, may result in a loss of customers and/or investors.

Another challenge, faced in supply chain management, is the development of supply chain framework and standards. This is to ensure smoothness in the interaction of the components of the supply chain. Supply chain managers use the information system support, where the components of the supply chain interact using a common protocol that unifies them. For instance, global manufacturing of air conditioning used SCOR (supply chain operation reference) to link processes, performance metrics, practices and skills to a unified structure (SCOR, 2012). The unified structure includes the suppliers, and it helps in the management of a relationship with them.

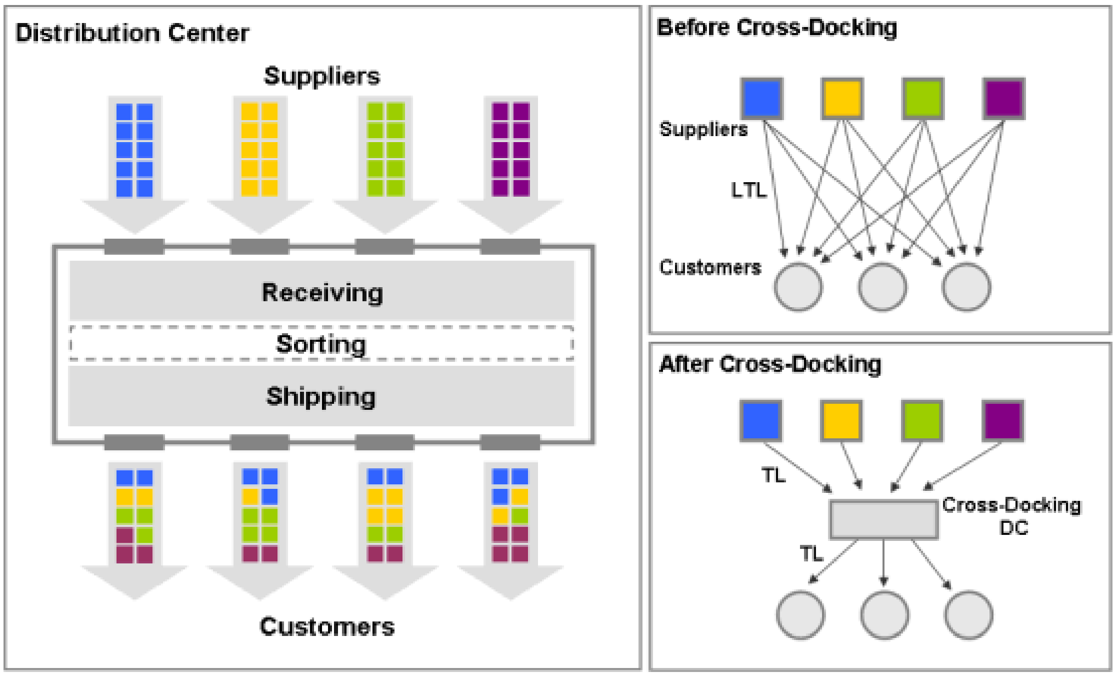
Minimization of the total operating cost is another challenge that supply chain managers are faced with. The total operating cost comprises of various cost components, such as transportation cost, labor cost, storage cost, inventory cost, and others. These cost components directly impact the profitability of a given supply chain. The main approach, used to mitigate the effect of increasing cost components, is the just-in-time (JIT) concept. The JIT concept reduces the operating cost by decreasing or completely eliminating inventory cost, producing goods directly for sale. This approach also proffers solutions to the challenge of customers' satisfaction, as customers must get their goods on time.

Finally, efficient management of a supply chain can only be realized if there is a way to measure the supply chain performance. Developing the most appropriate performance indicators is important. Since the supply chain involves several components, performance indicators can be divided into several groups in order to effectively measure the performance of each component. Once the performance of each component is assessed, the analysis of the whole chain can be conducted. The common metrics that have been developed as performance indicators include the following: (1) lead time – which is the time between order and delivery; (2) on-time delivery – which refers to punctuality of the product delivery (i.e., the products should be delivered at the time specified by the customer); (3) delivery frequency – the number of deliveries within a given period of time; and (4) the quality of delivered goods. A common approach, used in evaluating these measures, is the use of organizational historical data and simulation runs. This method enables the adequate evaluation of numerous scenarios and the outcomes. Such method can also be used to compare various alternative strategies for supply chain management in order to select the most efficient one.

1.1 Definitions of Cross-Docking

The strategy of cross-docking has been widely deployed by various companies in order to enhance the efficiency of their supply chain operations. This section contains some of the definitions of cross-docking, as presented by several authors in the respective research studies. There are slight variations in the definitions; however, there are common cross-docking attributes that have been widely accepted in most of the definitions. Cross-docking is typically defined as a logistic procedure, where the products are transferred from a supplier or a manufacturing plant by the inbound trucks to a cross-docking terminal (CDT), where these products a being deconsolidated, sorted, and then consolidated based on the customer's preferences (Theophilus et al., 2019). The consolidated products are directly moved to a customer or a retail chain by the outbound trucks with marginal to no storage time.

Figure 4 compares a schematic diagram of the distribution chain, where cross-docking is adopted, with the one, where cross-docking is not used. The diagram clearly shows that the sorting of items is a key improvement that cross-docking provides to a distribution chain (i.e., instead of different product types being transported by many trucks to the end customer, much fewer trucks deliver consolidated shipments that contain different product types to the end customer).



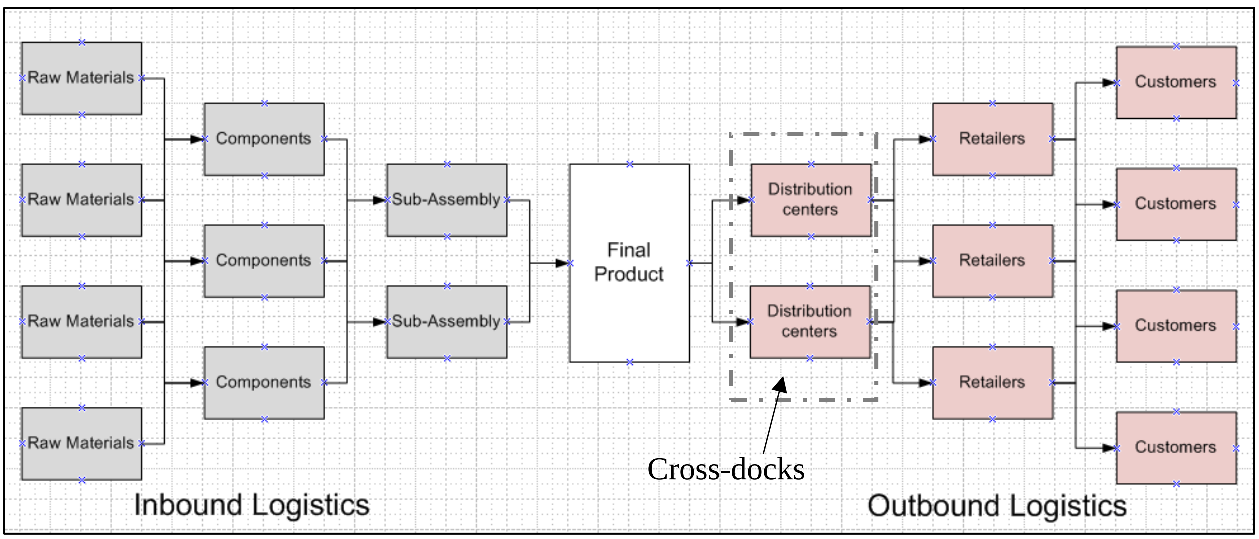
Source: Rob (2012)

Figure 4: Cross-docking for efficient product distribution.

Yu (2002) defines cross-docking as a material handling and distribution concept, based on which different items move directly from a receiving dock to a shipping dock without being stored in a warehouse or alternative distribution center. According to Boysen (2010), the cross-docking concept consists in "receiving product from a manufacturer for several end destinations and consolidating this product with other suppliers' product for common final delivery destinations". Also, according to Material Handling Industry of America (MHIA, 2011), cross-docking is described as "the process of moving merchandise from the receiving dock to shipping dock for shipping without placing it first into storage locations". Cross-docking is viewed as a distribution procedure that attempts to lessen the burdens that come with warehousing by reducing it to a transshipment center, where receiving and shipping are its main functions (Li et al., 2004). Sheikholeslam and Emamian (2016) define a CDT as a type of a distribution center that is dedicated to the transshipment and distribution of truck and container loads.

1.2 Existing Tendencies in Supply Chains: the Need for Cross-Docking

There is a number of existing tendencies in supply chain operations, which have been observed in the past and highlight the necessity of implementing the cross-docking concept. While a supply chain seeks to integrate several organizational units for efficient coordination of the flow of materials and processes, cross-docking provides a more efficient means of the product distribution. Figure 5 illustrates the key supply chain components, which are directly involved throughout the production and logistics operations. Cross-docks (or distribution centers) are used to accommodate the final products before they will be delivered to retailers.

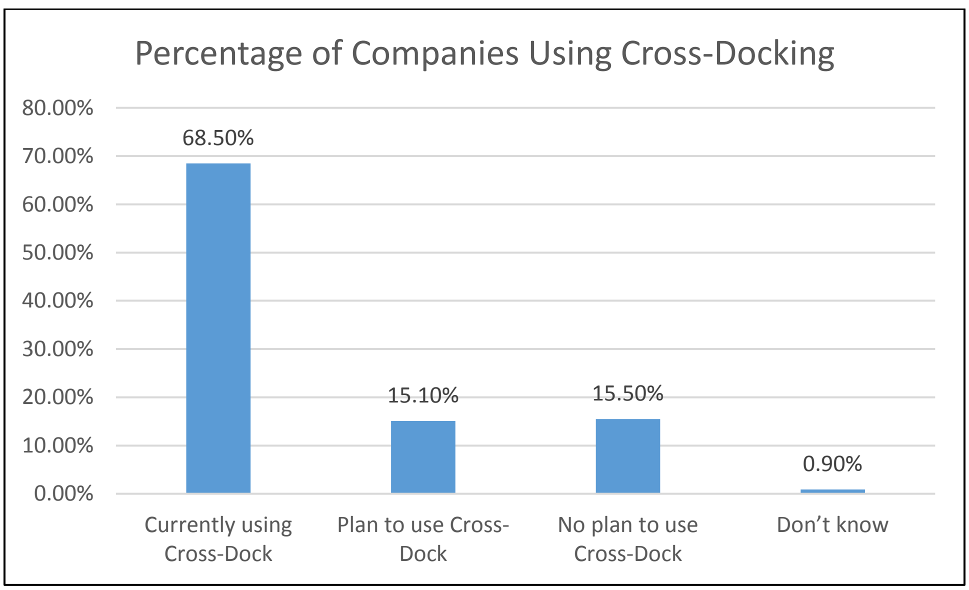


Note: This figures was prepared using the data from Thapa et al. (1970)

Figure 5: Supply chain synchronization in production and logistics.

Supply chains allow encompassing all the activities, which are associated with the transformation and flow of different goods from raw materials through to the end user, along with the associated information flows (Handfield and Nicholas, 1999). The existing trends show that supply chain operations are receiving increased attention due to the globally competitive and challenging environment. One of the recent studies, which was conducted by Dulebenets (2018a), showed that supply chains have also become more time-sensitive. Losses in terms of sales and customer relationship would be incurred as a result of delays in the supply chain operations.

A supply chain, as defined by Christopher (1992), represents a network of organizations that are involved, through upstream and downstream linkages, in a large number of various activities and processes that produce value in the form of services and products that are transported to the end consumer. The supply chain industry, therefore, cuts across multiple firms from supply (which can be considered as an "upstream"), to distribution, customers or end users (which can be considered as a "downstream"). According to Pal and Kant (2017), many customers demand the freshest possible food at the least possible prices; therefore, a simultaneous achievement of high efficiency in the distribution of perishable products that meet the quality requirements remains one of the major challenges in the management of food supply chains.



Note: This figures was prepared using the data from Saddle Creek Corp (2011)

Figure 6: Percentage of companies using cross-docks.

According to Saddle Creek Corp (2011), companies are in a constant search for viable ways to reduce cost, manage inventory, increase the efficiency of their supply chains and ultimately satisfy their customers. Implementation of the cross-docking strategy provides some of these benefits. Figure 6 shows the results of a survey among different companies, which aimed to determine whether the companies used cross-docks as a part of their supply chain or plan to use cross-docks in future. It was found that over 68% of companies, who responded to the conducted survey, used the cross-docking strategy in their supply chains, and more than 15% of companies plan to implement it.

Vogt (2010) examined the roles of cross-docking in different supply chain operations. As a result of an extensive evaluation of supply chains with cross-docks, it was found that cross-docks are an integral part of the supply chains that they serve; and, therefore, cannot be optimized in isolation. Some key questions, which were highlighted in that study, included: (1) where products are identified; (2) where the primary sorting occurs; and (3) whether the supplier provides one or several product types. Cross-docking could fail as a strategy if not appropriately integrated into a suitable supply chain it is meant to serve.

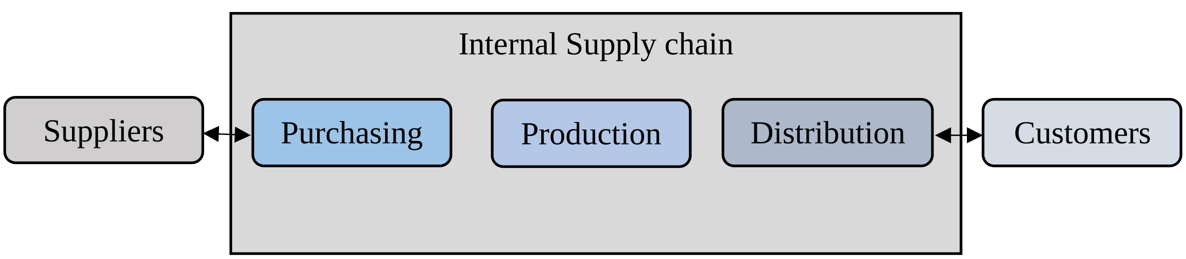
Cross-docking involves delivery of products directly to customers from a manufacturing plant with little or no material handling in between. Moreover, the cross-docking strategy not only decreases material handling but also reduces the need to store the products (unlike to what can be typically observed in warehouses). Van Belle et al. (2012) identified four major functions of warehousing, which are receiving shipments, order picking, storage, and shipping. The costliest of these four operations are storage and order picking. While storage incurs inventory holding cost, order picking incurs labor cost.

Implementation of the cross-docking strategy allows reducing or even completely eliminating the latter two operations. In most cases, the products that are sent to the loading dock from the manufacturing area have been allocated for outbound deliveries; therefore, cross-docking has become an important approach for proper handling of products and packaging in the supply chain. In order to achieve an effective cross-docking, it is necessary to securely arrange certain technologies, including portable and fixed bar code devices, data acquisition terminals, software applications, dedicated databases, networks, or combinations of networks and radio frequency (RF) devices. The aforementioned technologies will facilitate the product flow within cross-docks and ensure that supply will meet the existing demand.

The concept of cross-docking was introduced first to the Walmart's supply chain for the improvement in the efficiency of product distribution (Stalk et al., 1992). Therefore, as products arrived from manufacturers, they were directly loaded onto trucks that would transport them to the Walmart retail stores. This practice, therefore, reduced transportation costs, storage costs, and labor costs. Several other organizations have also adopted the cross-docking technique to improve the efficiency of their supply chain processes. Some examples include Target, COSTCO, FedEx, UPS, USPS, and DHL. The tendencies of using the cross-docking technique in a supply chain are particularly of a great interest in food supply chains (Dulebenets and Ozguven, 2017). Food supply chains have specific considerations for the product quality and change in the product quality, as the products are transported throughout the supply chain (Ahumada and Villalobos, 2009). The issue of food quality is also important, as it directly affects profitability, public health, and safety.

1.3 Importance of Cross-Docking for Supply Chain Operations

Cross-docking, as described by Sheikholeslam and Emamian (2016), can be considered as an integral component of a supply chain. A given supply chain may consist of one or more cross-docking facilities. While a supply chain comprises of the whole chain of distribution from the producers (i.e., manufacturing plants) to the ultimate consumers, cross-docking is a simple transshipment center within that chain. Figure 7 describes the flow of operations within a company's supply chain. The basic operations include purchasing, production, and distribution (cross-docking concept is generally applied at the distribution stage).

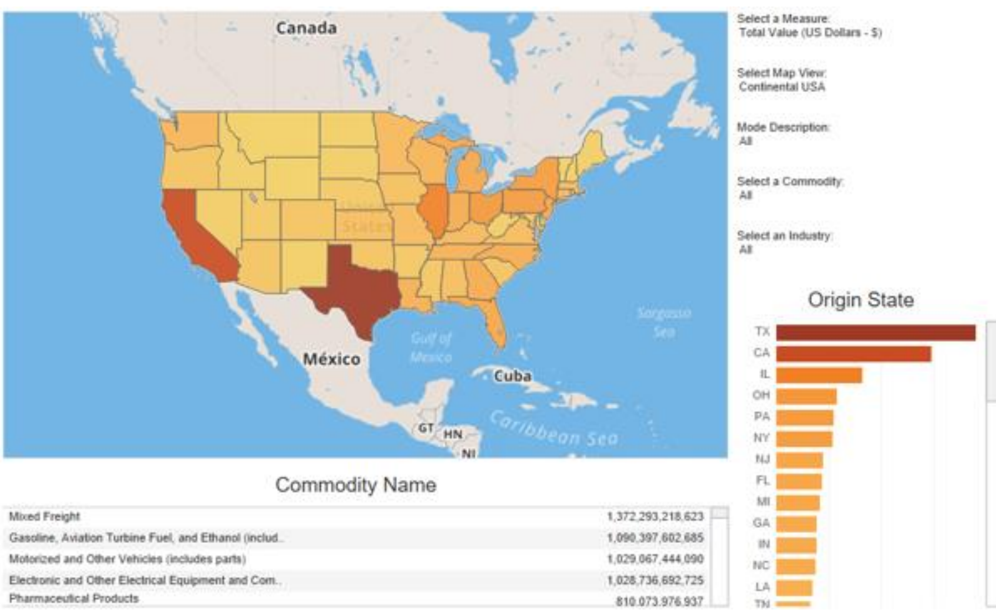


Note: This figures was prepared using the data from Bala (2014)

Figure 7: Company's supply chain.

The distribution aspect of the supply chain consists in the movement of products from the producer to the consumer. According to Geoff (2017), 79% of companies that have high-performing supply chains generally achieve greater revenue growth as compared to the average within their industries. A significant number of companies aim to automate some of their supply chain operations and to further improve their efficiency. CSCMP (2015) found that the total logistics costs, associated with delivery of a $3.60 box of cereal from the field to the consumer, was approximately $0.37 in the United States (U.S.). The net retail profit for the same product was approximately $0.05. It is obvious that a reduction in the cost of delivery and more effective distribution processes would lead to increased profitability.

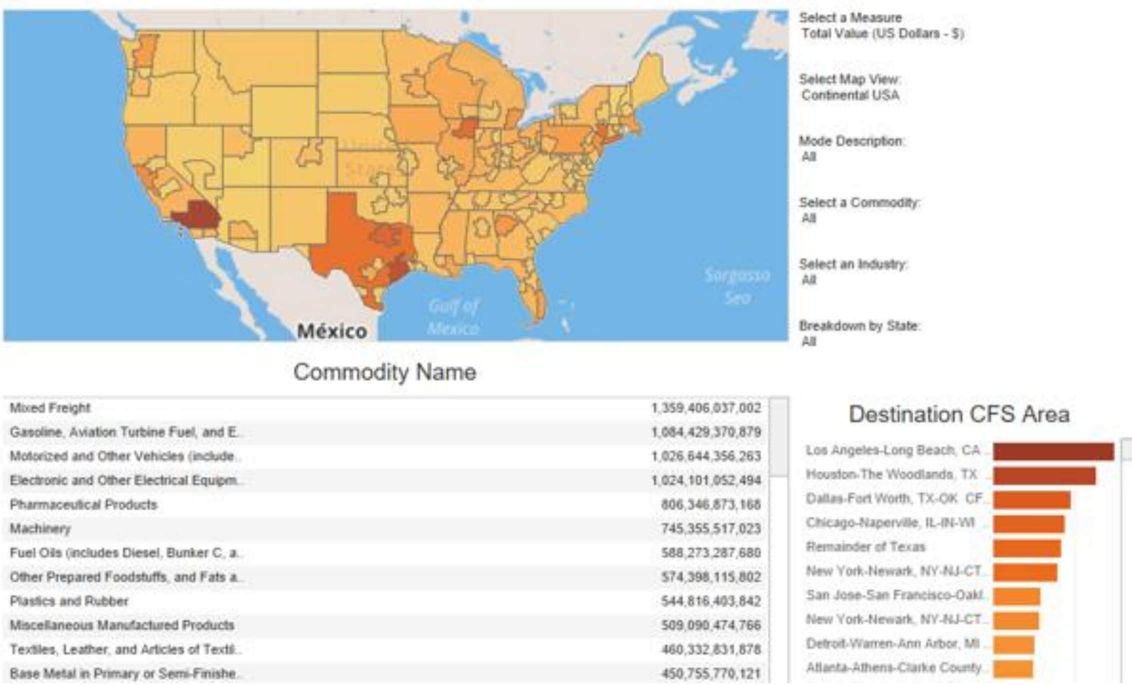
It is expected that the grocery industry could save up to $30 billion (~10% of the total operating cost) from introducing and implementing more effective logistics strategies (Jayant, 2016). For example, usually a box of cereal spends approximately 104 days between the production factory and the supermarket, while a new car spends an average of 15 days to be delivered from the factory to the dealership. In both cases, the real travel time is on average less than 5 days (Jayant, 2016). An implementation of the cross-docking strategy can be considered as one of the most effective alternatives for facilitating the movement of a product from the manufacturing plant to the retailer (or to the end customer).



Source: Bureau of Statistics (2012)

Figure 8: Commodity flow survey I (2012).

The flow of raw materials and commodities is growing rapidly in the supply chains of different industries. Figure 8 and Figure 9 show the survey of commodity flows within the U.S., where the first figure maps out the origins of commodities, while the second figure contains their respective destination locations. The commodity flow survey is performed after every five years. The Department of Transportation’s Freight Analysis Framework (FAF) is used as the principal source of data. The value of mixed freight, which comprised around $1.3 trillion in 2012 (see Figure 8 and Figure 9), had increased to $1.5 trillion by 2017. This confirms a significant growth in freight transportation across the nation and the need for effective distribution of products between various states.



Source: Bureau of Statistics (2012)

Figure 9: Commodity flow survey II (2012).

Figure 10 compares the weight of shipments in the years of 2012 and 2015. The conducted study was particularly interested in freight movements by trucks. A 6.8% increase in the total weight of freight flows was observed between the two years considered. This increase can be considered as significant in the freight transportation statistics and can be explained by increasing demand for commodities. The demand for commodities, if not well-managed, can create a tension in supply chains. Therefore, it is important for supply chains to create more efficient approaches to manage the growing demand. The cross-docking process generally takes place in a dedicated docking terminal, where the inbound products are received first at a dock and then sorted based on the assigned final destinations. The sorted products are consolidated, moved to the other side of the dock using forklifts, conveyor belts or other equipment, and then loaded on the outbound vehicles. Some manufacturers use cross-docking in their own facilities for moving finished products directly from the production side to an outbound dock (therefore, any sorting procedures are not performed).

	2012				2015			
Millions of Tons	Total	Domestic	Exports	Imports	Total	Domestic	Exports	Imports
Truck	10092	9899	105	89	10776	10568	108	100
Rail	1616	1481	53	82	1602	1459	55	89
Water	884	502	68	313	884	544	95	246
Air, air & truck	10	2	4	4	10	2	4	5
Multiple modes & mail	1311	309	596	406	1346	324	615	407
Pipeline	2942	2671	37	233	3326	3056	43	226
Other & unknown	41	37	1	3	33	29	1	3
Total	**16896**	**14901**	**864**	**1130**	**17978**	**15983**	**920**	**1075**

Note: This figures was prepared using the data from Bureau of Statistics (2015)

Figure 10: Weight of shipments by transportation mode: years of 2012 and 2015.

The modern-day distribution environment also seeks to reduce the product inventory and associated inventory costs at every step throughout the supply chain operations (Yu, 2002). According to Murray (2019), cross-docking solutions generally enable companies to expedite delivery of shipments to the end customers, which means that the customers will be able to get what they want and when they want it. The latter outcome is common for optimized supply chains, where operations are properly planned. The cross-docking technique makes supply chains more reactive and faster while keeping prices low (Ladier and Alpan, 2016). Such features of cross-docking are particularly valuable in the current competitive environment, which is characterized by customer's impatience.

Cross-docking is particularly important for the companies that handle a wide variety of products. Handling a wide range of consumer products requires the process of sorting. According to Saddle Creek Corp (2011), an effective implementation of the cross-docking strategy allows an efficient movement of durable goods through supply chains, high-value/high security goods, perishable items, as well as temperature-controlled items. In fact, since 2008, the use of cross-docks for perishables has increased by almost 5%. When goods are time-sensitive, the cross-docking strategy increases the speed, at which goods are made available to the consumer, while minimizing the costs involved. Another notable benefit that cross-docks provide to a supply chain is increasing turnover rates, mitigating the risk of loss or damage. Considering increasing volumes of freight (including time-sensitive products), transported between various states across the nation, the cross-docking strategy has been receiving more and more attention from practitioners.

1.4 Advantages and Challenges of Cross-Docking

This section specifically focuses on the advantages of the cross-docking strategy and the challenges, which are typically faced by various stakeholders when implementing cross-docking in a supply chain. It is noteworthy that these advantages only apply to the supply chains where the cross-docking strategy is suitable.

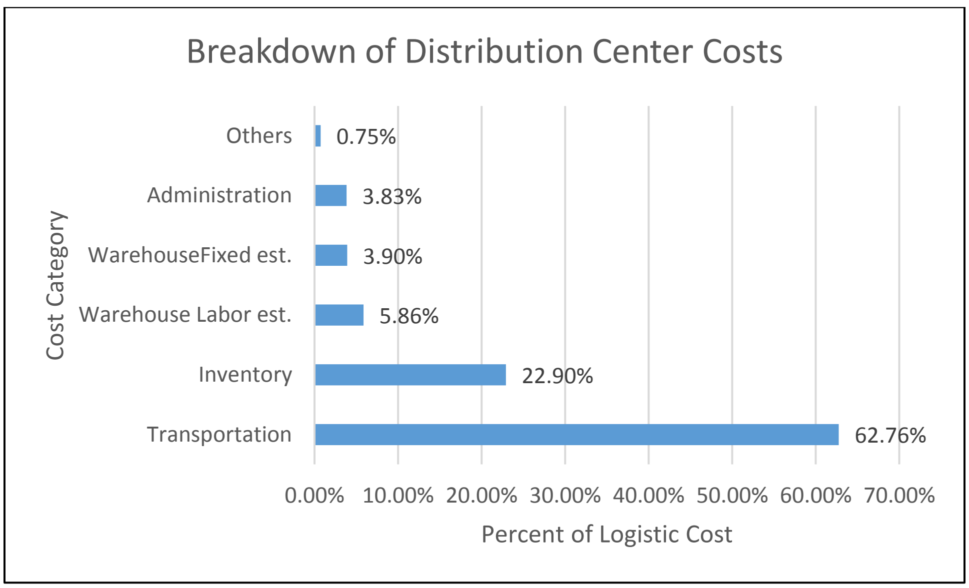
1.4.1 Advantages of Cross-Docking

The benefits, which can be achieved from the cross-docking concept, are numerous. It can save time and money by eliminating the need for moving goods from one location to another. It can also help in improving customer service by allowing customers to talk to their representatives instead of waiting for them to speak with an agent. It can also improve customer satisfaction by providing more information about products, services, or even particular product features.

The biggest challenge, faced by food supply chains, is the delivery of the food items to customers on time, at minimal cost, good quality, without compromising the profitability of the producers. This challenge is as a result of the distance that exists between the food production facilities and the consumption locations (Ahumada and Villalobos, 2009). The cross-docking strategy ensures that processing of products is completed within 24 hours from the time the products are received at a given CDT to the time the products are shipped out to the customers. Cross-docking is, therefore, used as a means of improving efficiency of food supply chains and other supply chains as well. Cross-docking is also used to reduce the cost of goods sold by the companies who use it as a strategy (i.e., the cost of products can be lowered due to savings achieved throughout the product distribution). In addition to the stated advantages, some of the common benefits that can be achieved by using the cross-docking technique include the following:

1) Cross-docking reduces the size of the distribution facility needed, eliminates storage, and reduces the associated costs. The elimination of storage reduces the square footage needed. The only space, required by cross-docking facilities, is the space for the basic transshipment activities.
2) Since the use of the cross-docking strategy involves direct transshipment, handling of the products is reduced. The latter feature makes the cross-docking strategy more beneficial as compared to the warehousing strategy, where the products are unloaded, stored (for a significant amount of time in many cases), and then retrieved.
3) Cross-docking utilization results in reduction of delivery times. Such benefit is considered as substantial for respective supply chains, as it directly leads to customer satisfaction. The delivery times are generally reduced because the outbound trucks will have a composition of various product types, created based on the customer preferences, rather than delivery of each product type independently by different trucks without sorting.
4) Cross-docking results in the transportation cost reduction. According to CSCMP's annual state logistics report for the year of 2011 (see Figure 11), the transportation cost comprises approximately 62.8% of the total supply chain distribution center cost in the U.S. The transportation cost would be relatively high if a supplier has to reach an array of customers directly. However, the cross-docking strategy optimizes that by sorting the products that have been transported by the inbound trucks from various suppliers based on their final destinations.

5) Application of the cross-docking strategy in its basic form does not include storage; therefore, damages and losses of the goods are also reduced. The products, transferred through a CDT, are safer and have a higher probability of maintaining a good quality throughout the supply chain operations.
6) Since the labor cost is proportional to material handling, the labor cost is reduced for the supply chains that rely on the cross-docking strategy. According to CSCMP's annual state logistics report for the year of 2011 (see Figure 11), the labor cost comprises approximately 5.9% of the total supply chain distribution center cost in the U.S. Reduction in the labor cost will further allow increasing the total profit to be generated by a given company, if the cross-docking strategy is successfully implemented.
7) Cross-docking eliminates order picking, which is an expensive supply chain process. The latter further facilitates handling of the products within the CDTs.



Note: This figures was prepared using the data from CSCMP (2011)

Figure 11: Breakdown of the distribution center costs.

All the aforementioned advantages justify such a wide implementation of the cross-docking strategy among various supply chains (Van Belle et al., 2012).

1.4.2 Challenges of Cross-Docking

Cross-docking operations have several challenges, ranging from strategic planning to daily operations planning at the facility. The first notable challenge of using a cross-dock system is the implementation decision, i.e. deciding whether the cross-docking strategy would be suitable for a given supply chain (so, the initial monetary investments into the development of cross-docking facilities across a given supply chain will pay off in future). As previously stated, cross-docking is an integral part of a supply chain and cannot be considered as a stand-alone component. The setting-up of a CDT requires quite extensive design and planning, and the processes have to be closely monitored to ensure that everything works as intended. The cross-docking implementation decision is also affected with the product types, which are being handled within a given supply chain. As discussed by Murray (2019), the cross-docking strategy is typically efficient for supply chains with the following types of products:

- Perishable goods;
- Goods that are pre-picked and pre-packaged from the place of production;
- High-quality goods that do not require quality inspection;
- Goods that have a stable demand (Gue, 2011).

Logistics providers and the CDT managers generally negotiate the expected arrival time of the trucks both in- and outbound; so, that the CDT manager can perform necessary planning, allocate sufficient handling equipment, and reserve adequate capacity of the dedicated locations of storage areas. However, there may be some uncertainties, associated with arrival time of the trucks. These uncertainties may be caused by congestion on the roadways of the transportation network, inclement weather conditions, potential truck breakdowns, and others. The CDT managers should be able to effectively reschedule the available handling resources to ensure that the trucks, which do not meet the agreed arrival times, will be served in a timely manner in order to prevent significant disruptions in the CDT operations and reduce truck service delays.

As identified by Dulebenets (2018a,b), another challenge in the management of the CDT facilities is the efficient scheduling of the arriving trucks. The truck scheduling problem aims to assign the arriving trucks (both in- and outbound) to the CDT doors that are available for service, determine the appropriate truck service order, truck start service times, and truck finish service times (Dulebenets, 2019a). Certain critical operational features have to be taken throughout truck scheduling, which include, but are not limited to, restrictions in the truck service order, availability of the existing equipment for internal transportation, the available space within the existing locations of storage areas, and others. Along with uncertainties in arrival time of the trucks, the CDT managers may face uncertainties, which are associated with service of these trucks (e.g., the handling time of trucks may increase due to congestion at the facility, breakdowns of the existing equipment for internal transportation, improper planning of the CDT operations).

Throughout scheduling of the arriving inbound trucks and outbound trucks, the CDT manager aims to achieve certain objectives. The common objectives include the following: (1) minimization of makespan – the total time of cross-docking operation calculated from the unloading the first inbound truck to the loading of the last outbound truck; (2) minimization of the total cost associated with the CDT operations; (3) minimization of the total travel distance of forklifts within the CDT; (4) minimization of the total delayed departures of the inbound trucks; (5) minimization of the total delayed departures of the outbound trucks; and others. Efficient truck scheduling and management of the internal handling operations are critical, as they directly influence the major performance indicators of supply chains.

1.5 Main Dissertation Objectives

As indicated earlier, a lot of efforts have been dedicated to the improvement of the efficiency of supply chain operations. The cross-docking strategy was found to be one of the most promising alternatives. CDTs could yield good results when used within supply chain operations, only if they are well-managed. The management of CDTs can be considered a challenging task. The CDT managers have to allocate the existing resources accordingly based on the projected demand (i.e., determine the most convenient layout of the facility; designate the available doors for service of the arriving trucks both in- and outbound; schedule the available handling equipment in order to load and unload the products from the arriving trucks; allocate adequate capacity for the temporary locations of storage areas, where the arriving products will be decomposed, sorted, and consolidated). Scheduling of the arriving trucks both in- and outbound to the available CDT doors is recognized as a challenging decision problem. The problem complexity may be further increased when additional operational features are considered (e.g., modeling of the decay for perishable products throughout handling process; truck service order restrictions; allocation of the temporary locations of storage areas for time-sensitive and perishable products).

The CDT truck scheduling problem with perishable products will be the main focus of this dissertation. The list of contributions of this work to the state-of-the-art and the state-of-practice can be summarized as follows:

1) Formulate a novel mixed integer mathematical model for the CDT truck scheduling problem, where the inbound and outbound trucks carry perishable products. The proposed model directly captures the decay of perishable products throughout handling and transfer operations within the CDT.
2) Propose the appropriate linearization methods to linearize the non-linear components (i.e., the components that are associated with the computation of the perishable products decay) of the mathematical model and formulate the linearized model, as presence of the non-linear components is expected to increase the computational complexity of the model.
3) Develop a novel Evolutionary Algorithm (EA) customized for the problem of interest as the solution method and compare it against the existing solution algorithms (specifically, Variable Neighborhood Search (VNS), Tabu Search (TS), and Simulated Annealing (SA)) as well as against the exact optimization algorithm.
4) Conduct a comprehensive set of computational experiments and sensitivity analyses to demonstrate the performance of the proposed solution method and the developed optimization model. The major managerial insights that would be important to the CDT managers and will assist with effective handling of perishable, temperature-controlled, and other time-sensitive products are highlighted as well.

CHAPTER TWO

2 LITERATURE REVIEW

This section contains a comprehensive review of the literature relevant to cross-docking, truck scheduling, and optimization models, which were previously developed to improve efficiency of truck schedules at cross-docking terminals (CDTs). Throughout review of the literature, the primary emphasis was given to the mathematical models proposed, new operational features considered in the CDT truck scheduling, solution methodologies presented, and the key findings. The collected studies are grouped into the following categories: (a) general CDT truck scheduling; (b) multi-objective CDT truck scheduling problems; (c) uncertainty modeling in CDT truck scheduling; and (d) miscellaneous. Furthermore, based on a detailed review of the state-of-the-art, a summary of findings will be provided at the end of this section.

2.1 General CDT Truck Scheduling

Yu (2002) studied the CDT operations as they directly affect the effectiveness of supply chains. Three basic CDT scenarios were generated in the study: (1) a CDT with temporary storage and the preemption (i.e., a given truck can leave the door, so another truck can be docked for service) is not allowed; (2) a CDT without temporary storage and the preemption is allowed; (3) a CDT with temporary storage and the preemption is allowed. The objective of the study was to search for the best truck schedule for the minimization of the makespan (i.e., the completion time of the service of the last truck). The adopted solution approaches included the exact optimization algorithms and heuristics. While the exact methods solved the problem instances to optimality, the heuristics had comparable solutions. Findings from the study showed that good solutions would result in more efficient operations of CDTs due to the reduction in the warehouse cost (otherwise known as the inventory cost).

Li et al. (2004) focused on the problem of CDT truck scheduling. The aim of the mathematical formulation was to find a schedule that minimizes the total penalty of operations at a CDT. The considered penalties included earliness and tardiness in processing the inbound and outbound containers. Furthermore, another study objective was to eliminate or minimize the storage and order-picking. The problem was formulated as a machine scheduling problem. The authors presented two heuristic algorithms to solve the problem, both of which relied on an Evolutionary Algorithm (EA). The first algorithm applied the Squeaky Wheel Optimization Genetic Algorithm (SWOGA), while the second one relied on linear programming and was referred to as Linear Programming Genetic Algorithm (LPGA). The proposed heuristic algorithms (SWOGA and LPGA) outperformed CPLEX in terms of the solution quality and the computational time for the large-size problem instances. The SWOGA computational time usually took less than one minute, even for the large-size problem instances. LPGA, on the other hand, frequently achieved the best performance, but required longer computational time than SWOGA.

Lim et al. (2005) studied the CDT transshipment problem with supplier and customer time windows in a constrained flow. The main objective was the minimization of the total transportation cost and total cost of product storage. Three types of transshipment schedules were discussed: (1) fixed schedule (each supplier can send goods multiple times using its schedules within specific time windows, and each customer can receive goods multiple times through the schedules within specific time windows); (2) flexible schedule (goods can be sent and received at any time within specific time windows); and (3) mixed schedule (a flexible schedule followed by a fixed schedule and/or a fixed schedule followed by a flexible schedule). The existence of an optimal solution for the various schedules was discussed, as well as the complexity of the problems. The single shipping-single delivery problem and the mixed shipping and delivery problem were also described in the study. The optimality conditions were provided for the cases that were solvable in a polynomial time, while the proofs of complexity were provided for the cases unsolvable in a polynomial time.

McWilliams et al. (2005) proposed a simulation-based scheduling algorithm (SBSA) that relied on an EA to drive the search for new solutions. The objective of the study was to search for the best schedules of trucks for the minimization of the time span of the transfer operation. The transfer operation involved the unloading of inbound trucks (ITs) and transporting the freight to the appropriate outbound trucks (OTs) based on the freight destination. The proposed algorithm had two components: (1) a detailed deterministic simulation model used for evaluating truck schedules; (2) a search procedure that generated and presented the schedules to the simulation model for evaluation. The experimental analysis, performed in the study, involved small-, medium-, and large-size problem instances. The time span of transfer operation was used as the performance measure. The results of experiments showed that SBSA performed better than the arbitrary scheduling (where the best truck schedule was identified over a large number of randomly generated truck schedules), which had been widely used by freight consolidation terminals at the time of the study.

Ley et al. (2007) modeled CDT truck scheduling. The objective of the model presented was the minimization of the total travel distance by the internal handling equipment. It was highlighted that the required computational time could substantially increase for the large-size problem instances that involve hundreds of trucks. Therefore, an EA was proposed as a solution algorithm to solve the CDT scheduling problem. The numerical experiments were conducted using (4, 5, 6, 7) ITs and (4, 5, 6, 7) OTs. The results showed that the computational time, required by the EA algorithm, to provide the solutions was lower, when compared to the exact optimization algorithm. The EA results also showed the accuracy of above 98%.

Song and Chen (2007) studied a two-stage CDT logistics optimization problem. The objective was to minimize the makespan. The considered CDT had multiple ITs and one OT. The problem was formulated as an MIP and solved using CPLEX for the small-size problem instances. Two heuristics were then created to solve the medium-size and large-size problem instances: (1) Johnson's Rule-based Heuristic (JRBH); and (2) Dynamic Johnson's Rule-based Heuristic (DJRH). Three lower bounds were derived to compare the results of the proposed heuristics: (1) minimal waiting time lower bound; (2) Johnson's rule-based lower bound; (3) integrated lower bound. The findings showed that the Johnson's Rule-based Heuristic performed better than the Dynamic Johnson's Rule-based Heuristic.

Shakeri et al. (2008) developed a generic model for the assignment of trucks to the CDT doors and scheduling of trucks at the assigned CDT doors. An MIP model was formulated for the problem, aiming to minimize the makespan. The authors highlighted that the proposed MIP model could be applied for the arbitrary number of the CDT doors, arriving trucks, product types handled, and different product handling times. Furthermore, it was pointed out that the developed methodology could be useful in Fast-Moving Consumer Goods (FMCG) supply chains. Development of the solution approaches for the formulated MIP model, including the exact and heuristic algorithms, was left for the future research.

Vis and Roodbergen (2008) focused on modeling the process of short-term storage of unit-loads at a CDT. The study objective was to identify the most appropriate storage location that minimized the total travel distance by forklift operators within the CDT. The problem was modeled as the minimum cost flow problem. The presented mathematical model was solved using a row-based storage assignment algorithm. A number of numerical analyses were carried out to examine performance of the developed row-based storage assignment algorithm for different types of layouts and priorities. The results were compared with a commonly used heuristic method, called as “the nearest location first heuristic”. It was found that the proposed solution approach was able to reduce the average forklift travel distance by 40% as compared to the nearest location first heuristic.

Yu and Egbelu (2008) addressed the problem of CDT truck scheduling with temporary storage. The objective of the study was to find the best schedule of the arriving trucks that minimizes the total operation time and maximizes the throughput. The problem was formulated as a mixed integer programming model with the objective function, aiming to minimize the makespan. Two solution approaches implemented were the exact optimization (enumeration) and the heuristic methods. The first method produced solutions of good-quality for the small-size problem instances; however, it was not efficient for the medium- and large-size problem instances. The heuristic method was developed to solve the model for the large-size problem instances in a reasonable computational time. Three strategies were created in selecting the next scheduled ITs and OTs (RS1 – RS3 for ITs and SS1 – SS3 for OTs), making a combination of nine strategies. The IT selection strategies included the following: (1) RS1: the IT that transfers the smallest number of products to the temporary storage; (2) RS2: the IT that transfers the largest number of products to the OT; (3) RS3: the IT that has the smallest ratio of the products transferred to the temporary storage to the products transferred directly to the OT. The OT selection strategies included the following: (1) SS1: the OT with the associated IT that transfers the smallest number of products to the temporary storage; (2) SS2: the OT that has the shortest time at the shipping dock; (3) SS3: the OT with the associated IT that has the smallest ratio of the products transferred to the temporary storage to the products transferred directly to the OT. The combinations that involved SS2 produced the most inferior solutions. The combination of RS1 and SS1 produced the best solutions. On average, the solutions, produced by the heuristic, were within 1.8% deviation from the optimal solutions. Furthermore, while the computational time of the enumeration method increased with increasing number of trucks, the computational time for the proposed heuristic was not significantly affected.

Chen and Lee (2009) studied a two-machine CDT flow shop scheduling problem, where processing a job at the second machine could begin only once some jobs at the first machine had been completed. The objective function of the proposed model was the minimization of the makespan. It was found that the considered problem had NP-hard complexity. A Polynomial Approximation Algorithm with an error-bound analysis and a Branch-and-Bound (B&B)-based Algorithm were developed to solve the problem. Based on the conducted computational experiments, the B&B-based Algorithm could solve the problem instances with up to 60 jobs to the global optimality. Furthermore, approximately 65% of the generated problem instances with 60 jobs could be solved within 5 minutes.

Chen and Song (2009) studied a two-stage hybrid CDT truck scheduling problem, where the arriving jobs are processed in stages (the second stage jobs can be processed after completing the first stage jobs). The objective of the study was to minimize the makespan. An MIP model was proposed, and the small-size problem instances were solved using CPLEX. Four heuristics were then proposed as the solution approaches to the medium- and large-size problem instances. The heuristics were: (1) Johnson's Rule-based Ready Time Heuristic (JRH); (2) Johnson's Rule-based Largest Processing Time Heuristic (JLPTH); (3) Dynamic Johnson's Rule-based Ready Time Heuristic (DJRH); and (4) Dynamic Johnson's Rule-based Largest Processing Time Heuristic (DJLPTH). A comparison among the heuristics was carried out with a derived Lower Bound (LB) using the loss as a criterion. The loss was estimated as the difference between the obtained makespan values and the derived LB divided by LB. The small-, medium-, and large-size problem instances were solved, and the results showed that JLPTH was superior to the other heuristic algorithms.

Maknoon and Baptiste (2009) conducted a study on the CDT operations, capturing a direct flow of products from the receiving to shipping doors. The objective of the study was to minimize the total travel distance by the internal handling equipment within the CDT by synchronizing incoming and outgoing trucks (i.e., determination of the optimal product flow path). Two approaches were proposed to solve the problem: the first approach used a dynamic programming (DP) algorithm to obtain the optimal loading and unloading policies, while the second approach used a heuristic approach. The numerical experiments showed that the heuristic approach was superior in terms of the computational time and quality of the obtained solutions.

McWilliams (2009) addressed a parcel hub scheduling problem (PHSP), which involved the assignment of ITs to unloading docks, aiming to minimize the transfer operation makespan. A 0-1 minimax programming model was proposed for the problem. The problem was found to have NP-hard complexity. A combination of time-based and resource-based decomposition was proposed. While the former breaks down the problem time horizon into smaller sub-periods, the later transforms the problem into parallel identical unload dock scheduling problem. The model was solved with an EA that was developed. The numerical experiments involved small, medium, and large CDT sizes. The solutions obtained were compared with the Simulation-Based Scheduling Algorithm (SBSA) and Random or Arbitrary Scheduling (ARB). The solution quality was evaluated based on the ratios of various workload parcel values, and the proposed EA offered superior solutions. Furthermore, the proposed algorithm required less computational time.

Miao et al. (2009) focused on the problem of truck-to-dock assignment at a CDT, considering the operational time constraints. The proposed model aimed to search for the best truck assignment that minimized the operational cost and total penalty due to unfulfilled shipments at the same time. An IP model was proposed to address the problem. Due to the inability of CPLEX to handle the large-size problem instances in a reasonable computational time, two metaheuristic approaches were proposed, including TS and EA. A number of small-, medium-, and large-size problem instances were generated to test performance of the solution approaches. The metaheuristics outperformed CPLEX in terms of solution quality and computational time. Superiority of the metaheuristics was even more obvious for the medium- and large-size problem instances. Furthermore, it was found that TS outperformed EA for the generated problem instances.

Boysen (2010) addressed a special CDT truck scheduling using a zero-inventory strategy for the food industry, where all the food products were directly loaded onto the refrigerated OTs without temporary storage. The objective of the study was to minimize the following performance indicators (independently): (1) the total flow time; (2) the total processing time; and (3) the total late departures of OTs. The flow time of an OT was defined as the whole-time span from the time period, when that OT docked, to the departure of the OT. The processing time was defined as the time span for which OTs remain under processing at the considered CDT. The developed solution approaches for the problem of interest included dynamic programming (DP) and SA. The computational experiments demonstrated that DP could solve the problem instances with up to 20-25 ITs to optimality. The SA approach was presented to handle the real-world problem instances, which could not be solved to optimality by DP. The quality of the solutions, obtained by SA, showed that the approach was more promising for the real-world problem instances.

Forouharfard and Zandieh (2010) focused on the problem of scheduling arriving trucks at a CDT with temporary storage. The objective of the study was to search for the best truck schedule that minimized the total number of products passing through the temporary storage. The problem had non-deterministic polynomial-time (NP)-hard complexity. The authors proposed the Imperialist Competitive Algorithm (ICA) to solve the problem. This algorithm was inspired by an imperialistic competition where powerful empires collapse weaker ones. Performance of the proposed ICA algorithm was evaluated based on the numerical experiments, which were based on the analysis against an EA. A relative percentage deviation (RPD) of the total products that passed through the temporary storage was used as a performance measure for the algorithms (ICA and EA). It was found that EA generally converged faster than ICA; however, the RPD results showed that ICA was superior to EA with 95% confidence.

Vahdani et al. (2010) considered a CDT that had no temporary storage but allowed for preemption. A mathematical model of the NP-hard complexity was proposed for the problem, aiming to minimize the total system flow time. The following metaheuristics were applied to solve the problem: (1) TS; (2) HEA; and (3) Hybrid Electromagnetic-like algorithm (HEM). A set of problem instances were generated alongside the problem instances, used by Yu (2002). The parameter selection analysis for the developed algorithms was conducted using the Taguchi's scheme. Based on the results, obtained from the performed experiments, it was found that the HEM algorithm outperformed the alternative algorithms in terms of the solution quality.

Vahdani and Zandieh (2010) presented an MIP model for scheduling ITs and OTs at a CDT, which was similar to the one that was developed by Yu and Egbelu (2008). The model's objective function was to minimize the makespan. A total of five metaheuristic algorithms were applied to solve the problem, including the following: (1) EA; (2) TS; (3) SA; (4) Electromagnetism-like algorithm (EM); and (5) VNS. The performance of the metaheuristics was compared based on the makespan and CPU time values. A total of 25 large-size problem instances were generated alongside the problem instances, used by Yu and Egbelu (2008), in order to evaluate the proposed metaheuristic algorithms. The developed metaheuristic algorithms outperformed the heuristic approach adopted by Yu and Egbelu (2008). Moreover, the VNS algorithm was the most promising algorithm if consider the values of the objective function at termination.

Alpan et al. (2011) studied the problem of transshipment and truck scheduling at a CDT, where the CDT manager had to decide whether to move the products directly from the receiving doors to the shipping doors (if the corresponding outbound trucks had been already docked at the outbound doors), temporarily store certain products and load them later on the corresponding OTs, or replace certain OTs in order to facilitate the loading process. The objective of the study was to find the best transshipment schedule for the minimization of the total cost of product storage and the total truck replacement cost. Three heuristics were used as the solution approaches and were inspired by the following aspects: (1) the first heuristic forced some destinations at the ODs; (2) the second heuristic avoided some destinations at the ODs; (3) the third heuristic was based on the limited storage capacity. The heuristics were compared to the previously developed DP method. Based on the numerical experiments, it was discovered that hybridization of heuristics (i.e., combination of various rules) was more promising rather than using separate heuristics (that relied on one of the rules).

Boloori Arabani et al. (2011) focused on the problem of truck scheduling at a CDT with temporary storage. The objective was to minimize the makespan. The MIP model, developed by Yu and Egbelu (2008), was adopted in that study. In addition to the previous work by Yu and Egbelu (2008), five (5) metaheuristics were applied to find the best sequence of ITs and OTs, including the following: (1) EA; (2) TS; (3) PSO; (4) ACO; (5) DE. The RPD and signal-to-noise ratio (S/N) were used as performance measures. The problem instances were generated to implement the developed algorithms. A comparative analysis of the five metaheuristics with the heuristics, proposed by Yu and Egbelu (2008), was conducted. The developed metaheuristic algorithms produced better solutions. Performance of the metaheuristics was also compared with one another using the mean makespan and the least significant difference (LSD). The proposed DE had the least mean makespan, while the proposed EA had the least LSD. The metaheuristics had generally similar performance, except TS that demonstrated a worse performance.

Choy et al. (2012) modeled the problem of scheduling of trucks at a space-constrained CDT (where the number of ITs exceeds the number of available docks). The objective of the presented model was the minimization of the makespan. An EA-based metaheuristic was deployed to find the solution to the problem. Performance of the adopted solution approach was compared with the First Come First Served (FCFS) dock assignment strategy. The problem instances were created for the small, medium, and large inbound order sizes. The results showed that the proposed method produced better solutions than the FCFS strategy. Furthermore, the makespan was reduced by 10% - 20% under heavy and normal congestion scenarios from application of the proposed methodology.

Liao et al. (2012) studied the sequencing of ITs and OTs at a CDT. The objective of the study was to minimize the makespan. The authors developed a new CDT operational policy, referred to as the Policy-LE, for computing the makespan, aiming to improve the Policy-YE that was developed by Yu and Egbelu (2008), which had the following major deficiencies: (a) doubling of the loading and unloading time for the products, passing through the locations of storage areas; and (b) no loading and unloading activities are taken during the time of truck change as well as during the time, when the first product is unloaded from the IT and moved to the OT. A total of three metaheuristics were proposed to address the problem, these include DE, Hybrid DE (HDE) – 1, and HDE-2. The HDE algorithms relied on the additional local search heuristics, unlike the canonical DE. Furthermore, the HDE-2 algorithm used the Tabu list. The computational experiments were conducted using thirty (30) problem instances in order to assess performance of the developed solution algorithms. Introduction of the additional local search heuristics and the Tabu list provided a computation advantage in the search process. The proposed approach for calculating the makespan was found to be superior to the one used by Yu and Egbelu (2008).

Liao et al. (2013) studied the scheduling of inbound trucks at a multi-door CDT. The objective of the study was to minimize the total weighted tardiness of the handled products. The total weighted tardiness was computed based on the number of unmet units of a specific product and the unit penalty weight of that product. The authors proposed six metaheuristics to solve the model: SA, TS, ACO, DE, and two hybrid DE algorithms (HDE-1 and HDE-2). Unlike a typical DE algorithm, the HDE-1 and HDE-2 algorithms deployed additional local search procedures. The algorithms were evaluated using 40 problem instances. The developed algorithms were found to be efficient for the problem of interest. The study highlighted the importance of selecting the appropriate parameter values for metaheuristics. The ACO and HDE-2 algorithms were found to be the most promising solution algorithms in terms of the solution quality. However, the ACO algorithm required lower computation time as compared to the HDE-2 algorithm.

Gelareh et al. (2013) identified some limitations in the IP model, proposed by Miao et al. (2009) for the truck-to-dock assignment problem at a CDT. An alternative mathematical model was presented, which addressed the identified issues and minimized the total cost. The total cost included the cost of transporting the products within the CDT and the penalties for the unfulfilled shipments. A number of valid inequalities were proposed in order to accelerate the problem resolution. CPLEX was adopted as a solution approach in the study. A computational time limit of 1,500 sec was imposed on CPLEX. A series of computational experiments demonstrated that introduction of the proposed valid inequalities was promising, as CPLEX was able to solve a significant amount of problem instances either to optimality or provide good-quality solutions within the time limit imposed.

Golias et al. (2013) developed a CDT truck scheduling model with the objective to minimize the total cost and the total truck service time related to early and late departure of trucks from a CDT. A Hybrid EA was deployed to search for solutions to the model for the generated problem instances. A hybridization technique was implemented to discover superior solutions after application of the insert and swap mutation operators. The results from numerical experiments indicated that the presented truck scheduling policy outperformed the previously proposed policies of sequential scheduling of ITs and OTs on average by 20%.

Joo and Kim (2013) considered the scheduling of trucks at a CDT with multiple doors. The trucks were classified into three groups: inbound-only trucks, outbound-only trucks, and compound trucks. The objective function of the proposed model was the minimization of the makespan. The authors proposed two EA-based metaheuristics, which were referred to as a Genetic Algorithm (GA) and a Self-Evolution Algorithm (SEA). Each of the algorithms had two different chromosome designs, including the following: (1) special character (SC) design; and (2) dispatch rule (DR) design. The results from numerical experiments showed that the proposed algorithms had relatively low computational time when compared to CPLEX. Furthermore, the SEA algorithm with the DR design demonstrated the best performance regarding the values of the objective function at termination for the considered problem instances.

Kuo (2013) conducted a study to improve the efficiency of truck sequencing and truck-to-door assignment at a CDT with multiple doors. The objective of the study was to minimize the makespan. The proposed model had two phases, where the first phase calculated the time required to unload all the products from ITs, while the second phase calculated the time required to complete loading for all the OTs. The problem was solved using the VNS algorithm and four variations of the SA algorithm. The computational experiments showed that VNS produced the solutions, which were superior to the randomly generated solutions with improvements ranging from 8.23% to 40.97%. However, VNS was not able to outperform the SA-based algorithms in terms of the values of the objective function for all the problem instances considered. Nevertheless, the VNS algorithm was able to yield robust solutions with a reasonable computational time.

Agustina et al. (2014) proposed an MIP model that focused on the problem of the scheduling of trucks and vehicle routing at a CDT simultaneously. The model was developed for the minimization of the total cost, which comprised of four components: (1) cost due to earliness penalty; (2) cost due to tardiness penalty; (3) cost of inventory holding; and (4) cost of transportation. The commercial solver CPLEX was applied to solve the model. Findings showed that only small-size problem instances could be solved in a reasonable amount of time using CPLEX. The concepts of customer zones and hard time windows were therefore introduced. The customer zone concept reduced the size of the search space, since each route considered only the customers within a zone. The hard time window, on the other hand, reduced the search space by not allowing deliveries outside of the scheduled time window. The numerical experiments were conducted using the introduced concepts. The results showed that the search space for the medium- and large-size problem instances was reduced. Hence, the computational time, required to solve the large-size and medium-size problem instances, significantly decreased. The model was found to be suitable for real-life generation of the optimal vehicle routing and scheduling strategies at CDTs, which could assist in food distribution at a minimum cost and preserve quality.

Guignard et al. (2014) considered a mathematical model, referred to as the cross-dock door assignment problem (CDAP), which was originally studied by Tsui and Chang (1990). The proposed model minimized the total distance traveled by the goods inside a CDT. Two approaches were considered for scheduling the arriving trucks: (a) static approach; and (b) dynamic approach. Under the static approach, the trucks are scheduled in advance before the beginning of the planning horizon. On the other hand, the dynamic approach allowed updating the initial truck schedule upon the arrival of each truck for service at the CDT. The study proposed a solution approach, which was named as a Convex Hull Heuristic (CHH). As a result of the conducted simulation experiments, it was found that the dynamic approach was able to yield the truck schedules with better CDT door utilization and more even distribution of flows among the inbound and outbound CDT doors. Furthermore, the dynamic approach improved the total CDT throughput and allowed serving more ITs throughout the considered planning horizon. However, the dynamic approach required more CDT doors, which could incur additional handling costs.

Madani-Isfahani et al. (2014) studied the problem of scheduling trucks at multiple CDTs. The shipping docks were assumed to have the temporary locations of storage areas. An MIP was formulated for the considered problem. The objective function was to minimize the total operational time (i.e., maximize the total throughput or minimize the makespan). Two solution approaches which included SA and the Firefly Metaheuristic (FM) were used. Two types of delays were considered in the study: (1) delay due to change in the shipping truck; (2) delay due to the shipping truck waiting for products to arrive. The numerical experiments were conducted using 20 large-size problem instances. A Taguchi's method was used to select the appropriate values for the parameters of the developed algorithms. The FM algorithm was found to be more promising in terms of both solution quality and computational time.

Morais et al. (2014) addressed a problem of vehicle routing with cross-docking. The objective aimed to minimize the total cost. Three heuristics were proposed, which were based on Iterated Local Search (ILS): (1) S-ILS, which is a standard implementation of ILS; (2) X-ILS; and (3) SP-ILS. X-ILS extended the S-ILS heuristic by keeping a set of elite solutions to be used in a restart procedure. SP-ILS used an intensification procedure that allowed exploring an exponentially large neighborhood. The numerical experiments were conducted in order to assess the performance of the proposed heuristics using the 20 problem instances, generated by Wen et al. (2009). Then, the authors proposed a new set of 16 problem instances, which were more realistic and challenging. Findings showed that the optimality gap of SP-ILS was smaller as compared to TS used by Wen et al. (2009). SP-ILS produced better solutions for the vehicle routing problem with cross-docking than the alternative approaches, previously used in the literature. Therefore, the proposed solution approach was considered as promising for real-life applications.

Shiguemoto et al. (2014) formulated a mathematical model for the scheduling of trucks at a CDT. The objective was to minimize the truck service makespan. The model was solved using the proposed hybrid GA with Tabu Search (HGATS). The computational experiments were conducted to assess performance of the proposed algorithm. The small-size problem instances with 4 – 6 ITs and OTs were retrieved from the previous study, conducted by Yu and Egbelu (2008), while the large-size problem instances with 10 – 35 ITs and OTs were generated by the authors. The solutions, obtained using HGATS, were compared with those obtained by CPLEX, Yu and Egbelu (2008), and the Firefly Metaheuristic (FM). HGATS obtained the optimal solutions and outperformed the algorithm, proposed by Yu and Egbelu (2008), as well as FM. For the large-size problem instances, HGATS proved to be robust with a gap of 5.85% from its best to worst solution and 21.24% better than FM. The HGATS was also found to be promising in terms of the computational time.

Yan (2014) identified scheduling of ITs and OTs as the most crucial process for improving CDT efficiency. The objective function of the presented mathematical formulation was to minimize the makespan. The author applied a total of 4 heuristic algorithms in the order to solve the scheduling problem at the CDT, including the following: (1) Local Search (LS); (2) SA; (3) Large Neighborhood Search (LNS); and (4) Beam Search (BS). CPLEX was also considered as a candidate solution approach. The numerical experiments were further performed in order to evaluate the computational performance of the proposed solution approaches. Two sets of problem instances were created: (1) each IT carrying a single product type and OTs carrying a variety of product types; (2) ITs and OTs carrying multiple product types. Both sets of problem instances were generated for the small-size and large-size cases. The findings showed that LNS and BS had the best performance regarding the quality of solutions obtained and the CPU time.

Cekala et al. (2015) studied the problem of scheduling trucks, delivering the timber products at the manufacturing facility. It was pointed out that the initial truck schedule could change during the day as a result of unexpected truck loading difficulties and unexpected truck arrival delays. The objective aimed to minimize the total truck waiting time. A number of penalties were introduced in order to ensure constraint satisfaction, including the following: (1) penalty for overlapping appointments; (2) penalty for scheduling a truck yet to arrive over the truck that has already arrived; (3) penalty for scheduling a delayed truck before the truck that arrived on time; (4) penalty for scheduling an appointment for the truck that has arrived but whose appointment time has not started; (5) penalty for moving an appointment of the truck that is yet to arrive to an earlier appointment. An EA was applied to solve the problem. The proposed EA algorithm was examined for the solution quality and the computational time. The EA performance was compared with PSO and the manual dispatching method. The results demonstrated superiority of the developed EA algorithm.

Mohtashami (2015) considered the scheduling of ITs and OTs at a CDT with temporary storage. The objective function of the optimization problem was to minimize the makespan. An EA-based algorithm was proposed as a solution approach to the scheduling problem. Two scenarios were considered, including the following: (1) preemption is allowed; and (2) preemption is not allowed. The computational experiments demonstrated that the developed solution algorithm outperformed the one, which was proposed by Yu and Egbelu (2008), in terms of the makespan values. Furthermore, it was found that preemption might improve efficiency of the CDT operations.

Yazdani et al. (2015) addressed a truck scheduling problem at a CDT with multiple doors, aiming to minimize the truck service makespan. A Hunting Search Metaheuristic (HSM) was proposed as a solution approach for the formulated model, utilizing three sub-procedures: (1) the moving mechanism; (2) the correcting mechanism; and (3) the repositioning mechanism. The interruption of the unloading process of ITs was not considered. The proposed algorithm was also enhanced by a simple form of Simulated Annealing (SA). Performance of the proposed algorithm was evaluated for the small-size and large-size problem instances. The small-size problem instances were solved with CPLEX. HSM returned the optimal solutions for the majority of the small-size problem instances. A relative percentage deviation (RPD) was used as a performance measure to evaluate the algorithm against Variable Neighborhood Search (VNS) and Differential Evolution (DE) for the large-size problem instances. RPD was calculated based on a relative difference between the average makespan and the minimum makespan, obtained for a given problem instance. The experiment results showed that HSM with the average RPD of

0.96% outperformed the VNS and DE algorithms with the average RPDs of 1.97% and 1.39%, respectively.

Arkat et al. (2016) studied the problem of scheduling ITs and OTs at a CDT with multiple dock doors. The ITs were assumed to dock at the CDT doors, considering their release times. On the other hand, no time constraints were imposed for the precedence of OTs. The objective function of the proposed mathematical model was to minimize the makespan. The authors pointed out that the considered problem had NP-hard computational complexity. The developed model was solved using CPLEX and the SA algorithm. The numerical experiments demonstrated that CPLEX could be used only for the small-size problem instances. However, the SA algorithm was able to yield solutions with good-quality and within a reasonable computational time for both small- and large-size problem instances.

Assadi and Bagheri (2016) addressed the problem of scheduling trucks at a CDT with a particular emphasis on the JIT approach. The study explicitly considered the interchangeability of products, the ready times for the arriving ITs and OTs, and the transshipment time between IDs and ODs. The objective of the presented mathematical model aimed to minimize the earliness and lateness of the departures of the OTs. The population-based SA metaheuristic and DE metaheuristic were developed to solve the problem. The algorithms developed were further compared against CPLEX for the small-size problem instances considered. As for the medium- and large-size problem instances considered, the DE and SA algorithms were compared against pure random search. Although both metaheuristic algorithms were found to be promising throughout the computational experiments, the DE algorithm was superior in terms of the solution quality.

Cota et al. (2016) focused on the scheduling of trucks at a CDT with multiple doors. The problem was formulated as a two-stage hybrid flow-shop problem, considering the basic CDT operational constraints. The objective function of the proposed time-indexed mixed integer linear programming formulation aimed to minimize the makespan. The proposed mathematical model was solved with a developed polynomial-time heuristic, referred to as CDH. The suggested solution approach was compared to the JRH and JLPTH heuristics, which were previously presented by Chen and Song (2009). The numerical experiments indicted that the JLPTH heuristic had better performance for the small-size problem instances (e.g., the problem instances with 2 machines). On the other hand, the CDH heuristic consistently yielded better performance for the large-size problem instances.

Golshahi-Roudbaneh et al. (2017) studied the problem of scheduling trucks at a CDT. The objective of the study was to minimize the makespan and determine the service sequence for both ITs and OTs. The mathematical model, developed by Yu and Egbelu (2008), was used in the study. The authors applied five metaheuristics to solve the problem: (1) Simulated Annealing (SA); (2) EA; (3) Keshtel Algorithm (KA); and (4) Stochastic Fractal Search (SFS); and (5) hybrid SA-PSO. The small-, medium-, and large-size problem instances were created to evaluate performance of the proposed solution approaches for the presented mathematical model. RPD was used as a performance measure for comparison of the algorithms. It was found that SA and SA-PSO had the minimum computational time, while KA and SFS generally had superior performance in terms of the recorded RPD values.

Khalili-Damghani et al. (2017) proposed a new multi-period CDT truck scheduling model with multiple product types, variable truck capacities, due dates, and temporary storage. The objective of the presented mathematical model was to minimize the total operational time (i.e., makespan). An EA-based metaheuristic was deployed to solve the problem. The chromosomes, the operators, and the constraint handling strategy were specifically designed within the proposed algorithm for the considered CDT truck scheduling problem. The numerical experiments were conducted using 90 problem instances, which were further classified into small-, medium-, and large-size problem instances. The developed EA-based algorithm was compared against the exact Branch-and-Bound method (applied within LINGO). It was found that the proposed EA algorithm achieved the optimal/near-optimal solutions, obtained by the exact Branch-and-Bound method, for the small- medium-size problem instances and required lower computational time. Furthermore, the EA algorithm produced good-quality solutions for the large-size problem instances, while the exact Branch-and-Bound method was terminated due to non-convergence after a pre-defined computational time limit.

Luo and Ting (2017) studied a multi-door CDT and considered the problem that involved a simultaneous scheduling of trucks (both ITs and OTs) and door assignment. The study was motivated by the real-world activities of e-commerce providers in Taiwan, where the products are transferred from suppliers to specific customers. The objective of the study was to minimize the total inventory holding cost, which is dependent on the time spent by the products at the CDT. An ACO-based metaheuristic was deployed to solve the problem. The numerical experiments were performed to evaluate effectiveness of the developed solution algorithm against GUROBI. The computational time limit was set to 7,200 sec for GUROBI. A total of 24 problem instances were generated. It was found that both GUROBI and ACO were able to provide the optimal solutions for 15 problem instances. The ACO yielded superior solutions for the remaining problem instances, as GUROBI was not able to provide the optimal solutions within the computational time limit imposed. A multi-objective model that would consider the cost of inventory holding as well as makespan was suggested as potential future research.

Wisittipanich and Hengmeechai (2017) studied a multi-door CDT truck scheduling problem with the objective of minimizing the makespan. An MIP model was formulated in order to perform the truck-to-door assignment and truck sequencing at each CDT door. The developed mathematical formulation was solved using a modified PSO-based algorithm (which was referred to as GLNPSO – the particle swarm optimization with combined g-best, l-best, and n-best social structures). The computational experiments were conducted using 35 problem instances. The performance of the GLNPSO proposed was evaluated based on a conducted comparative analysis against a traditional PSO and exact optimization (LINGO). The results from computational experiments showed that PSO and GLNPSO yielded the solutions with similar objective function values for the small- and medium-size problem instances. However, GLNPSO was generally able to obtain the solutions with smaller makespan values as compared to the PSO algorithm for the large-size problem instances. On the other hand, LINGO was not able to solve the medium-size and large-size problem instances within the time limit imposed (i.e., 8 hours). Furthermore, the proposed GLNPSO algorithm was found to be superior in terms of the convergence patterns when comparing to the traditional PSO algorithm.

Serrano et al. (2017) proposed an MIP model for scheduling the arrival times of ITs (taking into account the soft time window constraints), shop-floor repackaging operations, and departure times of OTs. The objective of the study was to minimize the total penalty cost, associated with the arrival of ITs outside the agreed time periods (that may further result in unbalanced repacking workload). CPLEX was adopted as the solution approach for the proposed mathematical model. The computational experiments were performed using the real-life operational data, which were provided by Renault. The number of variables varied between 18,290 and 82,038, while the number of constraints varied between 31,508 and 99,788. It was found that the computational time, required by CPLEX, was increasing with the problem size.

Behnamian et al. (2018) studied scheduling of ITs and OTs at multiple CDTs, considering truck service breakdowns. The objectives were to minimize the sum of makespans and the total transportation cost at CDTs. A two-stage MIP model was developed for the location of the CDTs in the distribution network and the CDT truck scheduling. The authors proposed four algorithms to solve the problem: (1) EA; (2) SA; (3) DE; and (4) Hybrid EA-SA. The small- and large-size problem instances were generated. While the small-size problem instances were solved using CPLEX, the proposed algorithms were assessed using both small- and large-size problem instances. The solutions, produced by the metaheuristics, did not substantially differ from the optimal ones for the small-size problem instances. RPD was used as one of the performance measures. The results from computational experiments showed that EA-SA had the best performance among the proposed algorithms in terms of the RPD values.

Dulebenets (2018a) developed an MIP model for the scheduling of trucks at a CDT. The objective was the minimization of the total truck service cost (i.e., comprising the total handling cost, total cost of waiting, and the total cost of delay on the departure of trucks), associated with operations at the CDT. The author proposed a Diploid Evolutionary Algorithm (DEA) to solve the model after considering the drawbacks of previous heuristic methods that used a haploidy concept. Based on the diploidy concept, a copy of each parent chromosome was stored before the crossover operation. The numerical experiments were conducted to assess the optimality gap of the DEA. It was found that the optimality gap did not exceed 0.18%. The application of the proposed diploidy concept reduced the total cost of service by 18.17%, which can be considered as a significant effect on profitability.

Dulebenets (2018b) conducted a comprehensive study on the use of strong and weak mutation mechanisms within EAs for the scheduling of trucks at a CDT. The study was based on the MIP model formulation for the scheduling of trucks at a CDT. The objective function aimed to minimize the total weighted truck service cost. Two variations of EA were used to solve the model: one EA applied strong mutation (the number of genes to be changed for each chromosome is fixed), while the other EA applied weak mutation (the number of genes to be changed for each chromosome may vary). The results from conducted numerical experiments showed that the EA with weak mutation outperformed the EA with strong mutation in terms of the solution quality. However, no significant difference in the computational times of the EA algorithms was recorded. Furthermore, the heuristic, developed for initializing the chromosomes and population, was found to be efficient as compared to the commonly used methods.

Ladier and Alpan (2018) studied the problem of scheduling of trucks at a CDT, explicitly modeling the internal pallet handling. The objective function of the presented mathematical model focused on minimization of the total number of pallets put in the storage and the dissatisfaction of the transportation providers. The problem was proved to have NP-hard complexity. A total of three heuristics were deployed to solve the problem. Two heuristic algorithms aimed to solve a set of IPs sequentially, while the third heuristic algorithm relied on TS. A set of computational experiments were conducted to determine the conditions under which a given heuristic algorithm outperforms the others. The conducted numerical experiments demonstrated effectiveness of the proposed methodology, as fairly large-size problem instances were solved within a reasonable computational time (e.g., 150 IDs and 150 ODs).

Wang et al. (2018) studied a CDT with multiple IDs and ODs. The objective of the study was to find the most superior door assignment and truck sequence that minimized the makespan. The problem was formulated as an MIP. Three solution methods were proposed by the authors, which include Harmony Search (HS), Improved Harmony Search (IHS), and EA, to solve the problem. The Taguchi's scheme was used for setting the parameter values for the developed solution algorithms. Nine problem instances were used in the study, divided into the small-size, medium-size, and large-size problem instances. The proposed solution methods were analyzed throughout the computational experiments. The computational results showed that the IHS algorithm generally yielded superior truck schedules as compared to the HS algorithm. The worst performance in terms of the objective function values was recorded for EA, when a pre-defined limit was set for the computational time.

Dulebenets (2019a) proposed a Delayed Start Parallel EA for scheduling the arriving trucks at a CDT. The developed algorithm executed different EAs on its islands sequentially and exchanged the solutions between the active islands using the adaptive migration criterion. The objective function of the proposed model aimed at the minimization of the total cost of service of the trucks, which included the following components: (1) the total handling cost; (2) the total cost of waiting; (3) the total scheduled truck departure time violation cost; and (4) the total CDT storage space utilization cost. The Delayed Start Parallel EA was compared against CPLEX and five other metaheuristic algorithms, which have been commonly used in the CDT literature (i.e., Parallel EA, EA, VNS, TS, and SA). It was found that the solution algorithm had the optimality gaps that were acceptable. Furthermore, the Delayed Start Parallel EA demonstrated a good performance in terms of both computational time and solution quality as compared to the other metaheuristic algorithms. Also, important managerial insights were revealed using the proposed solution algorithm from the transportation perspective.

2.2 Multi-Objective CDT Truck Scheduling

A number of studies have formulated the problem of the scheduling of trucks at a CDT as a mathematical model with multiple objective functions, which are conflicting in their nature. For example, Chmielewski et al. (2009) addressed the truck-to-door assignment problem at less-than-truckload (LTL)-terminals. The study aimed to optimize the total travel distance by the internal handling equipment, which is a major component of the operating cost, and decrease the waiting time for each truck (as an increasing waiting time leads to more crowded yards). The objective functions of the presented two-objective mathematical model were aimed at minimizing the total travel distance by the internal handling equipment and minimize the total truck waiting time. The authors used two approaches to solve the problem: (1) the Decomposition-and-Column-Generation approach; (2) the EA-based algorithm. The first solution approach was implemented within the GAMS environment (CPLEX was used as an optimization solver) in single-objective settings, where one objective was ignored. The numerical experiments were conducted for 10 different problem instances, which were developed by varying certain CDT operational features (e.g., the number of available doors, the number of available time periods, the number of areas for loading/unloading). The EA-based algorithm was found to be efficient in the analysis of tradeoffs between the conflicting objective functions (in contrast to the Decomposition-and-Column-Generation approach, which dealt with one objective function only).

Boloori Arabani et al. (2010) addressed the problem of scheduling trucks at CDT, particularly focusing on the just-in-time (JIT) approach. The JIT concept at a CDT assesses a system performance based on punctuality and exactness of product deliveries. A multi-criteria objective model was formulated. The primary objective of the model was the minimization of the total cost, which was associated with earliness and tardiness of OTs. Three metaheuristics were applied to solve the model, including EA, PSO, and DE. The algorithms were evaluated using two performance measures: least significant difference (LSD) and ANOVA measure (AM). A set of randomly created problem instances (8 in total) were used for the numerical experiments. The results showed that EA was superior to other algorithms in terms of the solution quality, while PSO had the best computational time. However, DE demonstrated the least sensitivity to the problem size.

Golias et al. (2010) studied the problem of scheduling of trucks at a CDT, aiming to assign the arriving ITs and OTs to the available IDs and ODs, respectively. The presented mathematical model had two objective functions, including the following: (1) minimize the total truck service time; and (2) the total cargo stay time (i.e., the product inventory). The storage was assumed to begin before the OT started its service and after the IT left the facility. A framework for the Multi-Objective Memetic Algorithm was presented. The scope of the future research included evaluation of the proposed multi-objective-based solution approach, as well as capturing additional practical constraints in the CDT operations.

Boloori Arabani et al. (2012) also addressed the problem of the scheduling of trucks at a CDT, considering multiple objectives. The objectives were: (1) to minimize the makespan; and (2) to minimize the total lateness of OTs. The authors formulated the scheduling problem as a multi-objective optimization problem. Three multi-objective metaheuristic algorithms were developed to solve the model, namely: (1) Sub-Population Genetic Algorithm-II (SPGA-II); (2) Sub-Population Particle Swarm Optimization-II (SPPSO-II); and (3) Sub-Population Differential Evolution Algorithm-II (SPDE-II). Four performance measures that have been widely used in the literature were adopted to evaluate the developed algorithms: (1) hyper-area measure (HAM); (2) spacing measure (SM); (3) mean ideal distance measure (MIDM); and (4) rate of achievement to two objectives simultaneously measure (RASM). The conducted numerical experiments involved 10 randomly created problem instances. Additional 27 conditions with various algorithmic parameter values were considered for each problem instance throughout the parameter selection analysis. The results showed that SPPSO-II was superior in terms of the solution quality, while SPGA-II was superior in terms of the computational time.

Golias et al. (2012) examined the problem of scheduling ITs to IDs at a CDT with two objectives: (1) minimizing the total service time of trucks; and (2) minimizing the delayed completion of truck service. A bi-objective model and a bi-level mixed integer model were formulated to capture the objectives. Two solution approaches were developed to solve the formulated models. A Pareto-based EA was used to solve the bi-objective model, while the k-th best algorithm was used to solve the bi-level model. The computational examples were used to evaluate the proposed algorithms. The findings from conducted numerical experiments showed that both approaches were applicable for the problem of IT scheduling at the CDT. Also, the proposed models could be applied to other related scheduling problems.

Naderi et al. (2014) considered a bi-objective problem of truck scheduling at a CDT with temporary storage. The following objective functions were considered: (1) minimize the makespan; and (2) minimize the total late departures of trucks. The Iterated Greedy Algorithm (IGA) was proposed for the bi-objective scheduling problem. The developed solution algorithm was compared against SPPSO-II and SPEA-II, which were used by Boloori Arabani et al. (2011, 2012). The authors employed different performance measures to compare the solution approaches, such as Dominance Ranking, Quality Indicator, Empirical Attainment Function, Hypervolume Indicator, and Unary Epsilon Indicator. The numerical experiments were performed using 20 problem instances. The IGA algorithm was found to be superior to the SPPSO-II and SPEA-II algorithms.

Amini and Tavakkoli-Moghaddam (2016) presented a bi-objective mathematical model for scheduling trucks at a CDT, considering potential truck service breakdowns. The study assumed that the number of breakdowns in a unit of time was Poisson distributed. The reliability function of a CDT was developed based on the IT and OT service breakdowns. The first of the two objectives of the model minimized the total tardiness of OTs, while the second objective maximized the system reliability (by minimizing the total weighted completion time, which was influenced by the defect rates). Three metaheuristics were proposed in that study to solve the multi-objective model, including Multi-Objective Simulated Annealing (MOSA), Non-dominated Sorting Genetic Algorithm II (NSGA-II), and Multi-Objective Differential Evolution (MODE). Five metrics were used to judge efficiency of the metaheuristics: (1) Quality metric (QM); (2) Spacing metric (SM); (3) Generational distance (GD); (4) Diversity metric (DM); and (5) Maximum Pareto Front error (MPFE). The results from numerical experiments showed high-quality performance of all the algorithms. MODE had the best performance of the three algorithms when considering all the five metrics, while MOSA produced the solutions of an acceptable quality and required the lowest computational time.

Fazel Zarandi et al. (2016) proposed a new multi-criteria model with nonlinear terms and integer variables for the problem of scheduling of trucks. The objective was to minimize the total earliness, tardiness, and preemption (i.e., a temporary service interruption of a currently docked truck by another truck) of OTs at the CDT. It was identified that it is practically impossible to simultaneously eliminate earliness and tardiness cost; however, minimizing them is rather feasible. The formulated model was mapped to a constraint satisfaction problem (CSP) and integer programming (IP) problem. The problem instances were created to perform comparison between IP, CSP, and EA. The results showed that IP produced good-quality solutions for the small-size problem instances. Furthermore, CSP was more effective and efficient in finding solutions for the small-size and medium-size problem instances, but a relatively poor performance was observed for the large-size problem instances. EA was found to be the most efficient solution approach for the large-size problem instances in terms of the solution quality and the computational time.

Goodarzi et al. (2018) formulated a mixed integer nonlinear model to address the problem of shipment pick-up and delivery at a CDT. The model had two objectives: (1) to minimize the total cost of transportation; and (2) to minimize to total tardiness and earliness that were associated with visiting retailers. The total transportation cost components involved the cost of a direct shipment from suppliers to retailers, indirect cost of shipment via the CDT, and the total operational cost of vehicles transporting the cargo. A Multi-Objective Imperialist Competitive Algorithm (MOICA) was proposed to solve the model. MOICA is considered as an optimization technique, which is inspired by the socio-political evolution of humans. The numerical experiments in that study considered three scenarios: (1) direct shipment; (2) indirect shipment (through the CDT); and (3) combination of direct and indirect shipments. The results from numerical experiments showed that MOICA performed better than NGSA-II and Pareto Archived Evolution Strategy (PAES) in terms of the solution quality. Other metrics used to evaluate performance of MOICA against NGSA-II and PAES included QM, MIDM, diversification metric (DM), and SM. These metrics also showed that MOICA was superior to the NGSA-II and PAES algorithms.

Kargari Esfand Abad et al. (2018) presented a bi-objective mathematical model for the pick-up and delivery pollution-routing problem, where the shipments were transferred from suppliers to a CDT for integration and consolidation and then delivered to the final customers. The first objective function aimed to minimize the total system cost, including the transportation costs for pick-up and delivery, setup and operational costs at the CDT, monetary savings from integration and consolidation at the CDT, and driver cost. The second objective function aimed to minimize the total fuel consumption of the vehicles that transport the products between suppliers, the CDT, and customers. Three multi-objective metaheuristics were used as solution approaches, including the following: (1) Non-dominated Ranking Genetic Algorithm (NGRA); (2) NSGA-II; and (3) Multi-Objective Particle Swarm Optimization (MOPSO). The computational experiments were conducted using 20 problem instances, which were decomposed into the small-, medium-, and large-size problem instances. The developed solution algorithms were compared in terms of NPS, ER, CPU, MID and DM. The results from a statistical analysis showed that there was a significant difference in certain comparative criteria at least between two solution algorithms. The findings also highlighted the advantages of the CDT concept over the classical approach, where the products are directly transferred from suppliers to customers.

2.3 Uncertainty Modeling in CDT Truck Scheduling

Tang and Yan (2010) proposed a model for a typical CDT, considering two types of operations: (a) the pre-distribution (Pre-C) operations; and (b) the post-distribution (Post-C) operations. In case of the Pre-C operations, the manufacturer is responsible for the product distribution preparation (i.e., attaching bar codes, labeling, and pricing items) before delivering the products to the CDT. On the other hand, in case of the Post-C operations, the product distribution preparation is performed at the CDT. The objective of the presented mathematical model focused on minimization of the total cost, which included the inventory holding cost, the inventory shortage cost, and the transshipment cost. The numerical experiments were conducted in order to evaluate the Pre-C and Post-C operations, considering various operational factors, such as demand uncertainty, operational cost at the CDT, unit inventory holding and shortage costs. It was found that the Pre-C operations lowered the operational cost at the CDT, but led to higher transshipment cost for the cases with higher demand uncertainty. Moreover, the Post-C operations resulted in higher operational cost at the CDT, but reduced the transshipment cost for the cases with lower demand uncertainty. The parameter values, including the demand variance, the unit inventory holding cost, the unit shortage cost, and the unit transshipment cost, could significantly influence the tradeoffs, associated with both Pre-C and Post-C operations.

Sathasivan (2011) studied the CDT truck scheduling problem, considering a stochastic nature of the handling time of trucks. The objective function of the proposed mathematical model focused on minimization of the total weighted completion time of trucks. A surrogate heuristic was developed in order to solve the problem. Furthermore, two variations of the EA algorithm were presented as well. The numerical experiments were conducted using the realistic operational data, collected from Central Texas Goodwill Industries and H-E-B Grocery Industry. The solutions, provided by the surrogate heuristic, were found to be conservative. A two-space EA, which relied on the bounds that were derived by the surrogate heuristic, was superior to a typical EA algorithm. The proposed methodology was found to be effective and can be used to assist the CDT managers with the truck-to-door scheduling. Moreover, the proposed methodology can be used to ensure timely delivery of the products to retail stores.

Konur and Golias (2013a) studied the problem of the scheduling of trucks, with a focus on the uncertainty in the arrival time of the trucks at a CDT. The objective of the study was to determine the IT schedule that minimizes the total service time of ITs at IDs. Four scheduling approaches were evaluated in that study: (1) deterministic approach; (2) optimistic approach; (3) pessimistic approach; and (4) hybrid approach. The deterministic approach ignored the tendencies of early and late arrival of trucks, the optimistic and pessimistic approaches considered the two extremes of the problem, while the hybrid approach was based on the expected truck arrival times obtained from optimistic and pessimistic approaches. The original truck scheduling model was reformulated as a bi-level optimization problem for the cases with optimistic and pessimistic arrival of trucks. EA was proposed to solve the problem under deterministic approach and was modified to solve the problem under the optimistic and pessimistic approaches. The numerical experiments were further undertaken to evaluate all the considered truck scheduling approaches under uncertain truck arrival times. A CDT with 10 IDs and 300 arriving ITs was modeled. The results showed that the hybrid scheduling approach outperformed the alternative scheduling approaches and was the most effective against the truck arrival time uncertainty.

Konur and Golias (2013b) addressed the problem of IT scheduling strategy at a CDT, considering the variation in the cost due to uncertainty in truck arrival time. The objective was to determine a cost-stable scheduling strategy while minimizing the total truck service cost. The problem was formulated as a bi-objective bi-level optimization problem, which was referred to as a Stable Scheduling Problem (SSP). The first objective function was to minimize the average total truck service cost, while the second objective function was to minimize the cost range. The Random Search Heuristic (RSH) was proposed to solve the Single Door Range Bound Problem (RBP). RSH was found to be more efficient than CPLEX in terms of the computational time, and the difference in the average objective function values was ~1%. A Genetic Algorithm-based Search Heuristic (GASH) was proposed to obtain an efficient Pareto frontier. The computational experiments were further undertaken to evaluate the performance of the presented approach. The performance was assessed with two different FCFS policies: (1) FCFS with start times; and (2) FCFS with finish times. The proposed cost-stable approach outperformed the FCFS policies and produced the results with a lower cost range.

Walha et al. (2014) conducted a comprehensive review of the literature on uncertainties in the CDT operations. Uncertainties were categorized by the authors as follows: (1) external uncertainties; (2) internal uncertainties; and (3) a combination of internal and external uncertainties. The internal uncertainties were related to the truck departure times, the handling time of trucks, the available handling resources (e.g., forklifts, conveyor belts), and the number of available trucks (e.g., the information regarding the position and number of OTs upon arrival of ITs). The external uncertainties were related to the truck arrival times, the number of inbound trailers, and the freight flow/content of trucks. The study concluded that there were only a very limited number of papers, considering uncertainties in the CDT operations. It was pointed out that the future works should focus on the following directions: (a) development of the mathematical models that capture various sources of uncertainties (both internal and external); (b) consideration of conflicting decisions within a multi-objective framework; and (c) consideration of realistic operational assumptions.

Ladier and Alpan (2016) proposed a robust model for the scheduling of trucks at a CDT with considerations for time windows. It was pointed out that the truck arrival times, the truck loading times, and the truck unloading times are often subject to various uncertainties. The study aimed to obtain a truck schedule that remained feasible and stable with the uncertainties considered. The objective of the deterministic model minimized the dissatisfaction of the transportation providers for ITs and OTs and the total number of pallets transiting through the storage. A total of nine robust optimization techniques were used in the study, including the following: (1) the minimax method; (2) the minimization of expected regret; (3) three models of resource redundancy; and (4) four models of time redundancy. Three robustness metrics were used to evaluate the formulated models: (1) robustness subject to variation in transfer time; (2)

robustness subject to variation in unloading time; (3) robustness subject to variation in truck arrival time. The exact optimization approach was used for the small-size problem instances, while heuristics were proposed for the large-size problem instances. The results from numerical experiments showed that the redundancy models were able to provide the most robust schedules. However, the redundancy models generally have higher implementation cost for the CDT managers.

Rajabi and Shirazi (2016) focused on the problem of scheduling ITs and OTs at a CDT, considering the uncertain nature of truck arrival times. The objective function of the proposed mathematical model focused on minimization of the total truck waiting time. Chance-constrained mathematical programming was used to capture the uncertain truck arrival times in the proposed mathematical model. The problem was solved using VNS. The numerical experiments were conducted for the small-size and large-size problem instances. VNS was compared against the exact optimization algorithm for the small-size problem instances. The optimality gap of the proposed algorithm did not exceed ~12% for the small-size problem instances. However, the exact optimization algorithm was not able to solve the large-size problem instances within a reasonable computational time. On the other hand, the developed VNS algorithm was able to obtain the solutions for the large-size problem instances within less than 2,300 sec.

2.4 Miscellaneous

Montana (1998) discussed the implementation of EAs for solving scheduling problems. Scheduling problems were described as difficult for several reasons: (1) the size and complexity of the solution space; (2) scheduling is a dynamic process, i.e. the results are only valid for a limited amount of time; (3) the solutions of different variations of the scheduling problem are required by different domains and applications. The author, therefore, proposed an EA as an approach that can efficiently find competitive solutions in a large and complex search space. According to the author’s findings, the EAs made the scheduling problems practically solvable, where it was previously impractical. In addition, the EAs were found to be promising as compared to the other common scheduling techniques, which have the following drawbacks: (1) limited success on large real-world scheduling problems; and (2) lack of flexibility. Other problems, where the EAs had been applied, included: (1) job-shop scheduling; (2) scheduling computing tasks; (3) scheduling laboratory equipment; (4) crew scheduling; (5) maintenance and rehabilitation scheduling; and (6) talent and project scheduling.

Bartholdi and Gue (2004) identified a set of common CDT shapes (such as I, L, and T) and a set of uncommon CDT shapes (such as U, H and E). The objective of the study was to find the best CDT shape. The authors conducted a computational experiment on the various CDT shapes in existence. Furthermore, the X-shape was also proposed. Some of the explored CDT characteristics included size, layout, concentration of flows, and a fraction of doors devoted to receiving. The findings showed that the efficiency of a CDT shape varies with the size. The I-shape was found to be the most efficient for the CDTs with less than 150 doors. The T-shape was found to be the best for the CDTs with intermediate size of 150 – 200 doors, while the X-shape was the most suitable for the CDTs with more than 200 doors.

Li et al. (2008) conducted a study to assist with product allocation for cross-docking and warehousing operations in a Fast-Moving Consumer Goods (FMCG) supply chain. The objective function of the presented linear programming formulation aimed at the minimization of the total cost of distributing various products between CDTs and warehouses. The Vogel's Approximation Method (VAM) and the UV method were applied to solve the problem. The performance of the proposed methodology was evaluated using a set of numerical experiments. It was found that the majority of the same product types were allocated to the CDTs and warehouses separately. Also, the importance of the product arrival times and requested departure times was highlighted at the operational level.

Ross and Jayaraman (2008) conducted a study on new heuristic solutions for CDT locations in a supply chain. The study considered central manufacturing plant sites, multiple product families, multiple CDTs, and retail outlets. Two mathematical models were formulated (i.e., strategic and executional models). The strategic model minimized the total cost, associated with opening warehouses and CDTs, transporting multiple products from warehouses to CDTs, and supply of multiple products to customer zones. The executional model minimized the total cost, associated with supply of products from CDTs to customer zones, transport of products from warehouses to

CDTs, and delivery of products to warehouses. Two SA-based heuristics were proposed to solve the proposed mathematical models, including the following: (1) TABU-SA; and (2) RESCALE-SA. The numerical experiments demonstrated that integration of the traditional SA method with Tabu Search was found to be promising for designing supply chains with CDTs. The developed TABU-SA heuristic was able to provide high-quality solutions within a reasonable computational time.

Agustina et al. (2010) conducted an extensive review of the mathematical models for CDT planning. The models were classified into three categories, including the following: (1) operational level – involved the models that considered scheduling, CDT door assignment, transshipment, and vehicle routing; (2) tactical level – involved the models that considered the CDT layout design; and (3) strategic level – involved the models that considered the CDT network design. The authors noted that there is a lack of the models that can investigate and determine the best strategy for distribution planning. It was also suggested that the new models should consider certain important factors, such as investment cost for new CDTs, penalty cost for closing CDTs, expansion planning, service level target, market share target, the profitability of particular strategies, and others.

Van Belle et al. (2012) conducted an exhaustive review of the state-of-the-art with a primary emphasis on the cross-docking concept. Some advantages of CDTs were stated, such as (1) distribution cost reduction (including inventory, handling, labor, and warehousing costs), (2) shorter delivery lead time, (3) reduced storage space, (4) reduced risk of loss and damage, and others. CDTs were categorized based on physical, operational, and flow characteristics. The physical characteristics consisted of the shape, number of doors, and internal transportation. The operational characteristics included the service mode and allowance for preemption. On the other hand, the flow characteristics included truck arrival patterns, departure time, product interchangeability, and temporary storage. The survey presented a detailed review of the previously conducted studies, focusing on the CDT attributes (e.g., shape, number of doors, internal transport, door service mode, truck arrival pattern, etc.), mathematical formulations, adopted objective functions, and developed solution approaches. The identified solution approaches included EA, SA, VNS, TS, PSO, ACO, DE, among the others. The most common objectives, which were used in the presented mathematical models, were the following: (a) minimization of the makespan; (b) minimization of the total cost; (c) minimization of the maximum workload; and (d) minimization of the variance, associated with the distribution of idle times. A number of the future research opportunities have been discussed, including the following: (1) modeling of various CDT layouts and storage space area design, (2) evaluation of different methods for internal transport; (3) modeling CDT operations with preemption; (4) modeling CDT operations with interchangeable goods; (5) consideration of the real-world assumption and practices, etc.

Shuib and Fatthi (2012) conducted a comprehensive study on quantitative approaches for the CDT door assignment problem (i.e., assignment of origins to IDs and assignment of destinations to ODs). A significant number of studies were reviewed in order to identify the gap in knowledge. The quantitative approaches, which were found in the literature, included the following: Integer Programming (IP), Linear Programming (LP), Mixed Integer Programming (MIP), Nonlinear Programming, Multi-objective Optimization, Statistics, Quadratic Programming, Simulation, Flowshop model, Exact Optimization, Heuristics, and Metaheuristics. The most common objective function, found in most of the reviewed studies, was the minimization of total operational time, otherwise known as makespan. The authors presented a set of the future research extensions, which included the following: (1) truck assignment to the CDT doors after solving the CDT door assignment problem; (2) consideration of internal handling operations inside the CDT; (3) consideration of potential congestion issues inside the CDT; (4) deployment of the forklift operators for the internal product transport; and others.

Wang and Li (2012) proposed a dynamic product quality evaluation-based pricing model, which can be applied to the supply chain of food products. The objective was to minimize the food spoilage and maximize the total profit of retailers. The quality deterioration of perishables was modeled based on the initial product quality and the deterioration rate, which was assumed to be an exponential function of temperature. A pricing model was developed in order to determine the optimal discount of a food product as it approaches its expiration date. The purpose of introducing discounts was to stimulate the demand, increase the total revenue, and reduce waste of perishable products. The numerical experiments were further undertaken to evaluate the proposed methodology for two perishable product types (i.e., meat products and vegetable products) and four retail stores. The results highlighted the importance of dynamically tracking the quality of perishable products in order to ensure business competitiveness and avoid potential monetary losses.

Naskaris et al. (2014) highlighted the importance of effective logistics procedures for food supply chains. The study focused on evaluation of the cross-docking concept for distribution of the products from local producers to a set of grocery stores. A cost-to-serve model was developed in order to calculate the costs, associated with delivery of the products to the grocery stores. It was found that the product handling at a CDT incurred an additional time, but ultimately would allow increasing the total profit. Moreover, when the food hub (or producer) is required to supply the products to more than 6 grocery stores, a combination of the cross-docking concept and the backhaul arrangement with the regional distribution center would be an efficient alternative.

Ladier and Alpan (2016) conducted a comprehensive review of the existing literature on the CDT truck scheduling. The study comprised of a literature survey, interviews with the CDT managers, and a direct observation of industry practices. Some of the CDT operational issues, faced in industry, include management of delayed trucks, scheduling of ITs and OTs, forecasting volume of activities, etc. According to the conducted field survey, the essence of preemption was not well understood by the CDT managers interviewed. Therefore, the concept of preemption was not used in any of the CDT platforms considered in the study. Most of the existing CDT models focused on minimizing the makespan, while the industry practitioners focused more on the door utilization rates. These gaps between theory and practice were pointed out as the directions for the future research.

Shiekholeslam and Emamian (2016) conducted a review of several studies in the area of CDT operations. A number of the key decision problems were highlighted, such as the CDT shape selection, allocation of handling resources, temporary storage allocation, truck scheduling, CDT door assignment, and others. The collected studies on the CDT operations were classified into two major categories, including the following: (1) the cross-cocking system as a subset of a supply chain; and (2) internal operations at the CDT. A large variety of solution approaches have been deployed to assist the CDT managers with operations planning, including EA, TS, DE, ACO, PSO, VNS, and NSGA-II, among others. As a result of the conducted survey, it was found that only a few studies focused on determination of the appropriate CDT shape, determination of the appropriate CDT layout, and selection of the appropriate CDT dimensions. The study pointed out that the future research efforts should consider temporary storage operations at CDTs. Also, the need for holistic optimization models, which could tackle several CDT operational decisions at a time, was underlined.

Sreedevi and Saranga (2017) pointed out that many firms are trying to expand the number of products they offer, which may lead to a high level of uncertainties in their supply chains. High environmental uncertainties may further result in supply disruptions, production delays, and delivery delays, which may negatively affect performance of the entire supply chain. The study aimed to investigate different types of risk, faced by various firms, and identify the conditions that could mitigate those risks. The authors used the data from the International Manufacturing Strategy Survey. The following three types of risk were considered as significant: (1) supply risk; (2) manufacturing process risk, and (3) delivery risk. It was found that uncertainties in the supply chain operations could further lead to a higher level of the supply chain risk. The results showed that certain emerging markets would not be able to reduce the supply chain risk with internal capabilities alone.

Wang et al. (2017) focused on the vehicle routing problem with CDTs, where multiple suppliers, multiple retailers, and split deliveries were considered. An MIP model was formulated to address the problem. The objective of the mathematical formulation was the minimization of the fixed cost of vehicles and the total travel cost. Two heuristic algorithms were proposed as the solution approaches to address the medium and large-size problem instances. The first heuristic combined a construction heuristic (CH) with a two-layer SA (CH-TLSA), while the second heuristic combined a construction heuristic (CH) with a two-layer TS (CH-TLTS). The first layer aimed to optimize the allocation of trucks to the CDTs, while the second layer aimed to optimize the visiting order of suppliers and retailers. The developed solution methodology was compared against CPLEX throughout the computational experiments. The findings from computational experiments showed that the CPLEX computational time was increasing exponentially with increasing problem size. Furthermore, CH-TLTS had better performance for the medium-size problem instances, while CH-TLSA had better performance for the large-size problem instances.

Yang et al. (2017) underlined the challenges of managing perishable food supply chains due to short lifetime of the products, potential product deterioration, and retail demand uncertainty. The study aimed to investigate the joint decisions regarding the pricing strategies, shelf space allocation, and replenishment policy for a perishable product, aiming to maximize the retailer's total profit. A stochastic nature of the retail demand was explicitly captured. A set of algorithms were further designed to solve the problem considered in single-item and multi-item food supply chain settings. The conducted numerical experiments demonstrated that the discount rate was significantly influenced with the other operational decisions, but its optimal value could be estimated. Furthermore, the product deterioration and demand uncertainty increase complexity of the supply chain operations.

He et al. (2018) presented a comprehensive literature survey on quality and operations management problems in food supply chains. The collected studies were classified into 4 categories, including the following: (1) storage problems; (2) distribution problems; (3) marketing problems; and (4) food traceability and safety problems. The study indicated that there is lack of approaches that model uncertainties in food supply chain operations. Moreover, the future research should focus on modeling various disruptive events in food supply chain operations (e.g., adverse weather, vehicle breakdowns, and accidents during the transportation process). Also, more realistic assumptions should be adopted in the future studies, dealing with food supply chains, in order to ensure accuracy of the outcomes.

Margolis et al. (2018) presented a multi-objective optimization model, aiming to design a resilient supply chain network. It was pointed out that supply chains generally evolve over time. Certain supply chains may expand as a result of planned construction and/or corporate mergers and acquisitions, while other supply chains may contract due to partnership terminations and various cost-cutting decisions. The following objectives were considered: (1) minimize the total network cost; and (2) maximize the overall supply chain network connectivity. The problem was solved using the ε-constraint method. An illustrative case study was presented in order to showcase applicability of the proposed methodology. The developed algorithm was found to be effective and could assist the decision makers with the analysis of tradeoffs between the total network cost and the overall supply chain network connectivity.

Schwerdfeger et al. (2018) considered the scheduling of JIT deliveries from the CDT toward the original equipment manufacturer (OEM) plant. The objective of the study was to minimize the size of the truck fleet required to deliver all part containers within their given JIT, considering truck capacities. Two mixed integer programs (MIPs) were provided to model the problem (JIT-VS1 and JIT-VS2), with the objective functions that minimized the truck fleet size. Lower bounds procedures (LB1 and LB2) were used to evaluate the quality of the solutions. A heuristic Decomposition Procedure (DEC) was derived to solve the two MIP models by decomposing them into two sub-problems. The JIT-VS1 and JIT-VS2 models were solved using Gurobi. Findings from the computational study showed that JIT-VS1 was only applicable to the small-size problem instances, while JIT-VS2 was capable of handling the large-size problem instances. Three experiments were conducted to test the influence of certain factors on the truck fleet size: (1) Influence of increasing distance between the CDT and the factory on the truck fleet size; (2) Influence of JIT interval size; and (3) Influence of the container standardization. It was shown that the fleet size was impacted by the aforementioned three factors. Increasing distance between the CDT and the factory and a strict application of the JIT principle increased the truck fleet size. Also, an increasing number of equal-size containers required more space in the trucks, which further required deployment of more trucks.

Xin et al. (2018) proposed a new methodology for modeling and controlling the quality degradation of perishable foods. An exponential food quality degradation function was adopted for modeling the food quality degradation. The function was based on the time factor and the temperature factor. The nonlinear quality degradation function was approximated using the piecewise affine (PWA) modeling representation. The PWA model was further transferred into the computational mixed logical dynamic model. Using the latter model, an optimal control policy was presented, which could assist with the analysis of tradeoffs between the food quality and associated energy consumption. The simulation experiments indicated that the proposed methodology allowed reducing the energy consumption by 16% and 5% at the final quality values of 0.80 and 0.90, respectively.

2.5 Summary of findings

The summary of the studies reviewed is presented in Table 2. Specific information extracted from each study includes: (1) author(s); (2) year of publication; (3) study objective(s); (4) objective component(s) (where applicable); (5) solution method(s); and (6) important notes. This review comprises of studies on the CDT truck scheduling published from the year of 1998 to date. Some of the common objectives, found in most of the studies, include minimization of the makespan, storage, total truck service cost, early and late departures. Many of the studies relied on EA as solution approach to solve the CDT truck scheduling model. Other solution approaches found include HEA, k-th best, HSM, MOSA, MODE, NGSA-II, IP, DEA, ACO, DE, TS, PSO, SPGA-II, SPPSO-II, SPDE-II, MOICA, and PAES. These solution algorithms are found to perform better than exact optimization approaches in terms of the quality of their solutions and the computational time for the large-size problem instances.

Table 2: Literature survey summary.

	Author(s) (year)	Objective	Objective component(s)	Solution Method	Notes
			General CDT truck scheduling		
1	Yu (2002)	Minimize the makespan	T	Exact optimization, Heuristics	CDT truck scheduling with and without preemption
2	Li et al. (2004)	Minimize the total cost	ED, LD	HEAs, CPLEX	Earliness and tardiness in truck service completion
3	Lim et al. (2005)	Minimize the total transportation cost, minimize the total cost of product storage	Z	Polynomial-time algorithm	The CDT transshipment problem with different schedule types
4	McWilliams et al. (2005)	Minimize the makespan	T	SBSA, ARB	Scheduling ITs at a parcel hub
5	Ley et al. (2007)	Minimize the total travel distance by the handling equipment	TD	EA, Exact optimization	Scheduling ITs and OTs at a CDT
6	Song and Chen (2007)	Minimize the makespan	T	JRBH, DJRH, CPLEX	Two-stage CDT scheduling with multiple ITs and one OT
7	Shakeri et al. (2008)	Minimize the makespan	T	N/A	Scheduling ITs and OTs at a CDT
8	Vis and Roodbergen (2008)	Minimize the total travel distance by the handling equipment	TD	Heuristic	CDT storage location assignment
9	Yu and Egbelu (2008)	Minimize the makespan	T	Enumeration, Heuristics	Scheduling ITs and OTs at a CDT
10	Chen and Lee (2009)	Minimize the makespan	T	Exact optimization, Heuristic	Scheduling ITs and OTs at a CDT
11	Chen and Song (2009)	Minimize the makespan	T	JRH, JLPTH, DJRH, DJLPTH, CPLEX	Two-stage CDT scheduling, where the second stage jobs can be processed after completing the first stage jobs
12	Maknoon and Baptiste (2009)	Minimize the total travel distance by the handling equipment	TD	DP, Heuristic	Scheduling ITs and OTs at a CDT
13	McWilliams (2009)	Mnimize the makespan	T	EA, SBSA, ARB	Scheduling ITs at a parcel hub
14	Miao et al. (2009)	Minimize the total cost	Z	EA, TS, CPLEX	Considered penalty due to unfulfilled shipments

15	Boysen (2010)	Minimize the total flow time, the total processing time, and the total late departures	FL, HT, LD	DP, SA	Considered a zero-inventory CDT for the food industry
16	Forouharfard and Zandieh (2010)	Minimize the total number of products passing through the temporary storage	MSC	ICA, EA	CDT with temporary storage
17	Vahdani et al. (2010)	Minimize the total flow time	FL	TS, HEA, Heuristic	Scheduling ITs and OTs at a CDT
18	Vahdani and Zandieh (2010)	Minimize the makespan	T	EA, TS, SA, VNS, Heuristic	Scheduling ITs and OTs at a CDT
19	Alpan et al. (2011)	Minimize the total cost of product storage and the total truck replacement cost	ST, MSC	Heuristics, DP	CDT transshipment scheduling
20	Boloori Arabani et al. (2011)	Minimize the makespan	T	ACO, DE, EA, PSO, TS	Scheduling ITs and OTs at a CDT
21	Choy et al. (2012)	Minimize the makespan	T	Heuristic, EA	Models a space-constrained CDT, where the number of incoming ITs exceeds the number of available doors
22	Liao et al. (2012)	Minimize the makespan	T	DE, HDE-1, HDE-2	Scheduling ITs and OTs at a CDT
23	Liao et al. (2013)	Minimize the total weighted tardiness of products	LD	SA, TS, ACO, DE, HDEs	Scheduling ITs at a CDT
24	Gelareh et al. (2013)	Minimize the total cost	Z	CPLEX	Considered penalty due to unfulfilled shipments
25	Golias et al. (2013)	Minimize the total cost	ED, LD, HT, WT	HEA	Scheduling ITs and OTs at a CDT
26	Joo and Kim (2013)	Minimize the makespan	T	CPLEX, EAs	Scheduling ITs and OTs at a CDT
27	Kuo (2013)	Minimize the makespan	T	VNS, SAs	Scheduling ITs and OTs at a CDT
28	Agustina et al. (2014)	Minimize the total truck service cost	ED, LD, TC, ST	CPLEX	Vehicle routing and CDT truck scheduling
29	Guignard et al. (2014)	Minimize the total travel distance	TD	Heuristic	Static and dynamic approaches for the truck-to-door assignment
30	Madani-Isfahani et al. (2014)	Minimize the makespan	T	SA, FM	Multiple CDT truck scheduling with temporary storage
31	Morais et al. (2014)	Minimize the total transportation cost	TC	Heuristics,	Vehicle routing and CDT truck scheduling

				CPLEX	
32	Shiguemoto et al. (2014)	Minimize the makespan	T	HEA, FM, Heuristic, CPLEX	Scheduling ITs and OTs at a CDT
33	Yan (2014)	Minimize the makespan	T	CPLEX, LS, SA, LNS, BS	Scheduling ITs and OTs at a CDT
34	Cekala et al. (2015)	Minimize the total waiting time	WT	EA, PSO, Heuristic	Penalties for violation of the operational constraints
35	Mohtashami (2015)	Minimize the makespan	T	EA	Scheduling ITs and OTs at a CDT; preemption allowed
36	Yazdani et al. (2015)	Minimize the makespan	T	HSM, VNS, DE, CPLEX	Scheduling ITs and OTs at a CDT
37	Arkat et al. (2016)	Minimize the makespan	T	CPLEX, SA	Scheduling ITs and OTs at a CDT
38	Assadi and Bagheri 2016)	Minimize the total early and late departures	ED, LD	DE, SA, CPLEX	Scheduling ITs and OTs at a CDT
39	Cota et al. (2016)	Minimize the makespan	T	CDH, JRH, JLPTH	Scheduling ITs and OTs at a CDT
40	Golshahi-Roudbaneh et al. (2017)	Minimize the makespan	T	EA, KA, SA, SFS, SA-PSO	Scheduling ITs and OTs at a CDT
41	Khalili-Damghani et al. (2017)	Minimize the makespan	T	EA, Exact optimization	Scheduling ITs and OTs at a CDT
42	Luo and Ting (2017)	Minimize the total cost	ST	ACO, Exact optimization	Scheduling ITs and OTs at a CDT
43	Wisittipanich and Hengmeechai (2017)	Minimize the makespan	T	GLNPSO, PSO, Exact optimization	Scheduling ITs and OTs at a CDT
44	Serrano et al. (2017)	Minimize the total cost	Z	CPLEX	Scheduling ITs and OTs at a CDT; penalties for violation of the agreed IT arrival time periods
45	Behnamian et al. (2018)	Minimize the total transportation cost and the sum of makespans at CDTs	T, TC	EA, SA, DE, EA-SA	Location-allocation and truck scheduling at CDTs
46	Dulebenets (2018a)	Minimize the total cost	HT, WT, LD	CPLEX, EAs	Diploid Evolutionary Algorithm for truck scheduling at a CDT
47	Dulebenets (2018b)	Minimize the total weighted truck	HT, WT, LD	CPLEX, EAs	Scheduling ITs and OTs at a CDT

		service cost			
48	Ladier and Alpan (2018)	Minimize the dissatisfaction of the transportation providers for ITs and OTs and the total number of pallets transiting through storage	MSC	Heuristics, TS	Scheduling ITs and OTs at a CDT; internal pallet handling
49	Wang et al. (2018)	Minimize the makespan	T	HS, IHS, EA	Scheduling ITs and OTs at a CDT
50	Dulebenets (2019a)	Minimize the total cost	WT, HT, ED, LD, ST	CPLEX, DSPEA, PEA, EA, VNS, TS, SA	Scheduling ITs and OTs at a CDT
		Multi-objective CDT truck scheduling			
1	Chmielewski et al. (2009)	Minimize the travel distance by the handling equipment and truck waiting time	TD, WT	EA, CPLEX	The Decomposition-and-Column-Generation approach, deployed within CPLEX, dealt with one objective function only
2	Boloori Arabani et al. (2010)	Minimize the total cost	ED, LD	DE, EA, PSO	Earliness and tardiness in truck service completion
3	Golias et al. (2010)	Minimize the total truck service time and storage time	WT, HT, ST	HEA	Bi-objective CDT truck scheduling with storage
4	Boloori Arabani et al. (2012)	Minimize the makespan and the total late departures	T, LD	SPGA-II, SPPSO-II, SPDE-II	Scheduling ITs and OTs at a CDT
5	Golias et al. (2012)	Minimize the total truck service time and the total late departures	LD, HT, WT	EA, k-th best	Bi-objective and bi-level mathematical models for the CDT truck scheduling
6	Naderi et al. (2014)	Minimize the makespan and the total late departures	T, LD	Heuristic, SPPSO-II, SPEA-II	Bi-objective CDT model with temporary storage
7	Amini and Tavakkoli-Moghaddam (2016)	Minimize the total late departures and maximize reliability	LD, FT	MOSA, MODE, NSGA-II	Bi-objective scheduling using Pareto solutions and multi-objective heuristics
8	Fazel Zarandi et al. (2016)	Minimize the total early departures, the total late departures, and preemption	ED, LD, MSC	Exact optimization, EA	Multi-criteria model with nonlinear terms and integer variables
9	Goodarzi et al. (2018)	Minimize the total transportation cost, the total early departures and the total	ED, LD, TC	MOICA, NSGA-II, PAES	Evaluation of various policies to reduce the number of unnecessary stops at a CDT

		late departures of visiting retailers			
10	Kargari Esfand Abad et al. (2018)	Minimize the total cost and the total fuel consumption	TC, FC	NRGA, NSGA-II, MOPSO	Pick-up and delivery pollution-routing problem with a CDT
			Uncertainty modeling in CDT truck scheduling		
1	Tang and Yan (2010)	Minimize the total cost	Z	Analytical method	Pre-distribution and post-distribution operations at a CDT
2	Sathasivan (2011)	Minimize the total weighted completion time	FT	Heuristic, EA, CPLEX	Uncertainty in handling time of trucks at a CDT
3	Konur and Golias (2013a)	Minimize the total truck service time	HT, WT	EA	Uncertainty in truck arrival times
4	Konur and Golias (2013b)	Minimize the average total cost of service and the total cost of service range	HT, WT	HEA	Uncertainty in truck arrival times
5	Walha et al. (2014)	Review of the literature on uncertainties in the CDT operations	N/A	Literature review	A comprehensive review of the literature on uncertainties in the CDT operations
6	Ladier and Alpan (2016)	Minimize the dissatisfaction of the transportation providers for ITs and OTs and the total number of pallets transiting through storage	MSC	Exact optimization, Heuristics	Uncertainties in the CDT truck scheduling
7	Rajabi and Shirazi (2016)	Minimize the total truck waiting time	WT	VNS	Scheduling ITs and OTs at a CDT; uncertainty in truck arrival times
			Miscellaneous		
1	Montana (1998)	Review of scheduling problems	N/A	EA	Review of scheduling problems and application of EAs for solving these problems
2	Bartholdi and Gue (2004)	Determine the best CDT shape	N/A	Analytical method	Evaluation of the CDT shapes
3	Li et al. (2008)	Minimize the total transportation cost	TC	Heuristics	Product distribution between CDTs and warehouses
4	Ross and Jayaraman (2008)	Minimize the total cost	Z	SA-based heuristics	CDT location in supply chains
5	Agustina et al. (2010)	Review of the mathematical models for CDT planning	N/A	Literature review	A comprehensive review of the mathematical models for CDT planning

6	Van Belle et al. (2012)	Review of the literature on the cross-docking concept	N/A	Literature review	A comprehensive review of the literature on the cross-docking concept
7	Shuib and Fatthi (2012)	Review of the quantitative approaches for the CDT door assignment problem	N/A	Literature review	A comprehensive review of the quantitative approaches for the CDT door assignment problem
8	Wang and Li (2012)	Maximize the total profit	TP	Analytical method	A pricing model for perishable products
9	Naskaris et al. (2014)	Evaluation of the cross-docking concept for food supply chains	N/A	Analytical method	Estimated the costs of product distribution via a CDT
10	Ladier and Alpan (2016)	Review of the CDT truck scheduling literature and practice	N/A	Literature review	A comprehensive review of the CDT truck scheduling literature; interviews with practitioners
11	Shiekholeslam and Emamian (2016)	Review of the literature on the cross-docking concept	N/A	Literature review	A comprehensive review of the literature on the cross-docking concept
12	Sreedevi and Saranga (2017)	Understanding the supply chain uncertainties and risks	N/A	Analytical method	Identification of the major supply chain uncertainties and risks
13	Wang et al. (2017)	Minimize the total cost	Z	CH-TLSA, CH-TLTS, CPLEX	Vehicle routing with CDTs
14	Yang et al. (2017)	Maximize the total profit	TP	Heuristics	Perishable food supply chain operations with stochastic demand
15	He et al. (2018)	Quality and operations management in food supply chains	N/A	Literature review	A comprehensive review of the literature on food supply chain operations
16	Margolis et al. (2018)	Minimize the total cost, maximize the overall network connectivity	Z, MSC	ε-constraint	Uncertainties in supply chain operations
17	Schwerdfeger et al. (2018)	Minimize the truck fleet size	MSC	Gurobi, Heuristic	JIT vehicle scheduling from a CDT toward the plant
18	Xin et al. (2018)	Modeling the food quality degradation	N/A	Analytical method	The food quality degradation modeling based on time and temperature

Objective Components
ED – Early Departure;
FC – Fuel consumption;
FL – Flow time;
FT – Finish time;
HT – Handling time;

LD – Late Departure;
MSC – Miscellaneous;
ST – Storage time;
T – Makespan;
TD – Travel distance;
TC – Transportation cost;
TP – Total profit;
WT – Waiting time;
Z – Total cost.

Solution Methods
ACO – Ant Colony Optimization;
ARB – Arbitrary Truck Scheduling;
BS – Beam Search;
DE – Differential Evolution;
DJLPTH – Dynamic Johnson's Rule-based Largest Processing Time Heuristic;
DJRH – Dynamic Johnson's Rule-based Heuristic;
DP – Dynamic Programming;
DSPEA – Delayed Start Parallel Evolutionary Algorithm;
CH-TLSA – a construction heuristic (CH) with a two layer SA;
CH-TLTS – a construction heuristic (CH) with a two layer TS;
EA – Evolutionary Algorithm;
FM – Firefly Metaheuristic;
HDE – Hybrid Differential Evolution;
HEA – Hybrid Evolutionary Algorithm;
HS – Harmony Search;
HSM – Hunting Search Metaheuristic;
ICA – Imperialist Competitive Algorithm;
IHS – Improved Harmony Search;
JLPTH – Johnson's Rule-based Largest Processing Time Heuristic;
JRBH – Johnson's Rule-based Heuristic;
JRH – Johnson's Rule-based Ready Time Heuristic;
KA – Keshtel Algorithm;
LNS – Large Neighborhood Search;

LS – Local Search;
MODE – Multi-Objective Differential Evolution;
MOICA – Multi-Objective Imperialist Competitive Algorithm;
MOPSO – Multi-Objective Particle Swarm Optimization;
MOSA – Multi-Objective Simulated Annealing;
NRGA – Non-Dominated Ranking Genetic Algorithm;
NSGA-II – Non-Dominated Sorting Genetic Algorithm - II;
PAES – Pareto Archived Evolution Strategy;
PEA – Parallel Evolutionary Algorithm;
PSO – Particle Swam Optimization;
SA – Simulated Annealing;
SBSA – Simulation-based Scheduling Algorithm;
SFS – Stochastic Fractal Search;
SPEA-II – Strength Pareto Evolutionary Algorithm;
SPPSO-II – Subpopulation Particle Swarm Optimization-II;
TS – Tabu Search;
VNS – Variable Neighborhood Search.

2.6 Distribution of Reviewed Studies

A distribution of the reviewed studies by year is presented in Figure 12. It can be observed that cross-docking and truck scheduling at CDTs receive an increasing attention of the community (i.e., 29 reviewed studies or 34.1% of the reviewed studies were published after 2015). The latter finding can be explained by tendencies in the national/international trade and supply chains (Dulebenets, 2016a,b). Specifically, supply chain stakeholders are looking for effective approaches to improve management of their operations (including truck scheduling at CDTs).

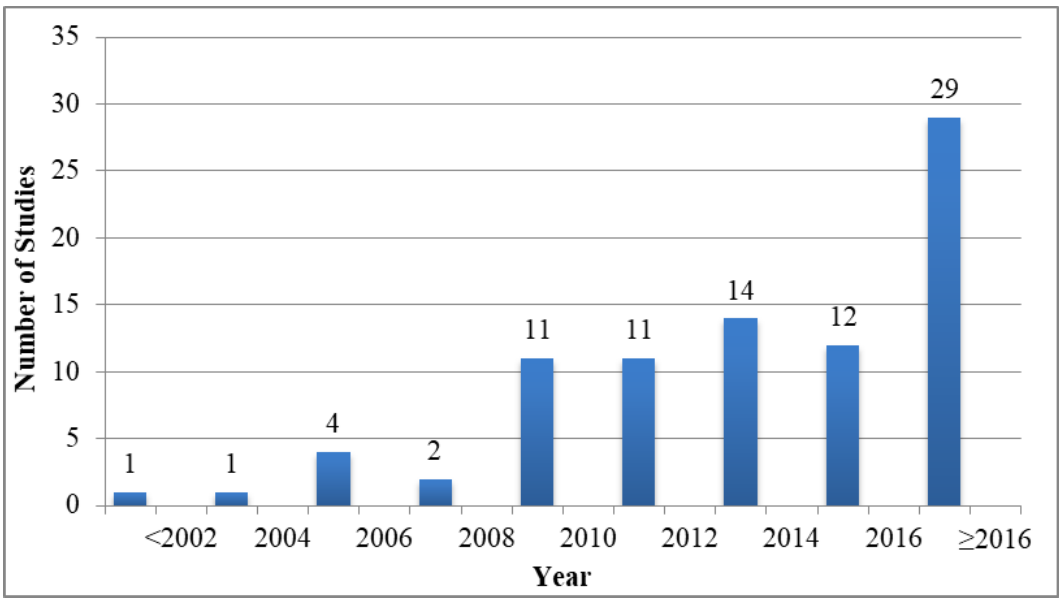


Figure 12: A distribution of the reviewed studies by year.

A distribution of the reviewed studies by objective component considered is presented in Figure 13. It can be noticed that the majority of the reviewed studies primarily focused on minimizing the truck service makespan. Moreover, a significant number of studies captured delayed truck departures, truck waiting time, and handling time of trucks. A distribution of the reviewed studies by solution method is presented in Figure 14. The chart shows that the reviewed studies considered various solution methods to the problem of truck scheduling at a CDT. It can be observed that EA is the solution method that is most utilized in the literature (as more than 36% of the reviewed studies adopted the EA-based solution approaches for the CDT truck scheduling). It is also worthy of note that many of the reviewed studies considered more than one solution method and made comparisons between them based on their performance, as earlier stated in the description of studies.

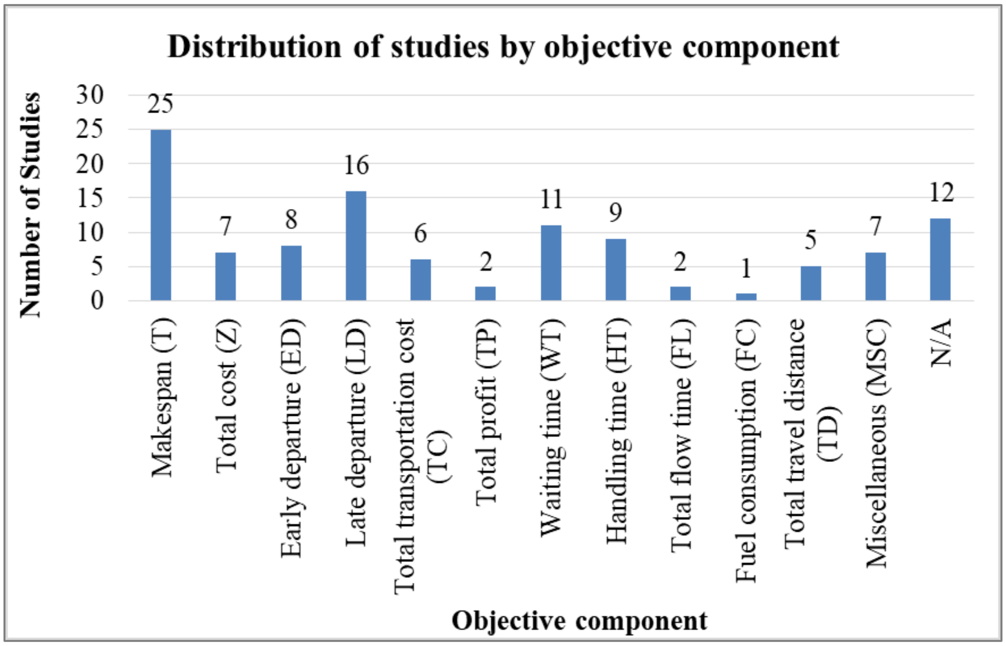


Figure 13: A distribution of the reviewed studies by objective component considered.

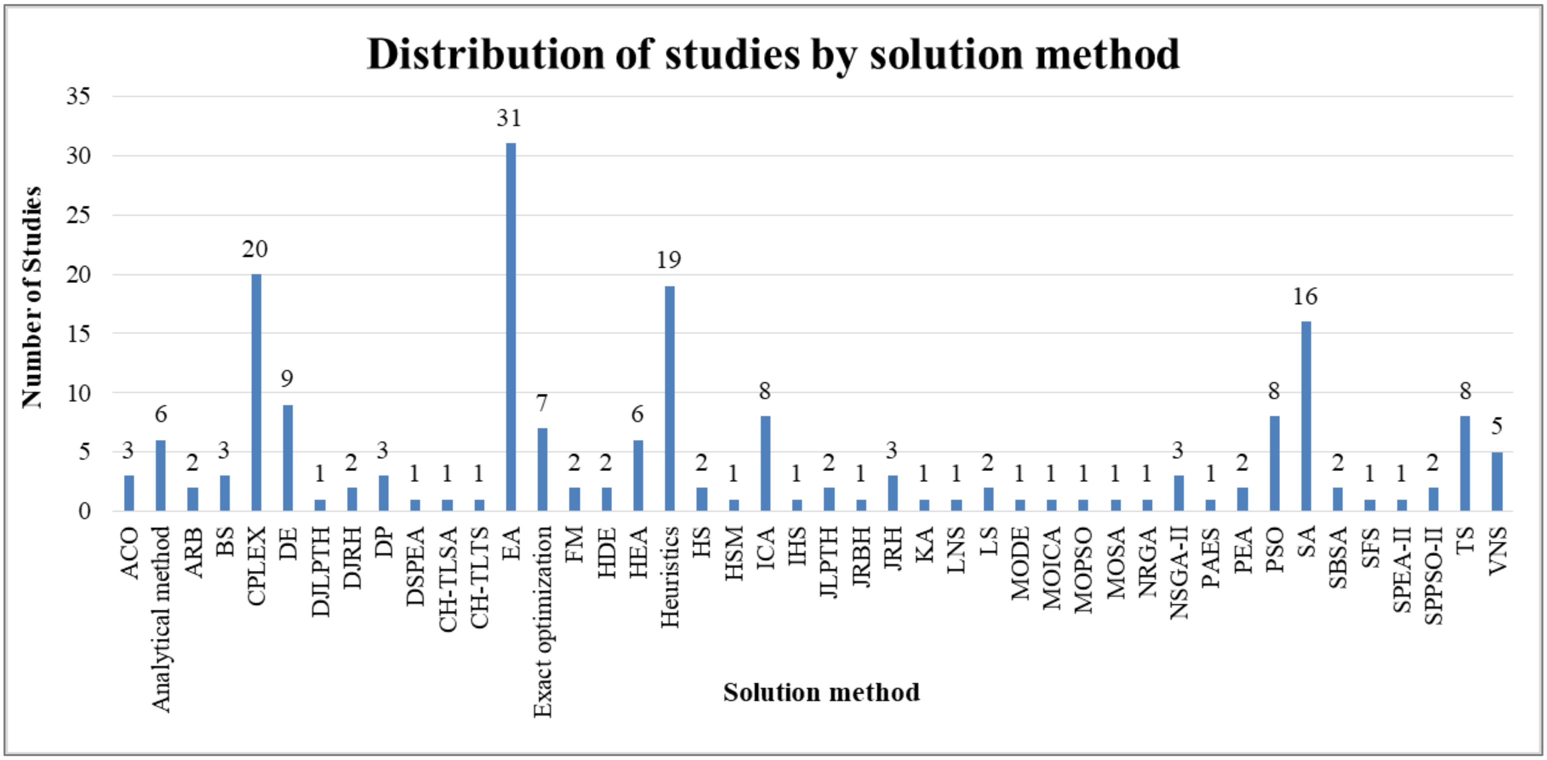


Figure 14: A distribution of the reviewed studies by solution method.

A distribution of the reviewed studies by doors considered is presented in Figure 15. As shown in the chart, most of the studies considered both inbound and outbound doors. A few studies considered only inbound doors (McWilliams et al., 2005; McWilliams, 2009; Liao et al., 2013; Golias et al., 2012; Konur and Golias, 2013a; Konur and Golias, 2013b). A study by Alpan et al. (2011), however, considered only outbound doors.

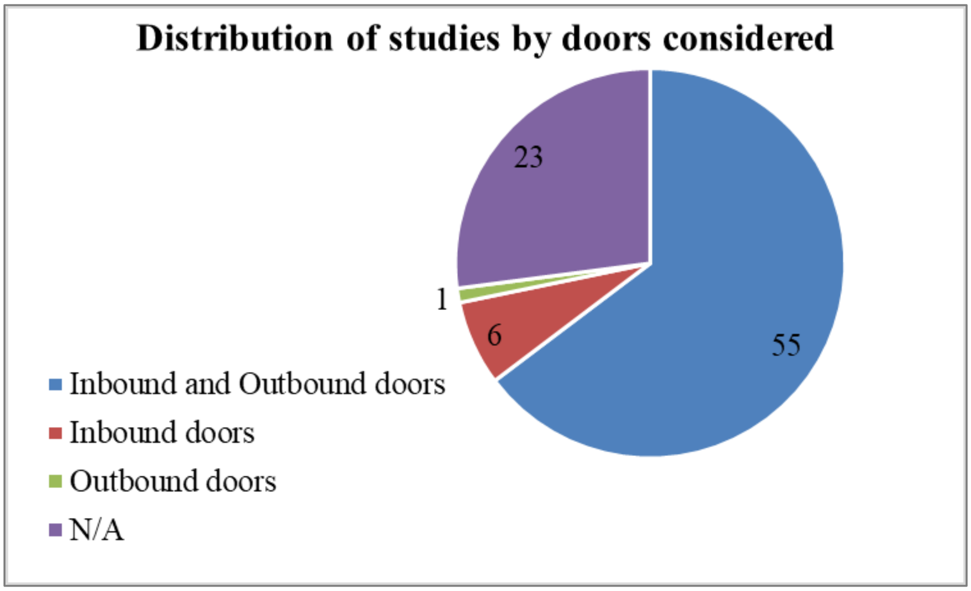


Figure 15: A distribution of the reviewed studies by doors considered.

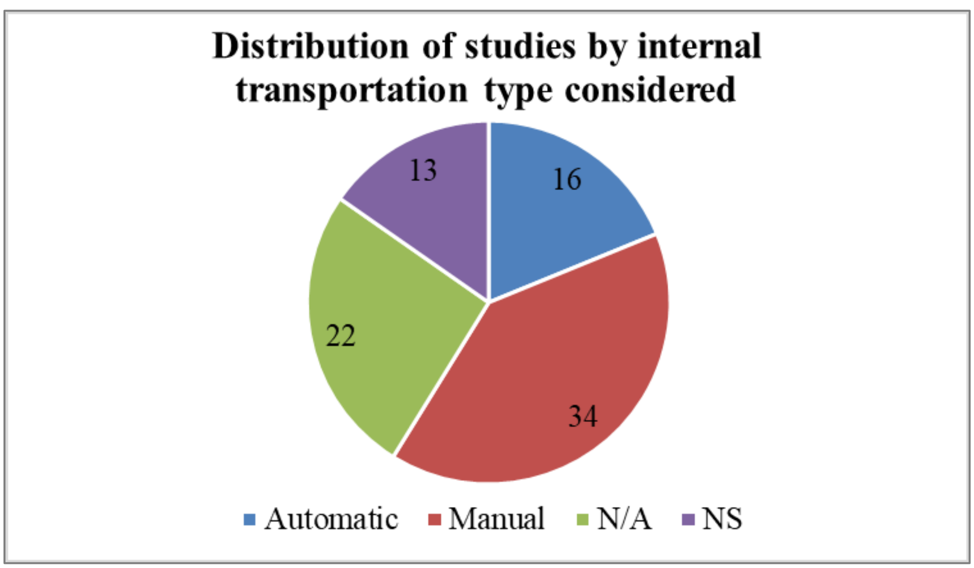


Figure 16: A distribution of the reviewed studies by internal transportation type considered.

A distribution of the reviewed studies by internal transportation type considered is presented in Figure 16. The automatic internal transportation implies that the goods are transported using a conveyor belt, while the manual internal transportation refers to the use of a group of forklifts. Based on a detailed review of the CDT truck scheduling literature, it was found that the majority of studies assumed that the forklift operators were used for unloading the cargo from the arriving inbound trucks and loading the cargo on the outgoing outbound trucks. A distribution of the reviewed studies by door service mode is presented in Figure 17. It can be observed from the chart that a large portion of the reviewed studies considered the exclusive door service (i.e., over 60% of the reviewed studies). In the exclusive door service mode, some of the CDT doors are dedicated to serving inbound trucks, while others are dedicated to serving outbound trucks. Only 7 studies assumed a mixed door service mode, where a truck (either inbound or outbound) can be served at any available door.

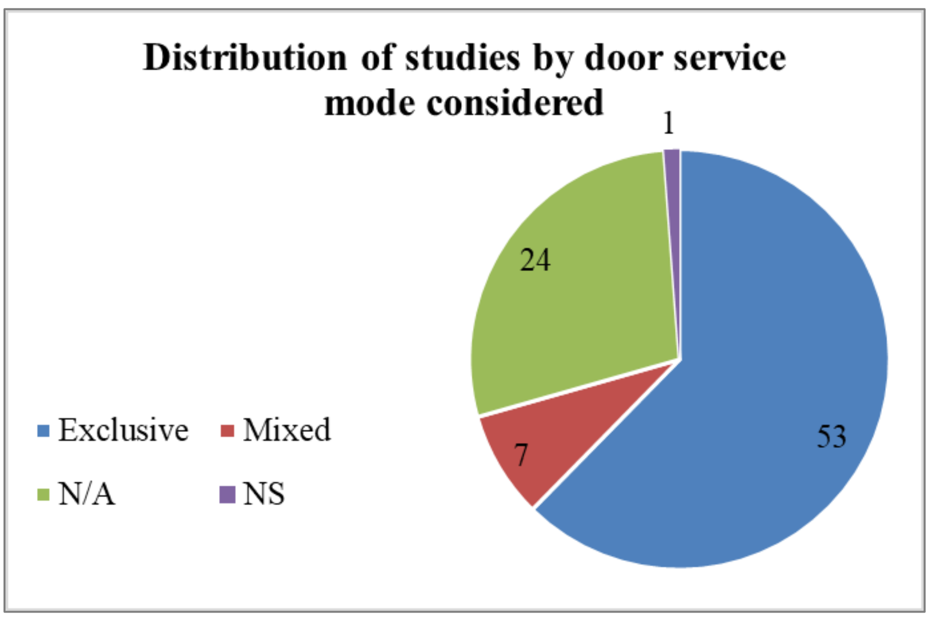


Figure 17: A distribution of the reviewed studies by door service mode considered.

The assumption of a CDT operation that allows for preemption (when a given truck, whether inbound or outbound, is allowed to unload or load some of its products by going in and out of the dock multiple times before service completion, leaving the room for other trucks) has been adopted in a few studies (Yu, 2002; Vahdani et al., 2010; Alpan et al., 2011; Mohtashami 2015; Fazel Zarandi et al., 2016). However, the majority of the reviewed studies do not allow for preemption; therefore, once a truck is docked at a door, it does not leave the door until its service completion. As it was mentioned earlier, the shape of a cross-dock significantly affects its operations (as it directly influences the total travel distance by the internal handling equipment). Several studies proposed the CDT truck scheduling models, considering the CDTs with I-shapes (see, for example, Ley et al., 2007; Golias et al., 2012; Joo and Kim, 2013; Dulebenets, 2018a; Dulebenets, 2018b; Dulebenets, 2019a). A substantial number of the reviewed studies proposed generic mathematical models and solution approaches for the CDT truck scheduling problem, which can be applied to the CDTs of different shapes.

Some of the reviewed studies modeled the CDT operations with limited temporary storage and/or limited resource capacity (see, for example, Lim et al., 2005; Vis and Roodbergen, 2008; Miao et al., 2009; Choy et al., 2012; Gelareh et al. 2013; Guignard et al., 2014; Serrano et al., 2017; Wang et al., 2018; Chmielewski et al., 2009; Rajabi and Shirazi, 2016; Li et al., 2008). However, the majority of the CDT truck scheduling studies, which were identified as a result of the conducted literature review, assumes infinite temporary storage and/or resource capacity.

2.7 The Existing Gaps in the State-of-the-Art

The comprehensive review of studies in this section shows an increased interest of researchers in the CDT truck scheduling problem. Many of the reviewed studies proposed mathematical models that focused on general truck scheduling problems, excluding the modeling of the decay of products that are perishable in nature throughout various handling operations within a given CDT. Other studies have been found to accurately model the decay of products that are perishable in nature during supply chain operations using an exponential function (Wang and Li, 2012; Dulebenets and Ozguven, 2017), although not within the CDT truck scheduling. As a gap in the existing literature, there is no study that addresses the modeling of the decay of products that are perishable in nature throughout various handling operations within a given CDT.

Considering substantial monetary losses due to mismanagement of supply chains with perishable products every year, this dissertation proposes a novel mixed integer mathematical model for CDT truck scheduling. The proposed model incorporates the decay of products that are perishable in nature as a component of the objective function. Based on existing studies which capture the decay of products that are perishable in nature, this study uses an exponential function to accurately compute the decay of perishable products throughout the handling process (i.e., unloading of perishable products from the inbound trucks and transfer of these products to the designated temporary locations of storage areas; transfer of perishable products from the designated temporary locations of storage areas and loading of these products on the outbound trucks). Furthermore, to enhance the computational speed, a novel customized EA-based metaheuristic algorithm is developed to address the problem and provide good-quality solutions in a timely manner. The proposed mixed integer mathematical model, the EA metaheuristic developed, and the managerial insights to be drawn from the sensitivity analyses would assist CDT managers in making decisions regarding perishable product distribution and facilitate supply chain operations.

CHAPTER THREE

3 PROBLEM DESCRIPTION

This section describes in detail the operations of the CDT that will be modeled in this study. Based on a detailed review of the CDT truck scheduling literature, it was found that the CDT truck scheduling models are typically differentiated based on the operational assumptions considered (Ladier and Alpan, 2016). These operational assumptions include, but are not limited to, the following: door service mode; existence of temporary storage; the pattern of truck docking (i.e., whether preemption is allowed or not); the arrival pattern of trucks (i.e., whether uncertainty in truck arrival times is considered or not); the pattern of truck service (i.e., whether uncertainty in truck service times is considered or not), and others. As reported in several studies on the CDT truck scheduling, the most commonly considered CDT shape is the I-shape (Ladier and Alpan, 2016; Dulebenets, 2019a). Therefore, an I-shape CDT will be modeled in this study (see Figure 18). However, without loss of generality, the developed mathematical model can be applied to other shapes as well (such as L, U, T, H, X, E, etc.).

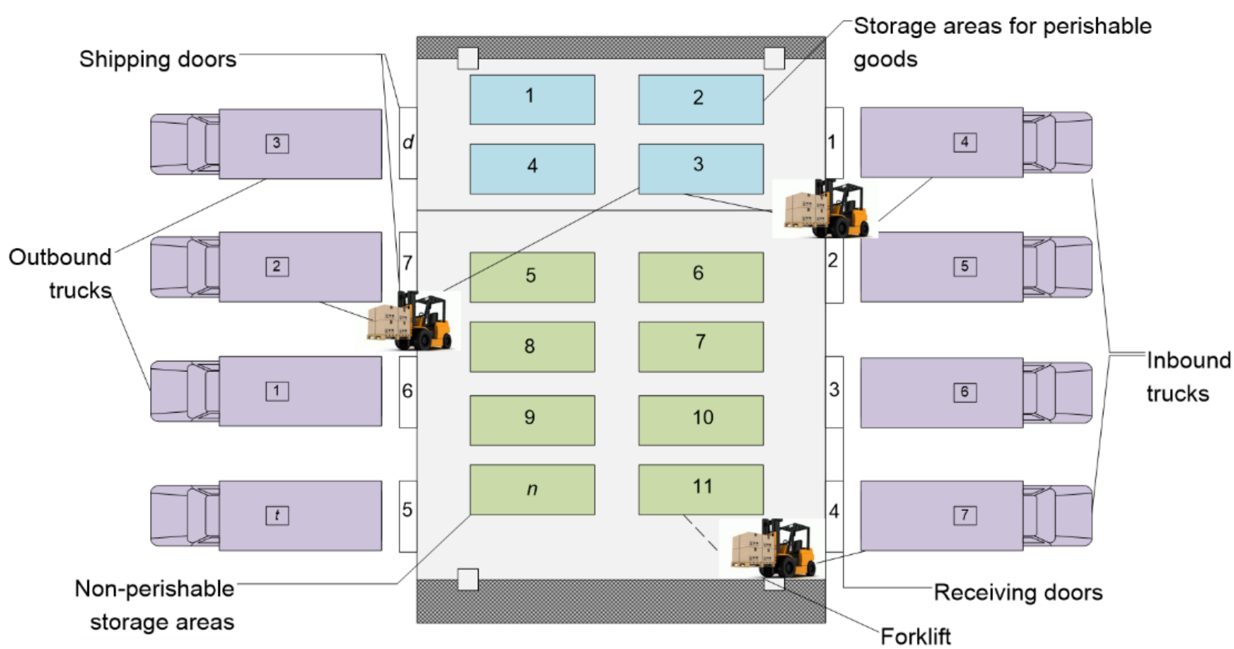


Figure 18: Cross-docking terminal.

The following subsections elaborate on truck scheduling at CDT: (1) arrival of trucks; (2) assignment of doors at the CDT; (3) internal operations at the CDT; (4) decay modeling of perishable products; and (5) CDT manager's key objective. Moreover, an example of the truck service for a pair of trucks is provided at the end of this section as well.

3.1 Arrival of Trucks

The operations of the CDT begin when the trucks both in- and outbound arrive at the terminal (see Figure 18). Let $T = \{1, \dots, w^1\}$ be the set of arriving trucks (i.e., both inbound and outbound) at the CDT. Let $T^{in} = \{1, \dots, w^2\}, T^{in} \subseteq T$ and $T^{out} = \{1, \dots, w^3\}, T^{out} \subseteq T$ be the set of the arriving inbound trucks as well as the set of the arriving outbound trucks, respectively. The CDT manager performs a truck-to-door assignment for an unloading or loading process to start. It is assumed that the loading procedure will be performed for the outbound trucks, while the unloading procedure will be conducted for the inbound trucks. The inbound trucks may arrive for service at the considered CDT either fully loaded or partially-loaded. Also, the outbound trucks may arrive for service at the considered CDT either fully unloaded or partially-unloaded. This study assumes that a given truck can be either inbound or outbound: $T^{in} \cup T^{out} = T$ and $T^{in} \cap T^{out} = \emptyset$. The arrival times of trucks are assumed to be deterministic in nature, i.e. all the arriving trucks follow an arrival schedule that is pre-determined, which was previously coordinated between the CDT manager and corresponding logistics companies. Let $\tau_t^{at}, t \in T$ (hours) be the time of arrival of truck t at the CDT. The assumption regarding deterministic truck arrival times is common in the CDT truck scheduling literature (Miao et al., 2009; Vahdani, et al.; 2010; Boloori Arabani et al., 2012; Liao et al., 2013; Ladier and Alpan, 2016). However, the scope of the future research may focus on modeling the CDT operations, when the truck arrival times are not known with certainty.

If the assigned door is available upon arrival of a given truck, the truck is served; if not, the truck waits for the door to be available in the designated parking area. The CDT manager incurs an additional truck waiting cost. Denote $c_t^{wt}, t \in T$ (USD/hour) as the unit waiting cost of truck t at the CDT. This cost is incurred when a truck is assigned for service at either the receiving door or the shipping door, but the door is not available immediately upon the truck arrival. On the other hand, if a truck is served as soon as it arrives at the CDT, no waiting cost is incurred by the CDT manager. The supply chain operations and the product deliveries may be negatively affected in case of excessive waiting times. The latter justifies consideration of the truck waiting cost.

3.2 Assignment of Doors at the CDT

Based on the CDT operations literature, there exist two types of the CDT door service modes, including the following (Ladier and Alpan, 2016): (1) mixed service mode – the same doors can be used as either receiving or shipping doors, and, therefore, can serve either inbound or outbound trucks (one door serves one truck at a time); and (2) segregated service mode – there is a set of receiving doors that are specifically dedicated to serve the arriving inbound trucks, and there is a set of shipping doors that are specifically designated to serve the arriving outbound trucks. Peak loads are usually handled at a CDT in the mixed door service mode. However, the CDTs that operate in the segregated service mode can be converted to the mixed service mode. This study assumes that the considered CDT operates the segregated mode (see Figure 18).

Let $D = \{1, \dots, w^4\}$ be the set of the doors that are available for the service of arriving trucks. Let $D^{in} = \{1, \dots, w^5\}, D^{in} \subseteq D$ and $D^{out} = \{1, \dots, w^6\}, D^{out} \subseteq D$ be the set of the available receiving doors and also the available shipping doors, respectively. An arriving inbound truck should be scheduled for service at one of the available receiving doors in one of the service orders. For example, if inbound trucks "1", "2", and "3" are assigned for service at the inbound door "2", one of these trucks will be served first, one of these trucks will be served second, and one of these trucks will be served third. In the meantime, an arriving outbound truck should be scheduled for service at an available shipping door in any of the service orders. Denote $O = \{1, \dots, w^7\}$ as the set of service orders available.

3.3 Internal Operations at the CDT

Once a given truck is docked at the assigned CDT door, the internal handling equipment is assigned for either unloading of the products (in case of the inbound trucks) or loading of the products (in case of the outbound trucks). Unloading starts after a given inbound truck is docked. The loading of an outbound truck starts only after the corresponding inbound truck begins service or directly from the dedicated temporary location of storage (in cases when the products assigned have already been moved to the dedicated temporary location of storage). Various internal handling equipment types can be deployed for the service of the arriving trucks, including forklift operators, conveyor belts, combination of forklift operators and conveyor belts, etc. This study assumes that the CDT manager will be using a group of forklift operators for the service of the arriving trucks. Based on the expected number of the arriving trucks, the CDT manager reserves forklift operators that are sufficient for the considered planning horizon. Once a given inbound truck is docked at the designated receiving door, it remains docked at that door until the completion of its service. Likewise, each outbound truck will remain docked at designated shipping door until the completion of its service. Therefore, preemption is not considered throughout service of the arriving trucks.

The products, which are delivered to the CDT by the inbound trucks, are usually transported in standard packaging units, such as pallets, insulated boxes, standard boxes, etc. A given inbound truck can carry a variety of product types for one outbound truck or multiple outbound trucks. The handling time refers to the time, required to unload or load a given arriving truck (i.e., a given inbound or outbound truck, respectively), denoted as $\tau_t^{ht}, t \in T$ (hours). The unloaded goods are sorted in the temporary locations of storage areas, which are situated between the receiving area and the shipping area (see Figure 18). Let $N = \{1, \dots, w^8\}$ be the set of the temporary locations of storage areas.

Along with storage, each temporary location of storage also serves the purpose of unpacking, sorting, and packing of the products based on the customer requests. Some of the incoming products are loaded directly to the assigned outbound truck, if the corresponding outbound truck is available for service at the assigned shipping door. Otherwise, the products are stored for a while at the designated temporary location of storage. The allocation of products among the available locations of storage areas is assumed to be dependent on the product type. Denote $P = \{1, \dots, w^9\}$ as the set of products that have been transported by the inbound trucks. The CDT considered in this study has temporary locations of storage areas which are classified into two categories (see Figure 18): (1) temporary locations of storage areas that are reserved for the arriving products that are non-perishable; and (2) temporary locations of storage areas that are reserved for the arriving products that are perishable.

The temporary locations of storage areas, which are designated for perishable products, are assumed to maintain a particular temperature to make sure that the delivered perishable products remain fresh, while they are being stored until arrival of the corresponding outbound trucks. It is noteworthy that perishable goods have a short lifespan and, therefore, are high-priority goods. Also, only perishable products are stored in the locations of storage areas designated for them. The other product types (i.e., non-perishable product types) will be stored in different temporary locations of storage areas within the CDT considered (see Figure 18). Based on the expected number of the trucks arriving at the CDT and the quantity of incoming products, the CDT manager allocates an adequate portion for the available temporary locations of storage areas for the considered planning horizon. Such assumption is in line with many studies on the CDT truck scheduling, which were conducted to date (Vis and Roodbergen, 2008; Ladier and Alpan, 2016).

Denote $\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{ts}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ (hours) as the length of storage time for a given product of type p that has been transported to the CDT by inbound truck $\underline{t}$ for outbound truck $\bar{t}$. The CDT manager incurs a cost, which is associated with storing a particular product type in the designated location of storage. Let $c_p^{ts}, p \in P$ (USD/hour) be the unit cost of storage for product type p. The unit cost of storage is assumed to vary based on the type of product, as products which are perishable in nature will require the temperature-controlled locations of storage areas, which are more costly to operate as compared to the locations of storage areas for non-perishable products that need no particular temperature.

The time of service of a given truck t ($\boldsymbol{\tau}_{tdo}^{tst}, t \in T, d \in D, o \in O$ – hours) can be decomposed into the main two parts, such as: (A) time of truck handling (τ_t^{ht} $t \in T$ – hours) that refers to the time that is needed to unload/load the products; and (B) product transfer time from the CDT door assigned to the reserved temporary location of storage ($\tau_{dn}^{tr}, d \in D, n \in N$ – hours). The service time for a given inbound truck includes the unloading time and the time, required to transfer the unloaded products from the assigned receiving door to the designated location of storage. On the other hand, the service time for a given outbound truck includes the time, required to transfer the products from the designated location of storage to the assigned shipping door, and the loading time. Let $z_{tn} = 1, t \in T, n \in N$ if the products that are carried by truck t will be further assigned to the temporary location of storage n (=0 otherwise). Note that the service time of trucks is further influenced by a number of factors such as: (1) the number of product types; (2) the amount of each product type; (3) the size and weight of the packaging units of the product; and (4) the distance between the assigned CDT door and the reserved location of storage.

The service time of the truck, carrying multiple product types, is expected to be larger as compared to the truck, carrying a single product type (i.e., additional time may be required in order to handle the product types within a truckload in an appropriate manner). Furthermore, the service time of a truck with a large quantity of each product type is expected to be larger as compared to the truck with a small quantity of each product type. The transfer of the overweight and/or oversize product packaging units by the forklift operators within the CDT is expected to be more time consuming as compared to the transfer of the standard product packaging units (i.e., which have standard weight and size). If the assigned door (whether receiving or shipping) is closer to the allocated temporary location of storage, the service time will be shorter as compared to the case, when the assigned door is located further away from the allocated temporary location of storage.

This study assumes that the truck service times are deterministic in nature, i.e. service of the arriving trucks will be completed based on a pre-determined schedule, as it was planned by the CDT manager. The assumption regarding deterministic truck service times is common in the CDT truck scheduling literature (Miao et al., 2009; Vahdani et al., 2010; Boloori Arabani et al., 2012; Liao et al., 2013; Ladier and Alpan, 2016). However, the scope of the future research may focus on modeling the CDT operations, when the truck service times are not known with certainty. Throughout the truck service, the CDT manager incurs the unit truck service cost ($c_t^{tst}, t \in T$ – USD/hour). It is assumed that the CDT manager is required to complete service of a given truck t before a particular deadline ($\tau_t^{sd}, t \in T$ – hours). When the service of a given truck t is completed after the established deadline, the CDT manager will incur a unit cost of delay in truck departure ($c_t^{dt}, t \in T$ – USD/hour). A large number of previous studies discussed the timely truck service completion, its importance for proper operations in supply chains, and imposed penalties for violating of scheduled truck departure times (Boloori Arabani et al., 2010; Ladier and Alpan, 2016; Dulebenets, 2018a,b; Dulebenets, 2019a).

3.4 Decay Modeling of Perishable Products

The moment perishable products arrive at the CDT on any given inbound truck, the products will be moved to the temporary locations of storage areas assigned, which are particularly used for storage of perishable products. Throughout the unloading period of perishable products from the arriving inbound trucks, transfer of the products within the CDT, and loading of perishable products on the assigned outbound trucks, the products are assumed to deteriorate (since they are not stored under a particular temperature). The following relationship will be adopted in this study to estimate the product quality of the given product type p that has been transported by arriving inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time τ (Dulebenets and Ozguven, 2017; Li et al., 2016):

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

where:

$\boldsymbol{Q}^{\tau}_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ – quality of a given product type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time τ (%);

$Q^{0}_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ – quality of a given product type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time 0 (%);

$\lambda_p, p \in P$ – decay rate of product type p (hours^{-1});

$\boldsymbol{\tau}^{tht}_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ – total handling and transfer time for the product of type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (hours).

For a given product of type p, the total handling and transfer time of the product that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$, can be calculated based on the unloading time of inbound truck $\underline{t}$, transfer time from inbound truck $\underline{t}$ to the reserved temporary location of storage, transfer time from the assigned location of storage to outbound truck $\bar{t}$, and the loading time of outbound truck $\bar{t}$ using the following relationship:

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

where:

$\boldsymbol{x}_{tdo} = 1, t \in T, d \in D, o \in O$ if truck t is scheduled for service at CDT door d in the o^{th} service order (=0 otherwise);
$\varphi_{\underline{t}\bar{t}p} = 1, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ if inbound truck $\underline{t}$ carries the product of type p for outbound truck $\bar{t}$ (=0 otherwise).

The decrease in the quality of a given product type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ throughout handling and transfer time at the considered CDT can be computed using the following relationship:

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

where:
$\boldsymbol{\Delta Q}_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ – change in quality of the product of type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (%).

In order to keep the end customers satisfied with quality of the delivered perishable products, the logistics companies and the CDT manager strive to keep these products as fresh as possible. It is assumed that the CDT manager incurs an additional cost, associated with the percentage decay of a given perishable product throughout the handling and transfer of the perishable products. Denote $c_p^{dc}, p \in P$ (USD/% decay) as the unit decay cost of a given product of type p. Note that since the temporary locations of storage areas reserved for the products which are perishable maintain a particular temperature, the decay of these products throughout the storage is assumed to be insignificant and will be ignored.

3.5 CDT Manager's Key Objective

In this study, the CDT manager is saddled with the responsibility of certain decision making with respect to major CDT operations. The arriving inbound trucks have to be assigned to the available receiving doors for service, while the outbound trucks should be assigned to the available shipping doors for service. Moreover, certain product types, handled at the considered CDT, are perishable in their nature. The quality of perishable products deteriorates throughout handling of these products and transfer to the reserved areas of temporary storage. The assignment of the arriving trucks for service at the available CDT doors directly affects the product transfer time between the CDT doors and the reserved areas of temporary storage, which further influences the product decay throughout the transportation process. The overall CDT manager’s key objective is to develop an effective truck schedule that is expected to have the least total cost to be further incurred by the CDT manager throughout service of the arriving trucks, which is composed of the following major components: (i) total cost of waiting; (ii) total cost of service; (iii) total cost of product storage; (iv) total cost of delay on the departure of trucks; and (v) total cost of perishable product decay.

3.6 An Illustration of the Service of Arriving Trucks

This section presents a detailed illustrative example of serving a pair of trucks both in- and outbound at the CDT considered (see Figure 19). In particular, this example shows service of the inbound truck “4”, bringing in perishable products that would be loaded on the outbound truck “1”. The CDT manager allocated the location of storage “4” for temporary storage of the products. The location of storage “4” is specifically designated for storage of perishable products and is able to maintain a particular temperature to prevent deterioration of perishable products. After conducting a preliminary scheduling, the inbound truck “4” is assigned for service at the receiving door “1”, while the outbound truck “1” is assigned for service at the shipping door “6”.

Note that the receiving door “1” and the shipping door “6” are located in a close vicinity of the location of storage “4”, which further allows reducing the total travel distance by the forklift operators. In case if the CDT manager decides to re-assign the outbound truck “1” for service at another door, which is located further away from the location of storage “4” (e.g., the shipping door “5”), the total travel distance by the forklift operators will increase. The latter will also cause an increase in the service time of the outbound truck “1”. Likewise, in case if the CDT manager decides to re-assign the inbound truck “4” for service at another door, which is located further away from the location of storage “4” (e.g., the receiving door “3”), the total travel distance by the forklift operators will increase. The latter will also cause an increase in the service time of the inbound truck “4”.

The service completion time for the outbound truck “1” can be computed using the following components: (1) the arrival time of the inbound truck “4”; (2) the waiting time of the inbound truck “4” (in case if the receiving door “1” is not available immediately upon arrival of the inbound truck “4”); (3) the service time of the inbound truck “4” (i.e., the product unloading time and the transfer time of the delivered products from the receiving door "1" to the location of storage "4"); (4) decomposition, sorting, consolidation, and storage of the products delivered for the outbound truck "1"; (5) the arrival time of the outbound truck "1"; (6) the waiting time of the outbound truck "1" (in case if the shipping door "6" is not available immediately upon arrival of the outbound truck "1"); and (7) the service time of the outbound truck "1" (i.e., the transfer time of the delivered products from the location of storage "4" to the shipping door "6" and the product loading time).

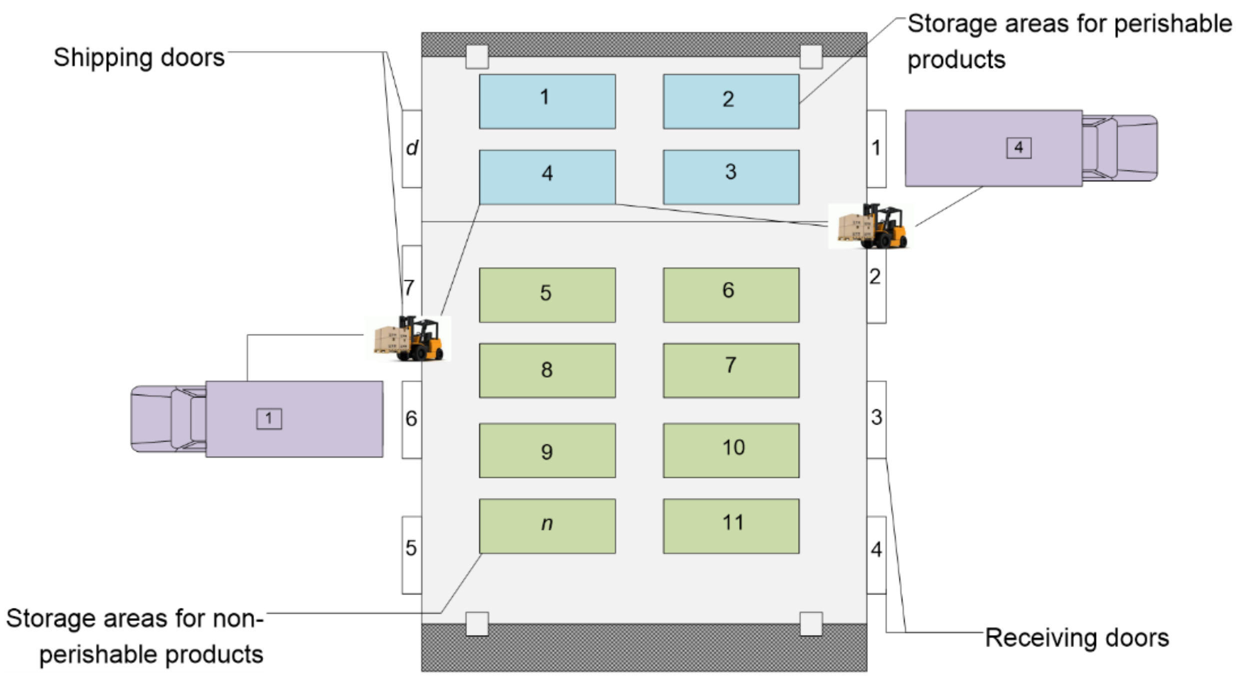


Figure 19: An example of service of the arriving trucks.

The truck service example, provided in this section, describes the operations that are associated with the inbound truck "4" and the outbound truck "1" only. Without loss of generality, the inbound truck "4" may be delivering the products for other outbound trucks to be served at the considered CDT as well. Furthermore, depending on the requests from the end customers, the outbound truck "1" may carry the products that have been transported by multiple inbound trucks (not just the inbound truck "4").

CHAPTER FOUR

4 MODEL FORMULATION

4.1 Non-Linear Mathematical Formulation

The formulated mixed integer mathematical model for the truck scheduling problem at the CDT with perishable products (**TSP-CDT**) is presented in this section. Table 3 presents the description of the main components of the **TSP-CDT** mathematical model in detail.

Table 3: The main components of the **TSP-CDT** mathematical model.

Model Component Type	Notation	Description
Sets	$T = \{1, \dots, w^1\}$	set of the arriving trucks
	$T^{in} = \{1, \dots, w^2\}, T^{in} \subseteq T$	set of the arriving inbound trucks
	$T^{out} = \{1, \dots, w^3\}, T^{out} \subseteq T$	set of the arriving outbound trucks
	$D = \{1, \dots, w^4\}$	set of the available CDT doors
	$D^{in} = \{1, \dots, w^5\}, D^{in} \subseteq D$	set of the CDT receiving doors
	$D^{out} = \{1, \dots, w^6\}, D^{out} \subseteq D$	set of the CDT shipping doors
	$O = \{1, \dots, w^7\}$	set of all the available service orders
	$N = \{1, \dots, w^8\}$	set of the available temporary storage locations
	$P = \{1, \dots, w^9\}$	set of the product types to be handled at the CDT
Decision variables	$\boldsymbol{x}_{tdo} \in \mathbb{B}\ \forall t \in T, d \in D, o \in O$	= 1 if truck t is assigned to be served at the CDT door d in the o^{th} service order (=0 otherwise)
Auxiliary variables	$\boldsymbol{TC} \in \mathbb{R}^+$	total cost to be incurred by the CDT manager throughout service of the arriving trucks (USD)
	$\boldsymbol{y}_{tdo} \in \mathbb{R}^+\ \forall t \in T, d \in D, o \in O$	time of idleness of the CDT door d between service of truck t and its preceding truck that was served in the $(o-1)^{th}$ service order (hours)
	$\boldsymbol{\tau}_t^{wt} \in \mathbb{R}^+\ \forall t \in T$	time of waiting for truck t (hours)
	$\boldsymbol{\tau}_t^{st} \in \mathbb{R}^+\ \forall t \in T$	start of service time for truck t (hours)
	$\boldsymbol{\tau}_{tdo}^{tst} \in \mathbb{R}^+\ \forall t \in T, d \in D, o \in O$	total time of service for truck t at the CDT door d served in the o^{th} service order (hours)

	$\boldsymbol{\tau}_t^{ft} \in \mathbb{R}^+ \; \forall t \in T$	end of service time for truck t (hours)
	$\boldsymbol{\tau}_t^{dt} \in \mathbb{R}^+ \; \forall t \in T$	delay on departure for truck t (hours)
	$\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{ts} \in \mathbb{R}^+ \; \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	temporary time of storage for the product type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (hours)
	$\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht} \in \mathbb{R}^+ \; \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	total handling and transfer time for the product of type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (hours)
	$\boldsymbol{Q}_{\underline{t}\bar{t}p}^{\tau} \in \mathbb{R}^+ \; \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	quality of the product of type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time τ (%)
	$\boldsymbol{\Delta Q}_{\underline{t}\bar{t}p} \in \mathbb{R}^+ \; \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	quality change for the product type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (%)
Parameters	$\tau_t^{at} \in \mathbb{R}^+ \; \forall t \in T$	arrival time of truck t at the CDT (hours)
	$\tau_t^{ht} \in \mathbb{R}^+ \; \forall t \in T$	time of handling of truck t (hours)
	$\tau_t^{sd} \in \mathbb{R}^+ \; \forall t \in T$	scheduled departure time for truck t from the CDT (hours)
	$\tau_{dn}^{tr} \in \mathbb{R}^+ \; \forall d \in D, n \in N$	transfer time from the CDT door d to location of storage 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 location of storage 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 product of type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (product units)
	$Q_{\underline{t}\bar{t}p}^{0} \in \mathbb{R}^+ \; \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	quality of product type p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ at time 0 (%)
	$\lambda_p \in \mathbb{R}^+ \; \forall p \in P$	rate of decay of product type p (hour^{-1})
	$c_t^{wt} \in \mathbb{R}^+ \; \forall t \in T$	unit cost of waiting for truck t (USD/hour)

$c_t^{tst} \in \mathbb{R}^+ \; \forall t \in T$	unit cost of service for truck t (USD/hour)
$c_p^{ts} \in \mathbb{R}^+ \; \forall p \in P$	unit cost of product storage for product type p (USD/hour)
$c_t^{dt} \in \mathbb{R}^+ \; \forall t \in T$	unit cost of delay in truck departure for truck t (USD/hour)
$c_p^{dc} \in \mathbb{R}^+ \; \forall p \in P$	unit decay cost for the product of type p (USD/% decay)
M	sufficiently large positive number

TSP-CDT: Problem of Truck Scheduling at the CDT with Perishable Products

$$min\, \boldsymbol{TC} = [\left(\sum_{t\in T} \boldsymbol{\tau}_t^{wt}\, c_t^{wt}\right) + \left(\sum_{t\in T}\sum_{d\in D}\sum_{o\in O} \boldsymbol{\tau}_{tdo}^{tst} c_t^{tst}\right) + \left(\sum_{\underline{t}\in T^{in}}\sum_{\bar{t}\in T^{out}}\sum_{p\in P} \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{ts} q_{\underline{t}\bar{t}p} c_p^{ts}\right) + \left(\sum_{t\in T} \boldsymbol{\tau}_t^{dt}\, c_t^{dt}\right) + \left(\sum_{\underline{t}\in T^{in}}\sum_{\bar{t}\in T^{out}}\sum_{p\in P} \boldsymbol{\Delta Q}_{\underline{t}\bar{t}p} q_{\underline{t}\bar{t}p} c_p^{dc}\right)] \tag{4}$$

Subject to:

$$\sum_{d\in D}\sum_{o\in O} \boldsymbol{x}_{tdo} = 1 \; \forall t \in T \tag{5}$$

$$\sum_{t\in T} \boldsymbol{x}_{tdo} \le 1 \; \forall d \in D, o \in O \tag{6}$$

$$\sum_{d\in D^{out}}\sum_{o\in O} \boldsymbol{x}_{tdo} = 0 \; \forall t \in T^{in} \tag{7}$$

$$\sum_{d\in D^{in}}\sum_{o\in O} \boldsymbol{x}_{tdo} = 0 \; \forall t \in T^{out} \tag{8}$$

$$\sum_{t^*\in T: t^*\neq t}\;\sum_{o^*\in O: o^*<o} (\boldsymbol{\tau}_{t^*do^*}^{tst} + \boldsymbol{y}_{t^*do^*}) + \boldsymbol{y}_{tdo} \ge \tau_t^{at} \boldsymbol{x}_{tdo} \; \forall t \in T, d \in D, o \in O \tag{9}$$

$$\boldsymbol{\tau}_t^{st} \ge \sum_{t^*\in T: t^*\neq t}\;\sum_{o^*\in O: o^*<o} \left(\boldsymbol{\tau}_{t^*do^*}^{tst} + \boldsymbol{y}_{t^*do^*}\right) + \boldsymbol{y}_{tdo} - M(1 - \boldsymbol{x}_{tdo}) \; \forall t \in T, d \in D, o \in O \tag{10}$$

$$\boldsymbol{\tau}_{\bar{t}}^{st} \ge \boldsymbol{\tau}_{\underline{t}}^{st} \varphi_{\underline{t}\bar{t}p} \; \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P \tag{11}$$

$$\boldsymbol{\tau}_{tdo}^{tst} = [\tau_t^{ht} + \sum_{n\in N} (\tau_{dn}^{tr} z_{tn})] \boldsymbol{x}_{tdo} \; \forall t \in T, d \in D, o \in O \tag{12}$$

$$\boldsymbol{\tau}_t^{ft} = \boldsymbol{\tau}_t^{st} + \sum_{d\in D}\sum_{o\in O} \boldsymbol{\tau}_{tdo}^{tst} \; \forall t \in T \tag{13}$$

$$\boldsymbol{\tau}_t^{wt} \ge \boldsymbol{\tau}_t^{st} - \tau_t^{at} \; \forall t \in T \tag{14}$$

$$\boldsymbol{\tau}_t^{dt} \ge \boldsymbol{\tau}_t^{ft} - \tau_t^{sd} \; \forall t \in T \tag{15}$$

$$\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht} \ge \left(\tau_{\underline{t}}^{ht} + \sum_{d\in D}\sum_{o\in O}\sum_{n\in N} \tau_{dn}^{tr} z_{\underline{t}n} \boldsymbol{x}_{\underline{t}do} + \tau_{\bar{t}}^{ht} + \sum_{d\in D}\sum_{o\in O}\sum_{n\in N} \tau_{dn}^{tr} z_{\bar{t}n} \boldsymbol{x}_{\bar{t}do}\right) \varphi_{\underline{t}\bar{t}p} \tag{16}$$

$$\forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$$

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

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

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

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

$$\begin{gathered}\boldsymbol{TC}, \boldsymbol{y}_{tdo}, \boldsymbol{\tau}^{wt}_{t}, \boldsymbol{\tau}^{st}_{t}, \boldsymbol{\tau}^{tst}_{tdo}, \boldsymbol{\tau}^{ft}_{t}, \boldsymbol{\tau}^{dt}_{t}, \boldsymbol{\tau}^{ts}_{\underline{t}\bar{t}p}, \boldsymbol{\tau}^{tht}_{\underline{t}\bar{t}p}, \boldsymbol{Q}^{\tau}_{\underline{t}\bar{t}p}, \boldsymbol{\Delta Q}_{\underline{t}\bar{t}p}, \tau^{at}_{t}, \tau^{ht}_{t}, \tau^{sd}_{t}, \tau^{tr}_{dn}, Q^{0}_{\underline{t}\bar{t}p}, \lambda_p, c^{wt}_{t}, \\ c^{tst}_{t}, c^{ts}_{p}, c^{dt}_{t}, c^{dc}_{p}, M \in \mathbb{R}^{+} \ \forall t \in T, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, d \in D, o \in O, p \in P, n \in N\end{gathered} \quad (21)$$

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

The objective function (4) of the proposed mixed integer **TSP-CDT** mathematical model is the minimization of the total cost ($\boldsymbol{TC}$) to be incurred by the CDT manager throughout service of the arriving trucks, which is composed of the outlined cost components: (i) total cost of waiting; (ii) total cost of service; (iii) total cost of product storage; (iv) total cost of delay on the departure of trucks; and (v) total cost of perishable product decay. Constraint set (5) shows that each arriving truck will be served at one of the available CDT doors in one of the service orders. Constraint set (6) indicates that at most only one truck can be further scheduled for service at one of the available CDT doors in one of the service orders. Constraint set (7) guarantees that the arriving inbound trucks will not be served at any of the available shipping doors. Constraint set (8) ensures that the arriving outbound trucks will not be served at any of the available receiving doors. Constraint set (9) ensures that the truck service can start after the arrival of these trucks at the CDT. Constraint set (10) shows that a given truck can only be served after the service of all the preceding trucks at a given CDT door is completed.

Constraint set (11) guarantees that a given outbound truck may only be served after the inbound truck, which delivers the products for that particular outbound truck, has started service. Constraint set (12) estimates total service time for each truck which is dependent on the handling time of truck (i.e., loading or unloading) and the product transfer time to the dedicated temporary location of storage. Constraint sets (13) through (15) calculate the time of service finish, the time of waiting, and the time of delayed departure of trucks at the CDT. Constraint set (16) calculates the total handling and transfer time for each one of the product types that have been transported to the CDT by one of the arriving inbound trucks for one of the arriving outbound trucks. Constraint set (17) calculates the temporary storage time for each one of the product types that have been transported to the CDT by one of the arriving inbound trucks for one of the arriving outbound trucks. Constraint set (18) computes the product quality for a given product, delivered to the CDT by a given inbound truck for a certain outbound truck, at time τ. Constraint set (19) estimates the decrease in quality for each one of the product types that have been transported to the CDT by one of the arriving inbound trucks for one of the arriving outbound trucks. Constraint sets (20) through (22) show the nature of the variables and the nature of the parameters of the proposed mixed integer **TSP-CDT** mathematical model.

4.2 Linearization of Non-linear Components

The proposed mixed integer **TSP-CDT** in this study is a non-linear mathematical model due to: 1) the objective function (specifically, the total cost of perishable product decay component); and 2) constraint sets (18) and (19). The factors, considered in the estimation of the decay of a given product p, include the total handling and transfer time $\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}$ and the product's specific decay rate denoted as λ_p. In the literature, non-linear functions are approximated using a number of common approaches, including the following:

1) linear static-approximation method – this method adds linear constraints to the model by generating tangent lines that guarantee effective approximation;
2) linear dynamic outer-approximation method – this method generally performs with fewer tangent lines when compared with its static counterpart (depending on the target approximation accuracy); the tangent lines are generated dynamically considering a specific approximation gap, i.e. the number of tangent lines to be generated is determined by the approximation gap;
3) linear branch-and-bound outer-approximation method – this method starts with a few tangent lines and uses the branch-and-bound strategy to branch into a narrowed feasible range; it does the narrowing by eliminating a tangent line (outside the range) and creating a new one; this process is repeated until the desired approximation accuracy is achieved;
4) quadratic static outer-approximation method – this method uses a set of parabolic curves instead of straight lines to approximate a non-linear function; although this approach performs better than using straight lines, it is more time-consuming due to non-linearity;

5) linear static secant-approximation method – this method uses secant lines instead of tangent lines; to achieve the same level of accuracy in approximating non-linear functions, fewer secant lines are needed compared to tangent lines (Wang et al., 2013; Dulebenets and Ozguven, 2017).

This study will adopt the linear static secant method of approximation, assuming that the non-linear component is linearized using its piecewise linear secant approximation A_s^0, where s indicates the number of linear segments. Let's denote $A_p^n(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}) = \frac{\boldsymbol{\Delta Q}_{\underline{t}\bar{t}p}}{Q_{\underline{t}\bar{t}p}^0} = \frac{Q_{\underline{t}\bar{t}p}^0 - \boldsymbol{Q}_{\underline{t}\bar{t}p}^{\tau}}{Q_{\underline{t}\bar{t}p}^0} = \frac{Q_{\underline{t}\bar{t}p}^0 - Q_{\underline{t}\bar{t}p}^0 e^{-\lambda_p \tau_{\underline{t}\bar{t}p}^{tht}}}{Q_{\underline{t}\bar{t}p}^0} = 1 - e^{-\lambda_p \tau_{\underline{t}\bar{t}p}^{tht}}$ as the decay function of product p. $A_p^n(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht})$ can be linearized for any given product p using its piecewise linear secant approximation denoted by $A_{ps}^0(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht})$.

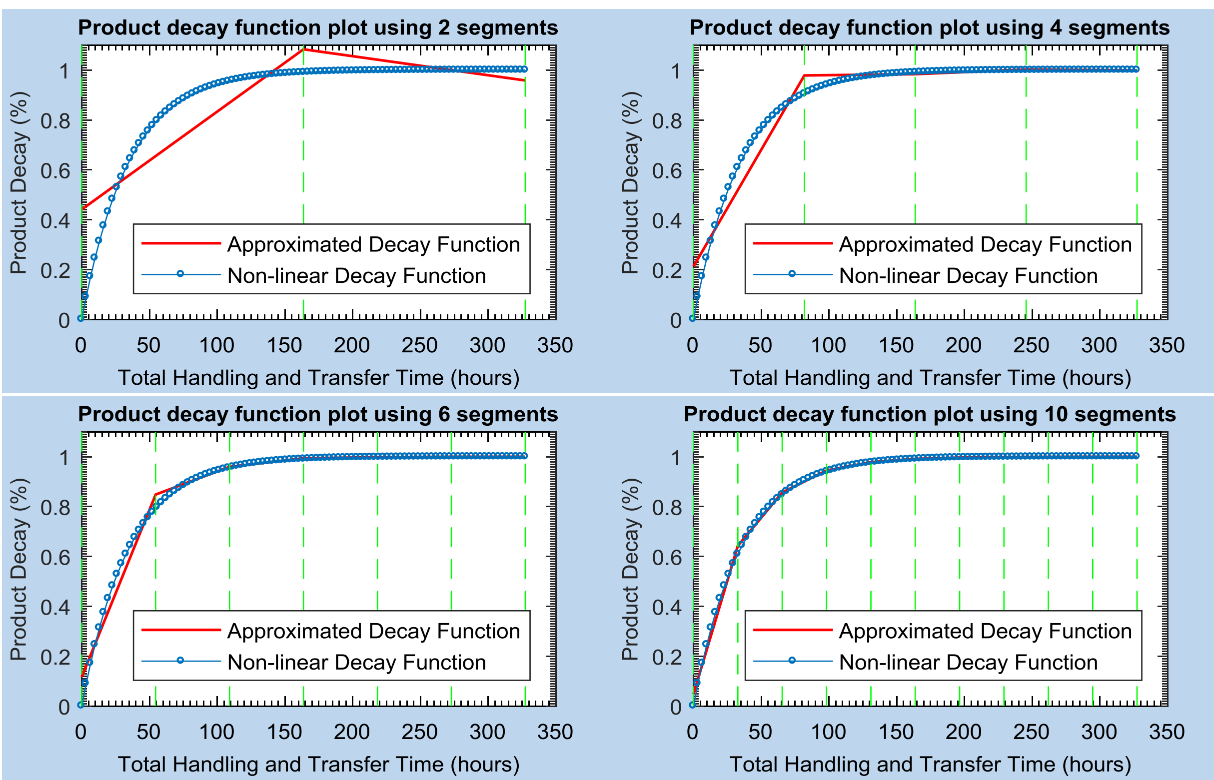


Figure 20: Approximation of the decay function of the perishable products.

Figure 20 provides examples of the approximations of the decay function given some numbers of linear segments ($s = 2, 4, 6, 10$). The total handling and transfer time $\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht}$ was assumed to vary from 0 hrs. to 327 hrs., i.e. $0\,\mathrm{h} \leq \boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht} \leq 327\,\mathrm{h}$. The decay rate λ_p was assumed to range $0.0124\mathrm{h}^{-1} \leq \lambda_p \leq 0.0291\mathrm{h}^{-1}$ for the total of ten products considered. Specifically, the decay rates are: $\lambda_p = [0.0287, \ldots, 0.0235]$ for $p \in [1, \ldots, 10]$. The decay rate λ_p for the product decay functions, illustrated in Figure 20, was set to be 0.0235 h^{-1}. From the examples shown in Figure 20, it can be observed that the accuracy of A_{ps}^0 approximation increases with increasing number of linear segments.

Let $S = \{1,2, \ldots, w^{10}\}$ be the set of linear segments in the piecewise function $A_{ps}^0(\boldsymbol{\tau}_{\underline{t}\bar{t}p}^{tht})$. Let $\boldsymbol{d}_{\underline{t}\bar{t}ps}^0 = 1$ if a linear segment s is used to approximate the decay function of product p that has been transported to the CDT by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ (=0 otherwise). Denote str_{ps}^0, en_{ps}^0 as the total handling and transfer time values of product p for the beginning and the ending, respectively, of a linear segment s; Slp_{ps}^0, Inc_{ps}^0 as the slope and intercept, respectively, of the linear segment s used to approximate the decay function of product p; and M is a large positive number. The **TSP-CDT** can, therefore, be represented as a linear problem (**TSP-CDTL**) as follows.

TSP-CDTL: Linearized Problem of Truck Scheduling at the CDT with Perishable Products

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

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 **TSP-CDTL**, the objective function (23) focuses on minimization of the total cost ($\boldsymbol{TC}$) to be incurred by the CDT manager throughout service of the arriving trucks. Constraint set (24) ensures that the decay function of product p that has been transported to the CDT by inbound truck $\underline{t}$ for outbound truck $\bar{t}$ will be approximated using only one linear segment. Constraints sets (25) and (26) delimit a range of values for total handling and transfer time values, when a linear segment s is chosen to approximate the decay function of product p that has been transported by inbound truck $\underline{t}$ for outbound truck $\bar{t}$. Constraint set (27) determines the approximated value of the decay of product p that has been transported to the CDT by inbound truck $\underline{t}$ for outbound truck $\bar{t}$. Given the linearization of the non-linear component of the objective function, and non-linear constraints of the previously formulated **TSP-CDT** model, the model has been reformulated as **TSP-CDTL**. Therefore, an exact mixed integer linear optimization technique, e.g. CPLEX, can be used to solve the model to optimality.

CHAPTER FIVE

5 SOLUTION METHODLOGIES

The mathematical formulation of the problem of truck scheduling, denoted in this study as **TSP-CDT**, can be reduced to an unrelated machine scheduling problem. This problem's complexity is usually characterized as NP-hard. Realistic-size problem instances cannot be solved in a reasonable computational time; therefore, the development of heuristics or metaheuristic algorithms is required to obtain good-quality solutions in a reasonable computational time. The following four metaheuristic algorithms are considered in this dissertation: (1) Evolutionary Algorithm (EA); (2) Variable Neighborhood Search (VNS); (3) Tabu Search (TS); and (4) Simulated Annealing (SA).

5.1 Evolutionary Algorithm (EA)

5.1.1 Steps in the proposed EA algorithm

The main steps of the proposed EA algorithm are presented in Pseudocode 1 and Figure 21. The data structures required are initialized in step 0. The population of chromosomes is initialized in step 1 using the First Come First Served (FCFS) policy with truck sequence considerations (see section 5.1.3 for more details). The evaluation of the initial chromosomes for fitness is done in step 2. The algorithm is stopped if the termination criterion is met; otherwise, it proceeds to the next step.

```
Pseudocode 1: Evolutionary Algorithm (EA)
START
0: Initialization of data structures
1: Initialization of chromosomes/population
2: Computation of fitness values
If termination criterion is reached - terminate, else go to steps 3-7.
3:  Parent selection
4:  Crossover operation
5:  Mutation operation
6:  Computation of fitness values
7:  Survivor selection
END
```

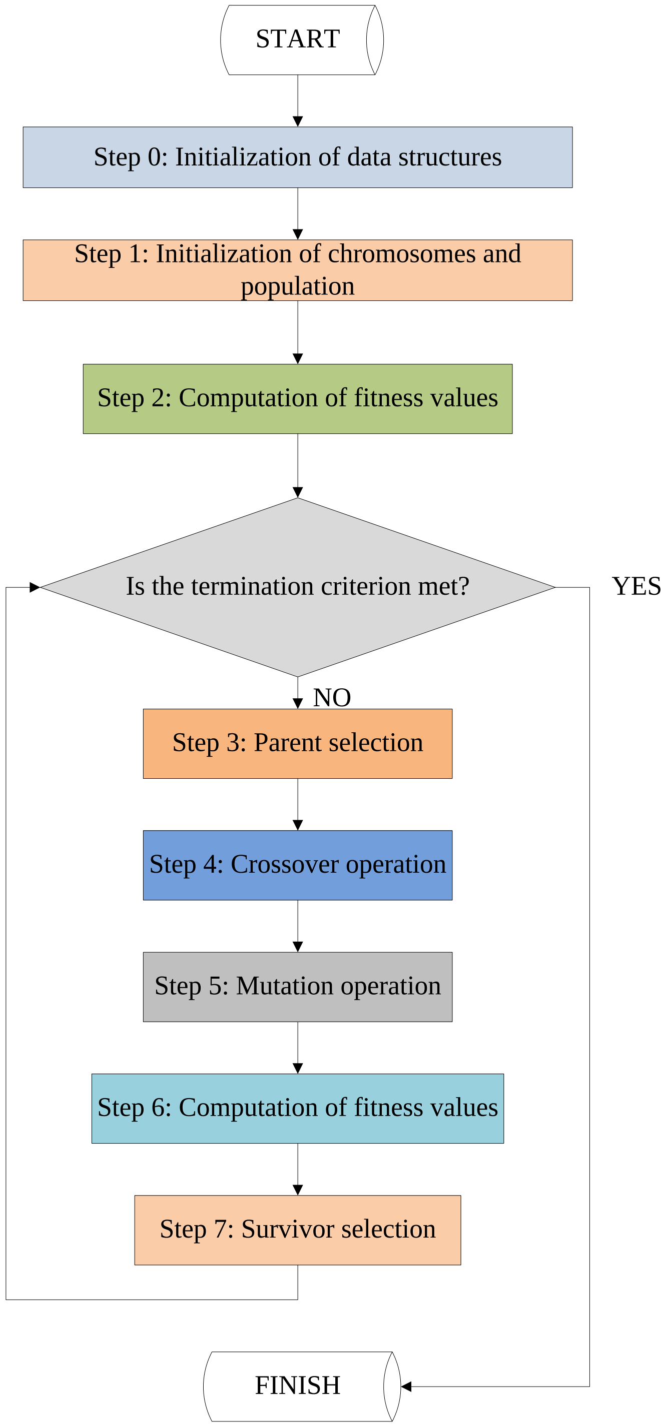


Figure 21: EA algorithm steps.

The parent selection is done in step 3. Then, the EA operation begins in step 4. In particular, the algorithm uses the crossover operator in step 4 and the mutation operator in step 5 to explore and exploit the search space domains, respectively, for promising solutions. The resulting chromosomes, also referred to as the offspring, are evaluated for fitness in step 6. In step 7, the survivors for the next generation are selected. More specifically, the fitness values of the generated offspring chromosomes are used to determine which of them would be used as potential parents in the next generation within the algorithm's iterative process. If the termination criterion is met, the algorithm terminates, otherwise, steps 3 – 7 are repeated until the termination criterion is met. The iterative process terminates upon convergence, and the EA algorithm returns the best solution (with the least total cost), representing a truck schedule.

5.1.2 **Chromosome representation**

In the proposed EA algorithm, the candidate solutions to the proposed mathematical model are represented with chromosomes. These chromosomes are encoded with integer values, which would be appropriate for the **TSP-CDT** mathematical model (as opposed to the canonical Genetic Algorithms that use binary chromosomes). The information, contained in a chromosome, includes the order of service of the arriving trucks and their respective assigned doors. An illustrative example of a chromosome is given in Figure 22. This chromosome represents the service order and the door assignment for 9 trucks, where inbound trucks "3","6","7", and "9" are assigned for service at door "1" (therefore, door "1" was designated to serve as a receiving door by the CDT manager). Note that inbound truck "3" is served first at door "1", followed by trucks "6" and "7", while truck "9" is served as the last truck. Also, outbound trucks "1","4", and "5" are assigned for service at door "2", and outbound trucks "2" and "8" are assigned for service at door "3 (therefore, doors "2" and "3" were designated to serve as shipping doors by the CDT manager).

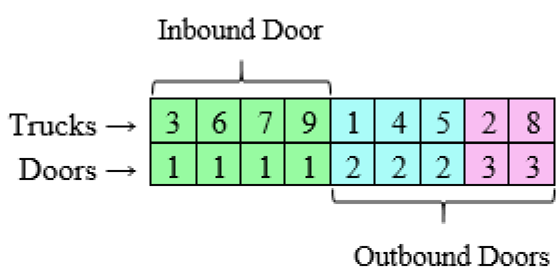


Figure 22: Chromosome representation for EA.

5.1.3 Population initialization

In typical EA-based algorithms, the population chromosomes are randomly generated. Randomly generated chromosomes may, however, result in many infeasible solutions (i.e., some of the generated chromosomes contain the outbound trucks assigned before the corresponding inbound trucks, which does not occur in reality). The generic First Come First Served (FCFS) policy, which is often used in the literature, will not be applied in this study in order to avoid the violation of the CDT logistics requirements. This study, therefore, adopts the FCFS policy with truck sequence considerations (named as "FCFS-TSC") to address this drawback. The main steps, involved in the FCFS-TSC procedure, are presented in Pseudocode 2.

Pseudocode 2: First Come First Served Policy with Truck Sequence Consideration (FCFS-TSC)

$\boldsymbol{FCFS-TSC}(T, T^{in}, T^{out}, D, O, N, \tau_t^{at}, \tau_t^{ht}, \tau_{dn}^{tr}, z_{tn})$

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 locations of storage areas; τ_t^{at}-arrival time for truck t; τ_t^{ht}-handing time of truck t; τ_{dn}^{tr}-transfer time from door d to location of storage n; z_{tn}-product to location of storage 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: $T^{IS} \leftarrow \boldsymbol{Order}(T^{in}, \tau^{at})$ ◁Arranging inbound trucks in the order of arrival
2: $T^{OS} \leftarrow \boldsymbol{Order}(T^{out}, \tau^{at})$ ◁Arranging outbound trucks in the order of arrival
3: $T^S \leftarrow T^{IS} \cup T^{OS}$ ◁Combination of arranged trucks both in- and outbound
4: **for** $t = 1{:}\,|T^S|$ **do**
5: $d \leftarrow argmin(\boldsymbol{\tau}_d^{AV})$ ◁Selection of the CDT door that is available first
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: $\boldsymbol{\tau}_t^{st} \leftarrow max(\boldsymbol{\tau}_t^{at}, \boldsymbol{\tau}_d^{AV})$ ◁Estimation of truck service start time
9: $\boldsymbol{\tau}_{tdo}^{tst} \leftarrow \tau_t^{ht} + \sum_{n \in N}(\tau_{dn}^{tr} z_{tn})$ ◁Estimation of the truck total service time
10: $\boldsymbol{\tau}_t^{ft} \leftarrow \boldsymbol{\tau}_t^{st} + \boldsymbol{\tau}_{tdo}^{tst}$ ◁Estimation of the truck service finish time
11: $\boldsymbol{\tau}_d^{AV} \leftarrow \boldsymbol{\tau}_t^{ft}$ ◁Door availability update
12: $t \leftarrow t + 1$
13: **end for**
14: **return** $\boldsymbol{x}$

In step 0, the FCFS-TSC heuristic initializes all the data structures required. Then, the trucks both in- and outbound are sorted in the order of their arrival times in steps 1-2. In step 3, the sorted sets of trucks both in- and outbound are merged together. The main function loop begins in step 4 and ends in step 13. In step 5, the first available CDT door is selected from the set of doors. In step 6, the earliest truck service order is selected from the set of truck service orders. In step 7, the next truck in line of sorted arriving trucks (i.e., inbound and outbound trucks) is assigned for service at the selected door and in the selected truck service order. In step 8, the truck service start time is estimated. In step 9, the total truck service time is computed as the sum of the total handling time and the total transfer time. The truck service finish time is estimated in step 10. The CDT door availability is updated in step 11. The FCFS-TSC exits the loop after the last truck in the set of sorted trucks both in- and outbound has been scheduled for service at one of the available CDT doors. In step 14, the updated service order of truck-to-door assignment is returned.

The FCFS approach and its modifications have been deployed widely in the freight terminal operations literature and the EA literature (Dulebenets, et. al. 2018a,b; Dulebenets, 2019a; Kavoosi et. al., 2020). The included sequence consideration ensures that the initial chromosomes are generated with a special consideration for inbound truck and outbound truck assignment priorities (i.e., the inbound trucks should be scheduled for service before the outbound trucks in order to avoid the situation, when the outbound truck is assigned for service before the inbound truck, bringing the products for that outbound truck). The first half of the initial chromosomes is generated using the FCFS-TSC strategy. However, the FCFS strategy generates identical chromosomes. This is a drawback that was previously identified in the literature and highlighting the required diversity in the initial population. Therefore, the FCFS-TSC will be used to initialize one half of the population, while the second half of the population will be randomly generated (for more details, refer to Dulebenets et al., 2018a,b).

5.1.4 Parent selection strategy

After the generation of the initial population, the proposed algorithm evaluates the fitness of each chromosome. This step determines the chromosomes that are best fit for further generation of the offspring chromosomes through the proposed EA algorithm. There are diverse mechanisms that can be used to select eligible parent chromosomes. For example, Dulebenets (2018a) adopted the fitness proportionate selection mechanism (also referred to as “roulette wheel selection”) that is based on a stochastic process, where certain individuals have higher probabilities of becoming parents due to their higher fitness values. Other parent selection mechanisms include tournament selection, ranking selection, random selection, and binary tournament selection (Dulebenets, 2018a,b). This study adopts the "*Tournament Selection*" mechanism. The main steps, involved in the Tournament Selection procedure, are outlined in Pseudocode 3.

```
Pseudocode 3: Tournament selection (Tournament)
Tournament(Pop, Fitness, TourSize, TourSel)
in: Pop-population in a given generation; Fitness-fitness of chromosomes in a
given generation; TourSize-the size of tournament; TourSel-the count of the
chromosomes selected in each tournament
out: Parents-parent chromosomes in a given generation
0:  |Parents| ← ∅    ◁Initialization
1:  for i = 1:(|Pop|/TourSel)
2:    [Tour, Fitness^Tour] ← Rand(Pop, Fitness, TourSize)   ◁Select chromosomes for the
          tournament
3:    for j = 1:TourSel
4:    k* ← argmin(Fitness^Tour)   ◁Fittest chromosome search
5:    Parents ← Parents ∪ {Tour_k*}  ◁Fittest chromosome selection as a parent
6:    Tour ← Tour - {Tour_k*}   ◁Fittest chromosome removal from the
        tournament
7:    j ← j + 1
8:    end
9:    i ← i + 1
10: end
11: return Parents
```

The selection process begins by creating the required data structure in step 0. Then, it enters a loop, starting in step 1 and ending in step 10. The chromosomes are chosen at random to participate in the tournament in step 2. The inner loop starts in step 3 and ends in step 8. Within that inner loop, the fittest chromosome is extracted from a given tournament (step 4) and selected as a parent (step 5). The fittest chromosome, which was selected as a parent, is removed from a given tournament is step 6. A total of $TourSel$ chromosomes are selected as parent chromosomes inside the inner loop. The tournaments are launched repeatedly until the required number of parent chromosomes is selected.

5.1.5 EA operations

An Evolutionary Algorithm (EA) is a technique adopted from the biological science, which is based on the genetics and natural selection theory that was initially proposed in the year of 1859 by Charles Darwin (Darwin, 1859). The theory relies on the concept "survival of the fittest". It basically means that individuals with higher fitness values have higher chances of surviving and procreating. The EA, therefore, uses this principle to search for the solutions, or otherwise known as individuals with higher fitness values. The EA-based algorithms generally deploy two basic operations throughout the search process, including the crossover operation and the mutation operation. Details regarding the crossover operation and the mutation operation within the developed EA are presented in the following sections.

5.1.5.1 Crossover operation

The deployment of the crossover operator in an EA algorithm is important for the exploration of promising search space domains. There are many crossover operators that have been previously applied in the EA literature. Examples of crossover operators include: one-point crossover, multi-point crossover, uniform crossover, simple arithmetic crossover, among others (Eiben and Smith, 2015). The aforementioned crossover operators are generally used for the chromosomes with binary representation and will not be applicable in this study, as integers are used for the representation of chromosomes in this study (see Figure 22).

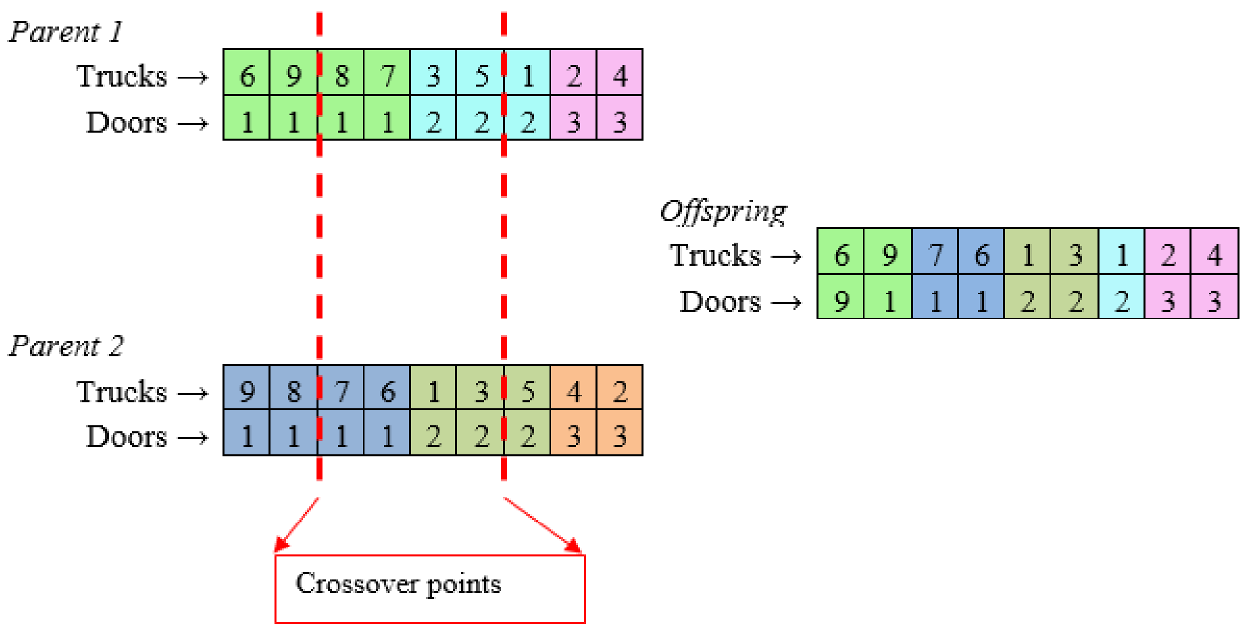


Figure 23: Infeasible offspring generated by two-point crossover.

Application of the inappropriate crossover operators will increase the number of infeasible chromosomes in the offspring, since the real-valued offspring can be produced instead of integer ones. Furthermore, infeasible truck schedules could be produced as well. An example of an inappropriate crossover is shown in Figure 23, where the offspring is produced using the two-point crossover operator. This resulted in an infeasible solution as the arriving trucks “6” and “1” were both assigned for service twice, whereas the arriving trucks “8” and “5” were not assigned for service at all. An infeasible offspring is repairable but may require a significant amount of computational effort (Dulebenets, 2020a).

Some suitable crossover operators for the chromosomes represented with integers include Davis’ order crossover, cycle crossover, and partially-mapped crossover (Sivanandam and Deepa, 2008; Dulebenets, 2018a,b). This study adopts the partially-mapped crossover, as this type of crossover typically offers a good performance for various optimization problems as compared to the other types of crossover (Goldberg and Lingle, 1985; Deep and Mebrahtu, 2012).

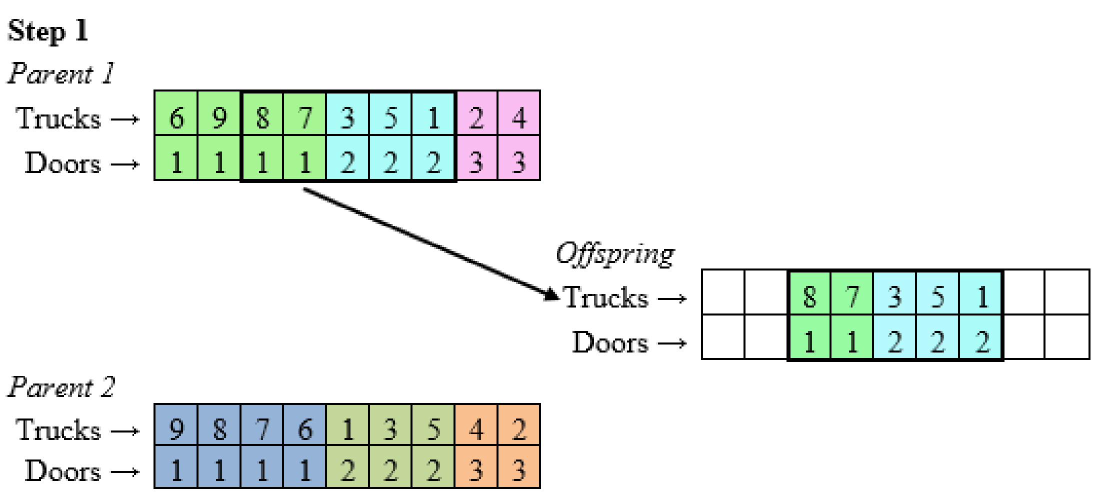


Figure 24: Partially-mapped crossover operation (step 1).

The partially-mapped crossover selects a cut-portion from one of the parent chromosomes in order to produce the offspring, while the missing genes are copied from the other parent chromosome with application of partial mapping (where appropriate). An illustration of the mechanism is given in Figure 24, Figure 25 & Figure 26. First, a random segment is selected from the first parent chromosome, and the genes are copied directly to the offspring. As shown in the example presented (see Figure 24), the genes with trucks “8”, “7”, “3”, “5”, and “1” are copied from parent chromosome “1” to the offspring chromosome. Second, the corresponding segment in parent chromosome “2” is checked for the genes in the parent chromosome that have not been copied. Our example shows that the gene with truck “6” is the only gene that has not been copied (see Figure 25). The partially-mapped crossover operator has to identify the appropriate position in the offspring chromosome for placing the gene with truck “6”. The latter is accomplished based on the following observations.

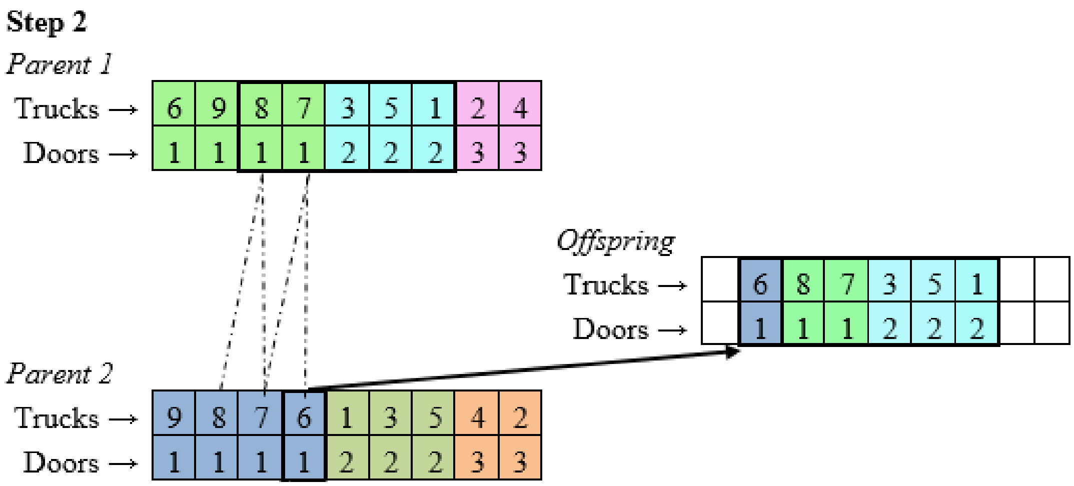


Figure 25: Partially-mapped crossover operation (step 2).

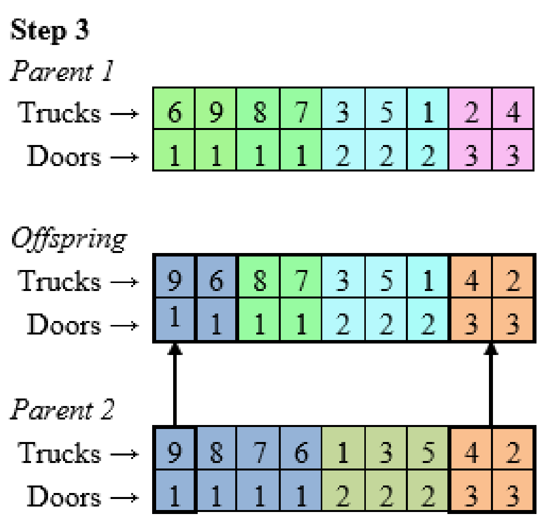


Figure 26: Partially-mapped crossover operation (step 3).

Truck “6” occupies the fourth position (i.e., “locus”) in parent chromosome “2”, while the corresponding position in parent chromosome “1” is occupied by the gene with truck “7”. Truck “7” is found in the third position in parent chromosome “2” and has been previously copied from parent chromosome “1” (see Figure 25). Therefore, the third position in parent chromosome “1” is checked. The third position in parent chromosome “1” is occupied by truck “8”. On the other hand, the gene with truck “8” is found in the second position in parent chromosome “2”. Since the second position of the offspring chromosome is empty, the gene with truck “6” will be copied to position “2” in the offspring chromosome as a result of the partial mapping operation. The remaining genes in parent chromosome “2”, which contain trucks “9”, “4”, and “2”, will be copied to the offspring chromosome in the order they appear in parent chromosome “2” (see Figure 26).

5.1.5.2 Mutation operation

The EA algorithm uses mutation operators to exploit the search space domains for superior solutions (Eiben and Smith, 2015). These operators function by changing the values of one or more genes within a given chromosome. A number of mutation operators have been used in the EA literature. The examples of mutation operators include insert mutation, swap mutation, inversion mutation, random resetting, and scramble mutation (Nitasha and Tapas, 2014). Each of the aforementioned mutation operators has its specific procedures. It is worthy of note that the application of mutation operators may result in the production of infeasible chromosomes due to their stochastic nature. Some studies adopt customized mutation, where one or more mutation procedures are applied along with additional steps that prevent infeasibility as a result of mutation. For example, Dulebenets (2019a) deployed a mutation operator, which combined the features of insert mutation and swap mutation and sorted the mutated offspring by the CDT doors in order to eliminate infeasibility due to disruption in truck service orders at the CDT doors.

This study adopts the inversion mutation operation. In this case, the corresponding genes within a selected portion of the chromosome are inverted. After that, the resulting mutated offspring will be checked for feasibility. An illustration of the performance of the inversion mutation is given in Figure 27. The highlighted portion of the chromosome (with trucks “7”, “3”, “5”, and “1”) is selected randomly. Then, the positions of genes are inverted to produce a completely different chromosome. The resulting chromosome, therefore, has the selected portion of genes in a reverse order as shown in the Figure 27.

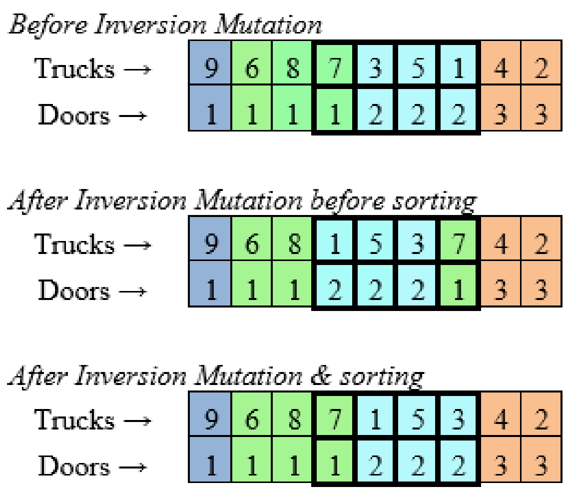


Figure 27: Inversion mutation operation for EA.

The selected genes are in the multiples of two, such that each pair of genes can exchange their positions. Our example shows that the outer genes of the selected chromosome segment with trucks “7” and “1” exchange their positions. Also, the inner genes of the selected chromosome segment with trucks “3” and “5” exchange their positions. After the mutation operation, the genes are sorted by doors as shown in Figure 27 in order to disallow an inappropriate alteration in the truck service orders (i.e., truck “7” is assigned for service at door “1” and has to be placed in a group with the other trucks that are assigned to be served at door “1”, including trucks “9”, “6”, and “8”).

5.1.6 **Fitness evaluation of infeasible individuals**

The fitness value of the chromosome is computed using the following mathematical function within the proposed algorithm:

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

A penalty term (Ψ) used to avoid infeasible chromosomes is included in the fitness function. It is worthy of note that the EA operators used (crossover and mutation) may produce infeasible chromosomes in the process of exploration and exploitation of the search space. The infeasible chromosomes do not represent a realistic truck schedule (i.e., a given outbound truck can be further scheduled for service before the inbound truck, bringing the products for that outbound truck – not before). The value of the penalty term will be set using the parameter turning analysis. Application of the penalty term is important in the algorithm, as it reduces the tendency of producing individuals that are not feasible within the algorithm's iterative process.

5.1.7 **Survivor selection strategy**

This study adopts a fitness-based selection strategy, referred to as "*Roulette Wheel Selection*" mechanism, for survivor selection. This strategy is commonly used at the selection stage of EAs in the literature, such as parent selection or offspring/survivor selection (Eiben and Smith, 2015; Dulebenets, 2018a). The main steps, involved in the Roulette Wheel Selection procedure, are presented in Pseudocode 4. The Roulette Wheel Selection mechanism allocates a portion of the roulette wheel to the chromosomes based on their fitness (i.e., a larger portion of the roulette wheel will be allocated to the chromosome that has a higher fitness). The wheel is rotated and the chromosome that has a larger portion of the roulette wheel has a greater chance to be chosen as the offspring for the next generation. The roulette wheel is repeatedly rotated until the required number of offspring is selected. For more information about this procedure refer to Dulebenets (2018a).

Pseudocode 4: Roulette Wheel Selection (RWS)

$\boldsymbol{RWS}(Pop, Fitness)$

in: Pop-population in a given generation; $Fitness$-fitness of chromosomes in in a given generation

out: $Offspring$-offspring chromosomes in a given generation

0: $Offspring \leftarrow \emptyset$; $|Fitness^{aux}| \leftarrow |Fitness|$ ◁Initialization
1: **for** $i = 1{:}|Pop|$ **do**
2: $Fitness_i^{aux} \leftarrow 1/Fitness_i$ ◁Chromosome fitness adjustment
3: $i \leftarrow i + 1$
4: **end for**
5: $Fitness^{aux} \leftarrow \boldsymbol{Normalize}(Fitness^{aux})$ ◁Normalize the adjusted fitness values of the population chromosomes
6: **for** $i = 1{:}|Pop|$ **do**
7: $Val \leftarrow \boldsymbol{Rand}(0.00; 1.00)$ ◁Generate a random number between 0.00 and 1.00
8: $i \leftarrow \boldsymbol{find}(Fitness^{aux} - Val > 0)$ ◁Select the chromosome based on step 7
9: $Offspring \leftarrow Offspring \cup \{Pop_i\}$ ◁Updating the offspring with the selected chromosome
10: $i \leftarrow i + 1$
11: **end for**
12: **return** $Offspring$

In step 0, the selection process begins by creating the data structures required. The fitness function value is iteratively adjusted for all the chromosomes in steps 1-4 (as the developed **TSP-CDT** mathematical model has a minimization objective). The adjusted fitness values will be normalized in step 5, so the total sum of the fitness values of the population chromosomes will be exactly equal to one. In the final loop that starts in step 6 and ends in step 11, a random number (Val), which ranges between "0.00" and "1.00", is generated (see step 7). Afterwards, the next generation chromosome is selected based on the chromosome with a normalized fitness value that is close to the generated random number (see steps 8 and 9). The procedure is repeatedly executed until the number of required offspring chromosomes is selected.

5.1.8 **Elitism**

The EA algorithm proposed in this study uses the elitism strategy. This involves the preservation of the fittest individual(s) for the following generation before the main algorithmic operators, such as selection of parents, application of crossover, application of mutation, and selection of survivors, are implemented throughout the given EA generation (Dulebenets, 2020b). This is important because the offspring chromosomes, produced from the parent chromosomes through parent selection, crossover, and mutation, are not guaranteed to have a higher fitness value. In fact, the crossover and mutation operators can make the fitness of the offspring chromosomes significantly worse than the fitness of the parent chromosomes as a result of substantial genetic changes in the parent chromosomes.

5.1.9 Termination criteria

The EA algorithm developed in this study terminates once one of the following two termination criteria is met: (i) the maximum number of generations reached; and (ii) no improvement is observed in the quality of the solutions after several generations. Both termination criteria are popular in the EA literature (Eiben and Smith, 2015; Dulebenets, 2018b). Furthermore, both termination criteria depend on the specifics of the considered problem (e.g., the maximum number of generations can be larger for one problem but smaller for another – depending on the complexity of the search space). In order to enhance the performance of the algorithm, a parameter tuning analysis will be carried out to choose the termination criteria in the appropriate manner (Kavoosi et al., 2019).

5.2 Variable Neighborhood Search (VNS) Algorithm

Variable Neighborhood Search (VNS) algorithm is a metaheuristic proposed by Mladenovic and Hansen in 1997 (Mladenovic and Hansen, 1997). VNS performs by systematically making changes within neighborhood data structures. These data structures comprise a finite number of neighborhoods $N_k: k = 1,2,\dots,k_{max}$. Each neighborhood N_k comprises a set of possible solutions; therefore, $N_k(x)$ refers to the set of solutions in the kth neighborhood of x. It uses a local search method to perform a sequence of local changes in the initial solution. These changes are expected to improve the value of the objective function until a local optimum value is found. Then, the algorithm moves to a distant neighborhood and explores it locally for improved solutions until the stopping criterion is reached. In the VNS algorithm, a local optimum refers to the best solution in a neighborhood, while the global optimum refers to the best solution with respect to all possible neighborhoods (however, for some problems all possible neighborhoods may not be feasible to represent – i.e., the decision problems with a complex search space and high computational complexity). This algorithm has wide applications in location theory, cluster analysis, vehicle routing, and scheduling (Brimberg, 1996; Hansen and Mladenovic, 2001). VNS generally allows for more efficient exploitation of solution search space (Dulebenets, 2019a).

```
Pseudocode 5: Variable Neighborhood Search (VNS) Algorithm
START
0: Initialization of data structures
1: Initialization of solutions for the first neighborhood
2: Computation of fitness values
If termination criterion is reached - terminate, else go to steps 3-5.
3:  Neighborhood local search
4:  Computation of fitness values
5:  Neighborhood change
END
```

5.2.1 Steps in the proposed VNS

The main steps of the proposed VNS algorithm are presented in Pseudocode 5 and Figure 28. The initialization of the required data structures is done in step 0. A pool of solutions for the first neighborhood is generated in step 1 using two strategies: 1) the First Come First Served policy with Truck Sequence Considerations (FCFS-TSC) for the first half; 2) the other half is generated randomly. This initialization method is in line with some of the previously conducted studies related to the CDT truck scheduling (Yu and Egbelu, 2008; Vahdani and Zandieh, 2010; Dulebenets, 2019a). The fitness values of the generated solutions within the first neighborhood are computed in step 2. The fittest solution from the candidate solutions will be further used throughout the local search for the first neighborhood. Then, the algorithm is stopped if the termination criterion is met; otherwise, it proceeds to the next step.

A local search is done in step 3 for a given neighborhood, aiming to discover more promising solutions. The new solutions, which were identified as a result of a local search within a given neighborhood, are evaluated for their fitness in step 4. In step 5, the algorithm moves to a more promising neighborhood in order to search for more promising solutions. Steps 3-5 are repeated until the termination criterion is met. The solutions, obtained by VNS, are dependent on different factors, such as the initial solution, the selected neighborhoods, and the local search methods (Rajabi and Shirazi, 2016).

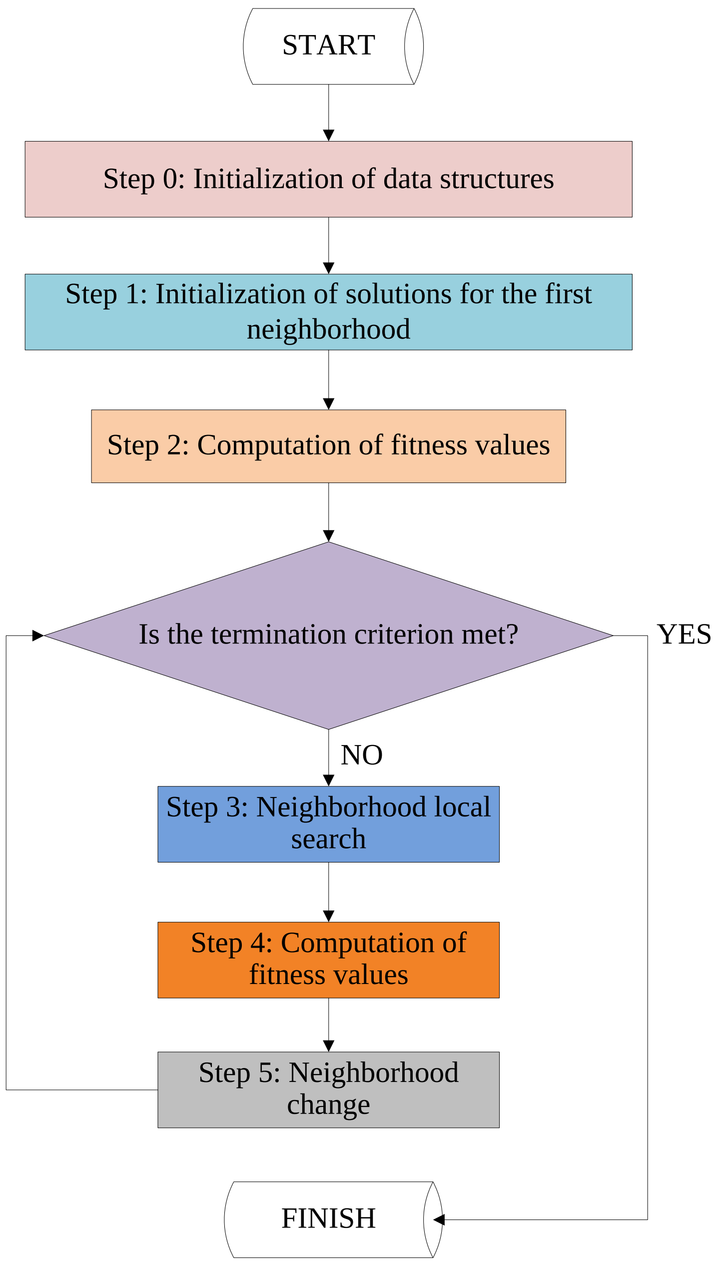


Figure 28: VNS algorithm steps.

5.2.2 **Representation of solutions**

The solution is an integer-coded representation of the real truck scheduling information. It includes the service order of arriving trucks, and their respective assigned doors. An example of this representation is given in Figure 29. The solution represents the service order for 9 trucks, where inbound trucks “5”, “6”, “7”, and “3” are assigned to be served at door “1” (i.e., truck “5” is served first, truck “6” is served second, truck “7” is served third, and truck “3” is served fourth). Also, outbound trucks “1”, “4”, and “9” are assigned to be served at door “2”, while outbound trucks “2” and “8” are assigned to be served at door “3”. Door “1” is designated as an inbound door, while doors “2” and “3” are designated as outbound doors.

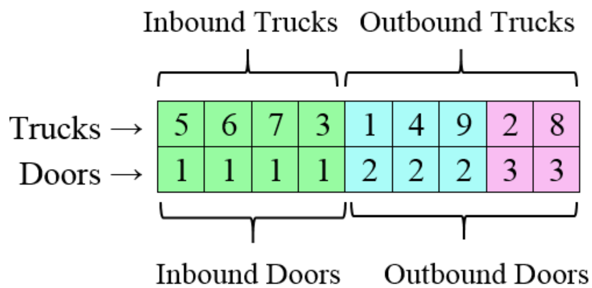


Figure 29: Solution representation for VNS.

5.2.3 **Initialization of solutions**

Canonical EA-based algorithms generate their initial solutions randomly. VNS is a single-solution based algorithm; therefore, randomly generated data may result in the generation of infeasible solutions within the first neighborhood (i.e., some of the generated solutions contain the outbound trucks assigned for service before the corresponding inbound trucks). As discussed earlier, this study relies on the following two strategies to generate the pool of solutions for the first neighborhood: 1) the First Come First Served policy with Truck Sequence Considerations (FCFS-TSC) for the first half; 2) the other half is generated randomly. Such strategy is expected to ensure feasibility of truck schedules as well as their diversity within the first neighborhood. The fittest solution from the candidate solutions will be further used throughout the local search for the first neighborhood. The effectiveness of the algorithm directly depends on the quality of the initially generated solutions (Vahdani and Zandieh, 2010).

5.2.4 **Neighborhood local search**

The local search method used by VNS is a key determining factor of the overall performance of the algorithm. This method exploits a given neighborhood for superior solutions, aiming to find the local optimum. Some of the local search methods, found in the literature, include (Vahdani and Zandieh, 2010): 1) adjacent pairwise interchange – a randomly chosen truck is exchanged with its adjacent truck; 2) removal and insertion of a truck – a truck is deleted from its position and inserted in a new position, all the trucks are shifted by a step after that; 3) inversion between two random points – this method involves the selection of two random points and reversing the subsequence of trucks between them; and others. This study applies the inversion between two random points. An example of the performance of this local search method is presented in Figure 30.

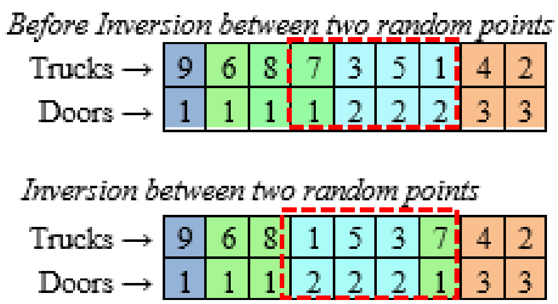


Figure 30: Inversion for neighborhood local search.

A segment of the solution is selected, which has trucks “7”, “3”, “5”, and “1” (see Figure 30). Then, this sequence of trucks is inverted alongside their respective doors. After that, the solution is sorted in the order of doors (see Figure 31). The application of this method, however, is based on a chosen probability, which will be determined during the parameter tuning analysis. This study considers [0.01; 0.02; 0.05] as the candidate probability values of applying inversion on a given truck, which belongs to the considered solution.

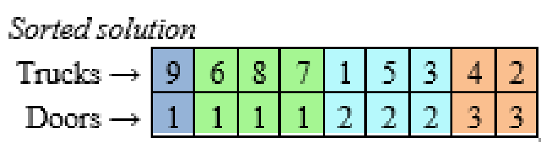


Figure 31: Sorted VNS solution.

5.2.5 **Neighborhood change**

As indicated earlier, the VNS algorithm is characterized by a finite number of neighborhood structures, each of which represents a set of solutions. After evaluating fitness of the solutions, generated as a result of the local search, VNS will select the most promising solution. The most promising solution will be further exploited for the solutions with superior objective function values within its neighborhood in the next iteration of the algorithm. The local search method will be applied on the new neighborhood to obtain the local optimum.

5.2.6 Termination criteria

The VNS algorithm developed in this study terminates once one of the following two termination criteria is met: (i) the maximum number of iterations reached; and (ii) no improvement is observed in the quality of the solutions after several iterations. Both termination criteria are popular in the literature (Dulebenets, 2019a). Furthermore, both termination criteria depend on the specifics of the considered problem (e.g., the maximum number of iterations can be larger for one problem but smaller for another – depending on the complexity of the search space). In order to enhance the performance of the algorithm, a parameter tuning analysis will be carried out to choose the termination criteria in the appropriate manner (Kavoosi et al., 2019).

5.3 Tabu Search (TS) Algorithm

Tabu Search (TS) is a metaheuristic algorithm proposed by Fred W. Glover in 1986 (Glover, 1986). Tabu Search relies on local search methods to iteratively move from one potential solution S to an improved solution S'. The word "tabu" describes the prohibition of the search method from visiting previous solutions. There are three memory structures in tabu search: 1) short-term memory – the list of considered solutions; 2) intermediate-term memory – rules intended to direct the search towards more promising areas in the search space; and 3) long-term memory – rules intended to direct the search to new search area (i.e., when the search gets trapped in a local optimum). The basic idea of Tabu Search is the avoidance of being trapped in a local optimum by allowing the acceptance of a worse solution (Pawar and Hanchate, 2014). Tabu Search has been widely used in the areas of scheduling, financial analysis, logistics, resource planning, etc.

5.3.1 Steps in the proposed TS

The main steps of the proposed TS algorithm are presented in Pseudocode 6 and Figure 32. The initialization of the required data structures is done in step 0. A pool of solutions for the first neighborhood is generated in step 1 using two strategies: 1) the First Come First Served policy with Truck Sequence Considerations (FCFS-TSC) for the first half; 2) the other half is generated randomly. This initialization method is in line with some of the previously conducted studies related to the CDT truck scheduling (Yu and Egbelu, 2008; Vahdani and Zandieh, 2010; Dulebenets, 2019a). A tabu list that will be used to store the record of visited solutions is also initialized. In step 2, the fitness values of the generated solutions are computed. The fittest solution from the candidate solutions will be further used throughout the local search for the first neighborhood.

```
Pseudocode 6: Tabu Search (TS) Algorithm
START
0: Initialization of data structures
1: Initialization of solutions for the first neighborhood
2: Computation of fitness values
If termination criterion is reached - terminate, else go to steps 3-5.
3:  Generation list of candidate neighbors
4:  Computation of fitness values
5:  Identify the best solution S'
If solution S' is in the tabu list, select the next best solution and go to
step 3 else go to step 6
6:  Include solution S' in the tabu list and remove the oldest solution from
the tabu list when the maximum tabu list size is reached
END
```

Then, the algorithm is stopped if the termination criterion is met; otherwise, it proceeds to the next step. In step 3, a list of candidate neighbors around a given solution is generated. In step 4, the fitness values of the candidates are computed. In step 5, the fittest candidate solution (S') is identified. If the candidate solution S' is contained in the tabu list, the algorithm selects the next best solution and goes back to step 3 (as S' cannot be selected since it is in the tabu list), otherwise, it proceeds to step 6. In step 6, the tabu list is updated by adding the candidate solution S', and the oldest solution is removed from the tabu list when the maximum tabu list size is reached. Steps 3-6 are continuously executed until the termination criterion is met.

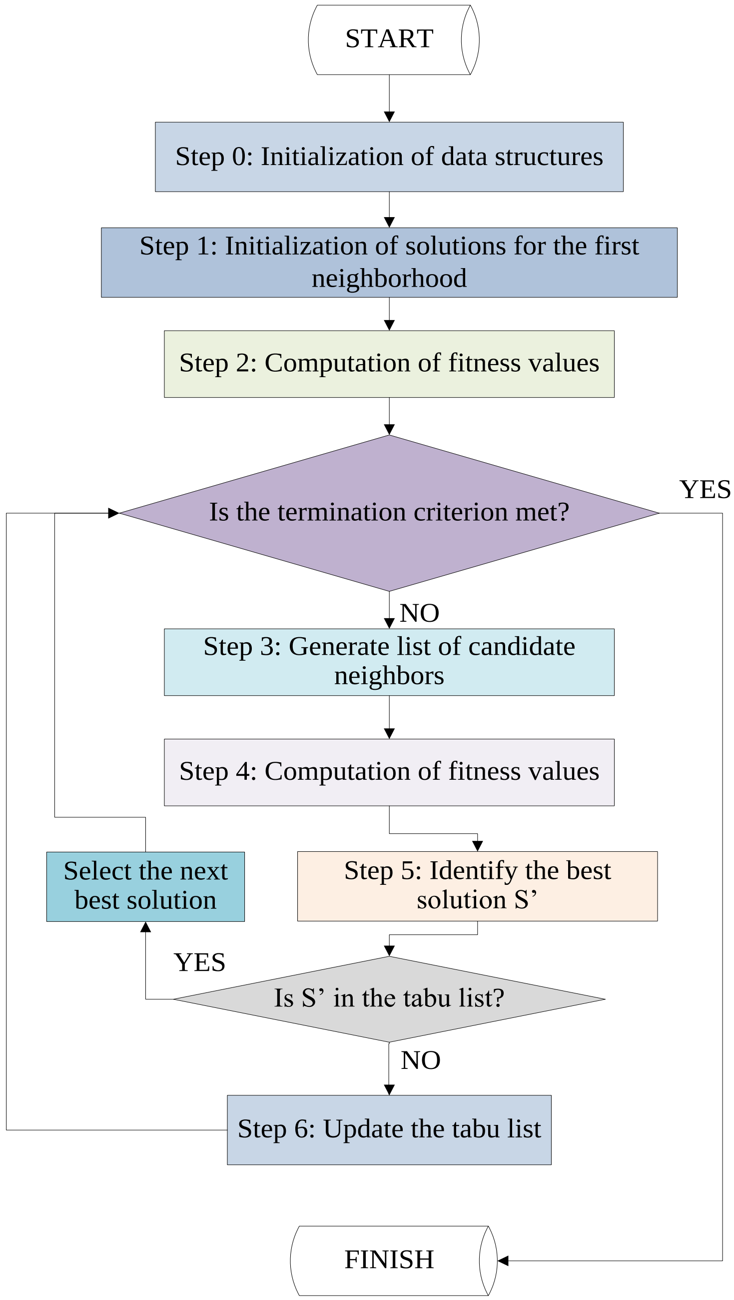


Figure 32: TS algorithm steps.

5.3.2 **Representation of solutions**

The solution is an integer-coded representation of the real truck scheduling information. It includes the service order of arriving trucks, and their respective assigned doors. An example of this representation is given in Figure 33. The solution represents the service order of 9 trucks, where inbound trucks “4”, ”6”, “7”, “8”, and “3” are assigned to be served at door “1” (i.e., truck “4” is served first, truck “6” is served second, truck “7” is served third, truck “8” is served fourth, and truck “3” is served fifth). Also, outbound trucks “2” and “1” are assigned to be served at door “2”, while outbound trucks “5” and “9” are assigned to be served at door “3”. Door “1” is designated as an inbound door, while doors “2” and “3” are designated as outbound doors.

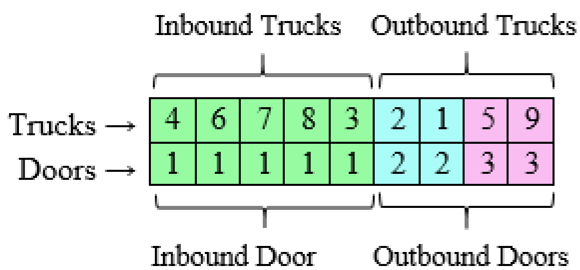


Figure 33: Solution representation for TS.

5.3.3 **Initialization of solutions**

The effectiveness of the algorithm directly depends on the quality of the initially generated solutions (Vahdani and Zandieh, 2010). Canonical EA-based algorithms commonly generate their initial solutions randomly. The use of randomly generated data for the TS algorithm, however, may result in the generation of infeasible solutions within the first neighborhood (i.e., some of the generated solutions contain the outbound trucks assigned for service before the corresponding inbound trucks). As discussed earlier, this study relies on the following two strategies to generate the pool of solutions for the first neighborhood: 1) the First Come First Served policy with Truck Sequence Considerations (FCFS-TSC) for the first half; 2) the other half is generated randomly. Such strategy is expected to ensure the feasibility of truck schedules as well as their diversity within the first neighborhood. The fittest solution from the candidate solutions will be further used throughout the local search for the first neighborhood.

5.3.4 **Generation of the list of candidate neighbors**

Several methods have been used in the literature to generate candidate neighbors. Miao et al. (2009) used two methods, which are “insert move” and “interval exchange move”. These methods were used with equal probability. The insert move method moves a single truck to a dock other than the currently assigned truck. On the other hand, the interval exchange move exchanges two truck intervals in the current service order. For example, assuming a truck service order of “5”, “3”, “7”, “8”,”9”, where the trucks are served respectively, the insert move changes the service order such that it becomes “5”, “7”, “3”, “8”, “9”. Instead of the previous order, truck “7” is served instead of truck “3” as the second truck.

The interval exchange move, on the other hand, exchanges two truck intervals in the service order. Using the previous example, the interval exchange move changes the service order to “5”, “7”, “3”, “9”,”8”. Therefore, truck “7” is served before truck “3”, and truck “9” is served before truck “8”. In this study, a variation of the interval exchange move, referred to as the segment inversion method, will be used. A segment of the solution is randomly selected, and the order of the trucks in this segment is inverted. An example of the performance of this method is presented in Figure 34. A new solution is generated by selecting a random segment from the initial solution, which has trucks “8”, “6”, and “5” (see Figure 34). Then, this sequence of trucks is inverted alongside their respective doors. After that, the solution is sorted in the order of doors (see Figure 34).

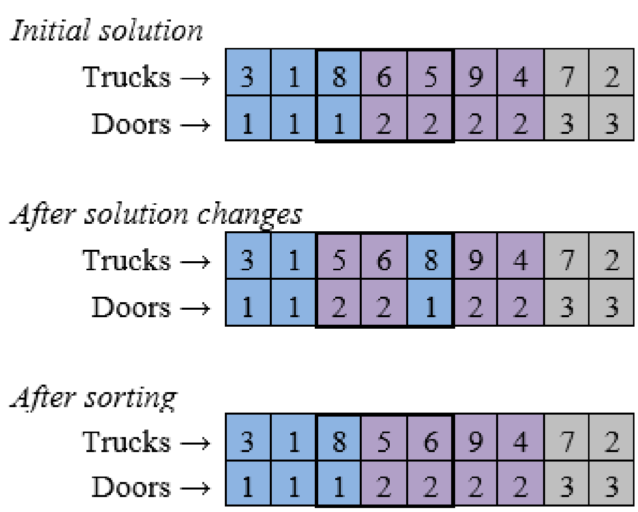


Figure 34: Generation of neighbors for the TS algorithm.

The application of this method, however, is based on a chosen probability, which will be determined during the parameter tuning analysis. This study considers [0.01; 0.02; 0.05] as the candidate probability values of applying inversion on a given truck, which belongs to the considered solution.

5.3.5 Neighborhood local search for the best solution

The proposed TS algorithm uses the fitness evaluation function to select the best solution. The solution S' that would be selected also has to satisfy a specific condition. In particular, the solution must not be contained in the tabu list (i.e., the solution is not same as any of the solutions in the tabu list). The number of solutions in the tabu list is defined by the tabu list size parameter. The tabu list size is a key parameter value for the TS algorithm that will be determined during the parameter tuning analysis. Furthermore, the number of solutions that would be evaluated during the local search will be determined during the parameter tuning analysis. This study considers [30; 40; 50] as the number of solutions to be evaluated during the local search (Dulebenets, 2019a).

5.3.6 Tabu list update

The tabu list refers to the list that contains the record of all the selected solutions over a certain number of consecutive iterations. The list is iteratively updated with the best solutions that meet the update criterion as described earlier. If a solution already exists in the tabu list, it will not be added to the list. The tabu list, therefore, ensures that the algorithm does not accept any repetitive solution. As indicated earlier, the tabu list size will be determined during the parameter tuning analysis. This study considers [10; 15; 20] percent of the total number of the considered solutions during the local search as the tabu list size candidate values. Once the tabu list reaches its maximum size, the oldest solution will be removed from the tabu list to allow the new solutions entering the list.

5.3.7 Termination criteria

The TS algorithm developed in this study terminates once one of the following two termination criteria is met: (i) the maximum number of iterations reached; and (ii) no improvement is observed in the quality of the solutions after several iterations. Both termination criteria are popular in the literature (Dulebenets, 2019a). Furthermore, both termination criteria depend on the specifics of the considered problem (e.g., the maximum number of iterations can be larger for one problem but smaller for another – depending on the complexity of the search space). In order to enhance the performance of the algorithm, a parameter tuning analysis will be carried out to choose the termination criteria in the appropriate manner (Kavoosi et al., 2019).

5.4 Simulated Annealing (SA) Algorithm

The Simulated Annealing (SA) algorithm is a metaheuristic inspired by the concept of annealing in metallurgy (the technology concerned with the properties, production, and purification of metals). The algorithm was initially proposed by Kirkpatrick et al. (1983) as an adaptation of the metropolis algorithm. The metropolis algorithm is a method that simply produces a successor state y from x by transferring x into y with some random distortion (Edelkamp and Schrodl, 2012). The annealing process describes how the particles of a metal can have a higher probability of moving to a lower energy state as their temperature decreases. The concept of slow cooling in simulated annealing directly translates into a decrease in the algorithm's acceptance of poor solutions as it gets close to convergence. SA has a number of parameters that must be specified when applying it to solve problems. These parameters include: 1) initial temperature; 2) temperature intervals; and 3) acceptance probability.

```
Pseudocode 7: Simulated Annealing (SA) Algorithm
START
0: Initialization of data structures
1: Initialization of solutions for the first population
2: Computation of fitness values
If termination criterion is reached - terminate, else go to steps 3-5.
3:  Generating new solutions
4:  Computation of fitness values
5:  Selection of solutions (Boltzmann selection)
END
```

5.4.1 Steps in the proposed SA

The main steps of the proposed SA algorithm are presented in Pseudocode 7 and Figure 35. The initialization of the required data structures is done in step 0. A pool of solutions for the first population is generated in step 1 using two strategies: 1) the First Come First Served policy with Truck Sequence Considerations (FCFS-TSC) for the first half; 2) the other half is generated randomly. This initialization method is in line with some of the previously conducted studies related to the CDT truck scheduling (Yu and Egbelu, 2008; Vahdani and Zandieh, 2010; Dulebenets, 2019a).

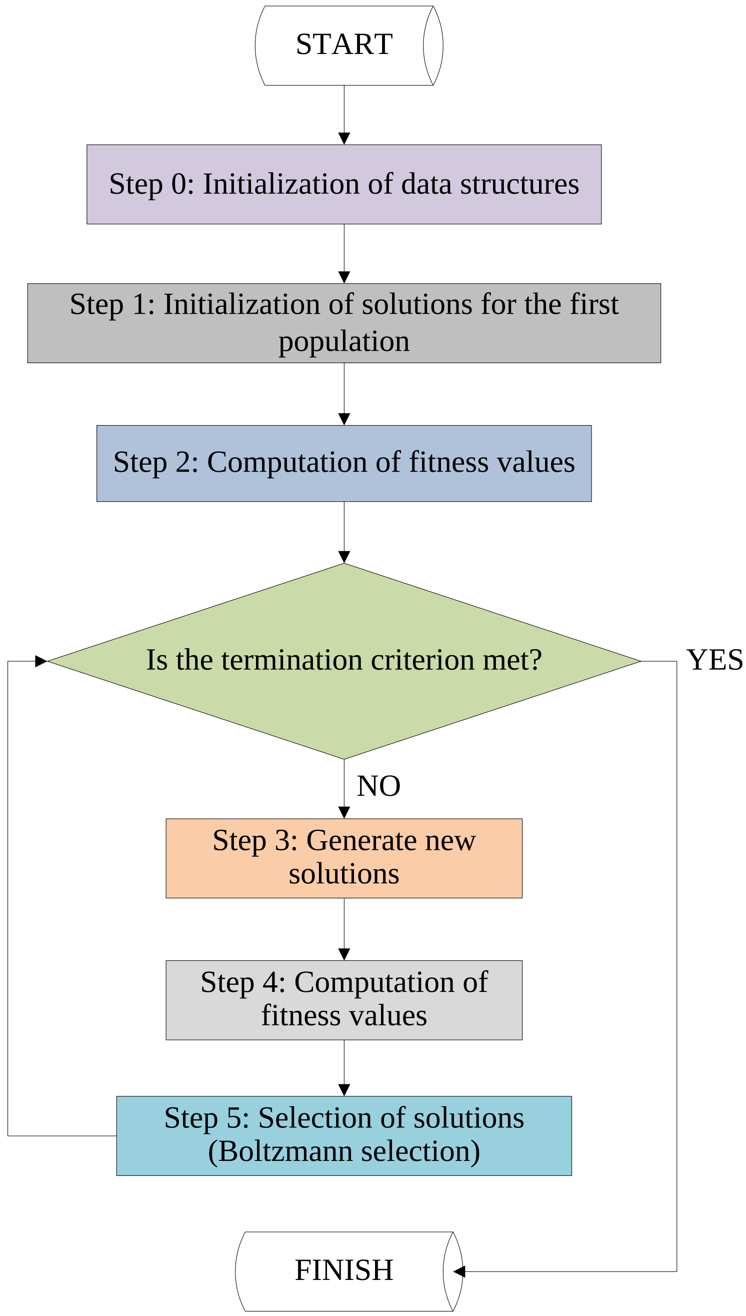


Figure 35: SA algorithm steps.

Note that the canonical SA algorithm is a single-solution-based algorithm that works only with a current solution and its neighbor. In order to improve the explorative capabilities of the algorithm, this study will use a population-based SA that works with a current solution and its neighbors (i.e., there will be a population of solutions composed of the current solution and its neighbors). In step 2, the fitness values of the generated solutions are computed. Then, the algorithm is stopped if the termination criterion is met; otherwise, it proceeds to the next step. In step 3, new solutions are generated. The fitness values of the solutions are computed in step 4. In step 5, the Boltzmann selection procedure is used to select the most promising solutions for the next iteration. Steps 3-5 are continuously executed until the termination criterion is met.

5.4.2 Representation of solutions

The solution is an integer-coded representation of the real truck scheduling information. It includes the service order of arriving trucks, and their respective assigned doors. An example of this representation is given in Figure 36. The solution represents the service order of 9 trucks, where inbound trucks "6","9", "2", and "4" are assigned to be served at door "1" (i.e., truck "6" is served first, truck "9" is served second, truck "2" is served third, and truck "4" is served fourth). Also, outbound trucks "3", "7", and "8" are assigned to be served at door "2", while outbound trucks "5" and "1" are assigned to be served at door "3". Door "1" is designated as an inbound door, while doors "2" and "3" are designated as outbound doors.

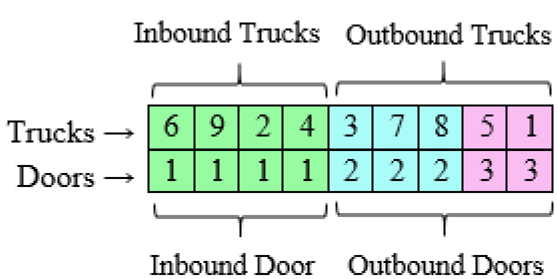


Figure 36: Solution representation for SA.

5.4.3 Initialization of solutions

The effectiveness of the algorithm directly depends on the quality of the initially generated solutions (Vahdani and Zandieh, 2010). Canonical EA-based algorithms generate their initial solutions randomly. However, randomly generated data will not be used for SA in this study due to its possibility of generating infeasible solutions within the first population of neighbors (i.e., some of the generated solutions contain the outbound trucks assigned for service before the corresponding inbound trucks). As discussed earlier, this study relies on the following two strategies to generate the pool of solutions for the first neighborhood: 1) the First Come First Served policy with Truck Sequence Considerations (FCFS-TSC) for the first half; 2) the other half is generated randomly. Such strategy is expected to ensure feasibility of truck schedules as well as their diversity within the first population of neighbors.

5.4.4 **Generation of new solutions**

The SA algorithm generates new solutions (i.e., neighbor solutions) using a procedure that modifies the initial solution and is referred to as segment inversion. A new solution is generated by selecting a random segment of the initial solution, which has trucks “3”, “1”, “9”, and “2” (see Figure 37). Then, this sequence of trucks is inverted alongside their respective doors. After that, the solution is sorted in the order of doors (see Figure 37). The application of this method, however, is based on a chosen probability, which will be determined during the parameter tuning analysis.

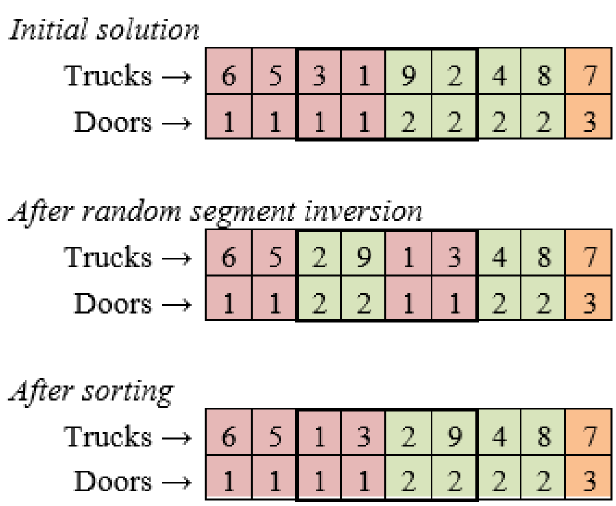


Figure 37: Generation of new solutions for the SA algorithm.

This study considers [0.01; 0.02; 0.05] as the candidate probability values of applying inversion on a given truck, which belongs to the considered solution. An example of the performance of this method is presented in Figure 37. Furthermore, the number of neighbor solutions in the SA population will be determined during the parameter tuning analysis. This study considers [30; 40; 50] as the candidate values for the SA size of population (Dulebenets, 2019a).

5.4.5 Boltzmann selection

The SA algorithm uses the Boltzmann selection mechanism to determine the more promising solutions. The probability of selecting a solution i is given (see equation 29). The equation avoids atypical temperature values due to very high or low fitness values by including a normalizing coefficient (N). The objective function of **TSP-CDT** is minimization, therefore the negative sign "-" is applied to the equation. The probability of a given solution depends on both the fitness and the temperature parameter (T). The temperature parameter (T) in any given iteration g in the algorithm is determined by the initial temperature (T^0) and the temperature interval (dT) using the expression that is represented by equation 30:

$$Pr_i = \frac{exp([-\frac{Fit_i}{N}]/T)}{\sum_r exp([-\frac{Fit_r}{N}]/T)} \; \forall i \in I \tag{29}$$

$$T = T^0 - dT(g) \tag{30}$$

The main steps of the Boltzmann selection mechanism are presented in Pseudocode 8. The required data structures for the next iteration solutions are initialized in step 0. The selection process executes iteratively (steps 2-10). In step 3, the temperature is updated for the current iteration. In step 4, a random value Val is generated using the range of 0.00 to 1.00. In step 5, the probability of solution i surviving the selection process (Pr_i) is estimated. In steps 6-8, the probability is compared with the randomly generated value Val. If $Pr_i > Val$, the solution is added to the selected solutions. This iterative process terminates when the required number of solutions has been selected. The parameter tuning analysis will be conducted to set the values of the parameters of the Boltzmann selection.

Pseudocode 8: Boltzmann Selection (BoltzmannSel)

$\boldsymbol{BoltzmannSel}(Pop, Fit, T^{o}, dT, N)$

in: Pop-population in a given iteration; Fit-fitness of solutions in a given iteration; T^{o} -the initial temperature; dT -the interval of the temperature; N-normalizing coefficient

out: $\overline{Pop}$-population in the next iteration

$|\overline{Pop}| \leftarrow \emptyset$ ⊲ Initialization

$i \leftarrow 1$

while $|\overline{Pop}| \neq |Pop|$ **do**

  $T \leftarrow T^{0} - dT(g)$ ⊲ Update the temperature for iteration g

  $Val \leftarrow \boldsymbol{Rand}(0.00; 1.00)$ ⊲ Generate a random number between 0.00 and 1.00

  $Pr_i \leftarrow \frac{exp([-\frac{Fit_i}{N}]/T)}{\sum_r exp([-\frac{Fit_r}{N}]/T)}$ ⊲ Estimate the probability of acceptance

  if $Pr_i > Val$ **then**

    $\overline{Pop} \leftarrow \overline{Pop} \cup Pop_i$ ⊲ Update the solutions

  end if

  $i \leftarrow i.(min[1, abs\{i - |Pop|\}]) + 1$

end while

return $\overline{Pop}$

5.4.6 Termination criteria

The SA algorithm developed in this study terminates once one of the following two termination criteria is met: (i) the maximum number of iterations reached; and (ii) no improvement is observed in the quality of the solutions after several iterations. Both termination criteria are popular in the literature (Dulebenets, 2019a). Furthermore, both termination criteria depend on the specifics of the considered problem (e.g., the maximum number of iterations can be larger for one problem but smaller for another – depending on the complexity of the search space). In order to enhance the performance of the algorithm, a parameter tuning analysis will be carried out to choose the termination criteria in the appropriate manner (Kavoosi et al., 2019).

CHAPTER SIX

6 NUMERICAL EXPERIMENTS

In this section, the experiments, which were conducted to evaluate the metaheuristics, are described in detail. The metaheuristic algorithms considered in this study, as described in the previous section (see section 5), include (1) Evolution Algorithm (EA); (2) Variable Neighborhood Search (VNS); (3) Tabu Search (TS); and (4) Simulated Annealing (SA). The four metaheuristics were developed using software "MATLAB" (with the version 2016a). The formulated **TSP-CDTL** model was developed using the software "General Algebraic Modeling System" (GAMS) version 24.8. Within the GAMS environment, CPLEX was selected as the mixed integer linear programming optimization solver. All the computational experiments setup was carried out on a DELL workstation that has the Microsoft Windows 10 Operating System installed, an Intel Core i1-7700k processor, and 32 GB RAM.

The scope of the computational experiments in this section can be subdivided into the following subsections: (1) parameter selection; (2) evaluation of the algorithms against the exact optimization approach; (3) detailed comparison of the metaheuristics; (4) managerial insights. In the first subsection, the parameter selection is conducted to select the most appropriate parameter values for each metaheuristic. In the second subsection, the performance of the metaheuristics will be evaluated against exact optimization using the small-size problem instances. In the third subsection, the metaheuristics will be evaluated in detail using various performance indicators. In the fourth subsection, the analysis of the managerial insights using the metaheuristic that has the best performance (after the detailed evaluation in the third subsection) will be conducted.

6.1 Parameter Selection

6.1.1 Parameter values for the proposed TSP-CDT mathematical model

The **TSP-CDT** mathematical model was developed to address the considered CDT truck scheduling problem in this study. The model's parameter range of values, as presented in Table 4, were adopted from the previous studies in the CDT truck scheduling literature as well as the studies dealing with different scheduling operations (Vahdani et al., 2010; Dulebenets, 2015a,b;

Behnamian et al., 2018; Abioye et al., 2019; Dulebenets, 2019a; Kavoosi et al., 2019; Abioye et al., 2020; Theophilus et al., 2021).

Table 4: **TSP-CDT** parameter range of values.

Parameter	Selected Value
Number of arriving trucks: $\lvert T\rvert$ (trucks)	Value varies with problem instance
Number of available doors: $\lvert D\rvert$ (doors)	Value varies with problem instance
Number of available service orders: $\lvert O\rvert$ (orders)	$\lvert O\rvert = \lvert T\rvert$
Number of available locations of storage areas: $\lvert N\rvert$ (locations of storage areas)	5
Number of product types to be handled: $\lvert P\rvert$ (product types)	10
Average inter-arrival time of trucks: $\Delta\tau^{at}$ (hours)	$\Delta\tau^{at} = E[0.083]$
Truck arrival time t: $\tau_t^{at}, t \in T$ (hours)	$\tau_{t+1}^{at} = \tau_t^{at} + \Delta\tau^{at}\ \forall t \in T$
Time of truck handling t: $\tau_t^{ht}, t \in T$ (hours)	$\tau_t^{ht} = U[0.167; 0.444]\ \forall t \in T$
Scheduled time of truck departure t: $\tau_t^{sd}, t \in T$ (hours)	$\tau_t^{sd} = \tau_t^{at} + min(\boldsymbol{\tau}_{tdo}^{tst}) \cdot U[1.000; 1.200]\ \forall t \in T$
Time of transfer from door d to location of storage n: $\tau_{dn}^{tr}, d \in D, n \in N$ (hours)	$\tau_{dn}^{tr} = U[1.250; 2.333]\ \forall d \in D, n \in N$
Temporary location of storage assignment: $z_{tn}, t \in T, n \in N$	$z_{tn} = round(U[0; 1])\ \forall t \in T, n \in N$
Product-to-truck assignment: $\varphi_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$	$\varphi_{\underline{t}\bar{t}p} = round(U[0; 1])\ \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, p \in P$
Quantity of product: $q_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ (product units)	$q_{\underline{t}\bar{t}p} = round(U[30; 40])\forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, p \in P$
Quality of product at time "0": $Q_{\underline{t}\bar{t}p}^0, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$ (%)	$Q_{\underline{t}\bar{t}p}^0 = 100\ \forall \underline{t} \in T^{in}, \bar{t} \in T^{out}, p \in P$
Rate of decay of product type p: $\lambda_p, p \in P$ ($hour^{-1}$)	$\lambda_p = U[0.010; 0.030]\ \forall p \in P$
Unit cost of waiting for truck t: $c_t^{wt}, t \in T$ (USD/hour)	$c_t^{wt} = U[100; 200]\ \forall t \in T$
Unit cost of service for truck t: $c_t^{tst}, t \in T$ (USD/hour)	$c_t^{tst} = U[150; 300]\ \forall t \in T$
Unit cost of product storage for product type p: $c_p^{ts}, p \in P$ (USD/hour)	$c_p^{ts} = U[50; 100]\ \forall p \in P$
Unit cost of delay for truck t: $c_t^{dt}, t \in T$ (USD/hour)	$c_t^{dt} = U[350; 700]\ \forall t \in T$
Unit cost of decay for product type p: $c_p^{dc}, p \in P$ (USD/% decay)	$c_p^{dc} = U[100; 150]\ \forall p \in P$
Sufficiently large positive number: M	10,000

The selection of the parameter values was done before conducting evaluation of the considered metaheuristic algorithms against the exact optimization. The arrival time of the trucks was modeled based on the exponentially distributed pseudorandom numbers with an average inter-arrival time of 5 mins (or ≈0.083 hours), represented as: $\Delta\tau^{at} = E[0.083]$. The handling time of the trucks that refers to the unloading time of the arriving inbound trucks or the loading time of the arriving outbound trucks was assumed to vary from 10 min (or ≈ 0.167 hours) to 26.667 min (or ≈ 0.444 hours), which can be represented as: $\tau_t^{ht} = U[0.167; 0.444]\ \forall t \in T$ (hours) (where "$U[Num_1; Num_2]$" corresponds to the uniformly distributed pseudorandom numbers ranging between Num_1 and Num_2). The scheduled departure time of every arriving truck was established based on its arrival time and the time required for the truck service to be completed: $\tau_t^{sd} = \tau_t^{at} + min(\boldsymbol{\tau}_{tdo}^{tst}) \cdot U[1.000; 1.200]\ \forall t \in T$ (hours). A total of 5 temporary locations for the product storage ($|N| = 5$) were assumed to be available at the considered CDT. A total of ten different product types ($|P| = 10$) were assumed to be delivered by the inbound trucks to the considered CDT.

The transfer time ($\tau_{dn}^{tr}, d \in D, n \in N$), which is the time that is needed to transfer the products that have been transported to the CDT by a given inbound truck from a given CDT door to one of the locations of storage areas (in case of inbound trucks) or the time that is needed to transfer the products from one of the locations of storage areas to a given CDT door to be loaded onto a given outbound truck (in case of outbound trucks), was set to range from 75 min (or ≈1.250 hours) to 140 min (or ≈2.333 hours). The quantity of a given product type transported by a given inbound truck through the CDT for a given outbound truck ($q_{\underline{t}\bar{t}p}, \underline{t} \in T^{in}, \bar{t} \in T^{out}, p \in P$) was assumed to vary from 30 product units to 40 product units, which can be represented as $U[30; 40]$ (product units). The quality of all the incoming products that have been transported to the CDT by the inbound trucks ($Q_{\underline{t}\bar{t}p}^0, \underline{t} \in T^{in}, \bar{t} \in T^{out}, \underline{t} \neq \bar{t}, p \in P$) was set to be 100% at time "0" (i.e., at the arrival time). The rate of decay for a given product type ($\lambda_p, p \in P$) was assumed to vary from 0.010 hour^{-1} to 0.030 hour^{-1}.

For the unit cost parameters, there are a total of 5 unit cost parameters which include: (1) the unit truck waiting cost ($c_t^{wt}, t \in T$): $U[100; 200]$ USD/hour (i.e., varied from 100 USD/hour to 200

USD/hour); (2) the unit truck service cost ($c_t^{tst}, t \in T$): $U[150; 300]$ USD/hour (i.e., varied from 150 USD/hour to 300 USD/hour); (3) the unit temporary product storage cost ($c_p^{ts}, p \in P$): $U[50; 100]$ USD/hour (i.e., varied from 50 USD/hour to 100 USD/hour); (4) the unit cost of delay in truck departure ($c_t^{dt}, t \in T$): $U[350; 700]$ USD/hour (i.e., varied from 350 USD/hour to 700 USD/hour); and (5) the unit cost of product decay ($c_p^{dc}, p \in P$): $U[100; 150]$ USD/% decay (i.e., varied from 100 USD/% decay to 150 USD/% decay). It is worthy of note that all the unit costs are estimated in USD/hour, except for the product decay cost which is estimated in USD/% decay. Based on the problem scale, this study identifies two groups of problem instances. The first group of problem instances was composed of a total of 30 small-scale problem instances, where the number of available CDT doors was changed from a minimum of 2 to a maximum of 4, and the number of arriving trucks was changed from a minimum of 6 to a maximum of 15. On the other hand, the second group of problem instances was composed of 30 large-scale problem instances, where the number of available CDT doors was changed from a minimum of 4 to a maximum of 8, and the number of arriving trucks was changed from a minimum of 84 to a maximum of 120.

6.1.2 **Parameter values for the considered metaheuristics**

As a part of this study, the parameter tuning analysis was performed for all the four metaheuristic algorithms (i.e., the EA, VNS, TS, and SA metaheuristics). The aim of the analysis was to identify the values of parameters that would enhance the performance of each solution approach (Dulebenets, 2018c-e; Dulebenets, 2019b; Jiang et al., 2019; Pasha et al., 2020a,b). The conducted analysis was similar for all the algorithms in order to identify the best combination of parameter values for each of the algorithms. This section clearly explains the parameter tuning analysis in detail.

The EA algorithm proposed in this study has six parameters which are: (1) size of population; (2) penalty to be applied for infeasible solutions; (3) size of tournament; (4) number of individuals to be chosen in each tournament; (5) probability of crossover; and (6) probability of mutation. The EA parameter values were selected based on the evaluation of different combinations of the parameters while considering the tradeoff between the CPU time and objective function value (e.g., an increase in the size of the population may further lead to some improvements in terms of the algorithmic explorative capabilities and also some improvements in terms of the values of the objective function at termination but may cause a substantial increase in terms of the CPU time). The analysis for the EA parameter tuning was performed for three randomly selected large-scale problem instances, with an assumption that each parameter has three candidate values (see Table 5). Moreover, a total of ten EA replications were launched for every combination of the parameter values to increase the accuracy of the average values of objective function and CPU time. Hence, the EA algorithm was executed for a total of (3 candidate values) $^{(6\ \text{parameters})}$ · (3 problem instances) · (10 replications for each combination) = 21,870 times in the course of the analysis. Table 5 shows the results of the EA parameter tuning analysis.

The VNS algorithm proposed in this study has three parameters which are: (1) neighborhood size; (2) penalty for infeasible solutions; and (3) probability of exchange. Note that the term "exchange probability" is defined as the probability of inverting a number of trucks that are consecutive in the current solution in order to create the neighbor solutions when performing the local search procedure. The VNS parameter values were selected based on the evaluation of different combinations of the parameters while considering the tradeoff between the CPU time and objective function value (e.g., increasing neighborhood size may improve the explorative capabilities of the algorithm and improve the values of the objective function at termination but may require a higher CPU time). Ten VNS replications were executed for each combination of parameter values to increase the accuracy of the average values of the objective function and CPU time. Therefore, the VNS algorithm was launched for a total of (3 candidate values) $^{(3\ \text{parameters})}$ · (3 problem instances) · (10 replications for each combination) = 810 times in the course of the analysis. Table 5 shows the results of the VNS parameter tuning analysis.

The proposed TS algorithm has a total of four parameters: (1) number of solutions evaluated during local search; (2) penalty for infeasible solutions; (3) probability of exchange; and (4) size of the tabu list. The selection of TS parameter values was conducted based on the evaluation of different combinations of the parameters while considering the tradeoff between the CPU time and objective function value (e.g., increasing the number of solutions evaluated during local search may enhance the explorative capabilities of the algorithm and improve the objective function values at termination but will require more CPU time). Ten TS replications were executed for each combination of parameter values to increase the accuracy of the average values of the objective function and CPU time. Therefore, the TS algorithm was launched for a total of (3 candidate values) $^{(4\text{ parameters})}$ · (3 problem instances) · (10 replications for each combination) = 2,430 times in the course of the analysis. Table 5 shows the results of the TS parameter tuning analysis.

Table 5: Selected metaheuristic parameter values.

EA	Size of population ($Psize^{EA}$)	[30; 40; 50]	50
EA	Penalty term (Ψ)	[4.00; 6.00; 8.00]	4.00
EA	Size of tournament ($Tsize$)	[8; 10; 13]	13
EA	Number of selected individuals in each tournament ($Tsel$)	[4; 5; 7]	7
EA	Probability of crossover (p^c)	[0.30; 0.50; 0.70]	0.30
EA	Probability of mutation (p^m)	[0.01; 0.02; 0.05]	0.01
VNS	Neighborhood size ($Psize^{VNS}$)	[30; 40; 50]	50
VNS	Penalty term (Ψ)	[4.00; 6.00; 8.00]	4.00
VNS	Probability of exchange (p^{ex})*	[0.01; 0.02; 0.05]	0.01
TS	Number of evaluated solutions during local search ($Psize^{TS}$)	[30; 40; 50]	40
TS	Penalty term (Ψ)	[4.00; 6.00; 8.00]	4.00
TS	Probability of exchange (p^{ex})*	[0.01; 0.02; 0.05]	0.02
TS	Tabu list size ($Tabu^{max}$) – % of $Psize^{TS}$	[0.10; 0.15; 0.20]	0.20
SA	Number of evaluated solutions during local search ($Psize^{SA}$)	[30; 40; 50]	50
SA	Penalty term (Ψ)	[4.00; 6.00; 8.00]	4.00
SA	Probability of exchange (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

*The term "exchange probability" is defined as the probability of inverting a number of trucks that are consecutive in the current solution in order to create the neighbor solutions when performing the local search procedure.

The proposed SA has a total of five parameters: (1) number of evaluated solutions during local search; (2) penalty for solutions which are infeasible; (3) probability of exchange; (4) initial temperature; (5) temperature intervals. The selection of SA parameter values was conducted based on the evaluation of different combinations of the parameters while considering the tradeoff between the CPU time and objective function value (e.g., increasing the number of solutions evaluated during local search may enhance the explorative capabilities of the algorithm and improve the objective function values at termination but will require more CPU time). Ten SA replications were executed for each combination of parameter values to increase the accuracy of the average values of the objective function and CPU time. Therefore, the SA algorithm was launched for a total of (3 candidate values)$^{(5 \text{ parameters})}$ · (3 problem instances) · (10 replications for each combination) = 7,290 times in the course of the analysis. Table 5 shows the results of the SA parameter tuning analysis.

Based on preliminary algorithmic runs for EA, 3,000 generations were set as the highest number of generations, while 2,500 generations were set as the highest number of generations without any significant improvement in the objective function. Likewise, based on preliminary algorithmic runs for VNS, TS, and SA, 3,000 iterations were set as the highest number of iterations, while the highest number of iterations without any significant improvement in the values of the objective function was set to 2,500 iterations.

6.2 Evaluation of the Algorithms against Exact Optimization

The numerical experiments conducted in this study also included an additional analysis focused on examining the solution accuracy obtained from the four metaheuristics under consideration. This section includes the following subsections: (1) the design of the linearized product decay function; and (2) detailed comparison of the solution approaches for the small-size problem instances.

6.2.1 Design of the linearized product decay function

As stated in section 4, the **TSP-CDT** is a non-linear mathematical model. Therefore, **TSP-CDTL** was formulated as the linearized version of the **TSP-CDT** model to reduce its computational complexity by the means of applying a specific linearization technique (see section 4.2). In particular, a linear static secant method of approximation was adopted to linearize the non-linear perishable decay function for each product type delivered by the inbound trucks to the considered CDT. As indicated earlier, the accuracy of linear static secant approximation increases with increasing number of linear segments. However, the CPU time required will increase as well primarily due to increasing number of variables and parameters of the **TSP-CDTL** model. Therefore, an additional analysis was conducted to determine the appropriate number of segments in the piecewise linear static secant approximation for the non-linear decay function for each perishable product type.

In this analysis, the number of linear secant segments was varied from 2 to 10, and the averages of the values of the objective function over ten replications as well as the CPU times required by CPLEX were recorded for the small-size problem instances "1" through "5". Note that the objective function values did not change from one replication to another for each of the considered small-size problem instances (as CPLEX is a deterministic solution approach). However, some fluctuations in the CPU time were observed from one replication to another for each of the considered small-size problem instances; therefore, multiple replications were performed.

The results are presented in Table 6, which comprises the following outlined data for each of the small-size problem instances considered: (i) the number of the problem instance; (ii) the number of the available CDT doors – $|D|$; (iii) the number of the trucks that arrive for service at the CDT – $|T|$; (iv) "Segments" – the number of segments in the linear static secant approximation; (v) "NumVar" – the total number of variables in the **TSP-CDTL** model; (vi) "NumCon" – the total number of constraints in the **TSP-CDTL** model; (vii) the objective function value of the solution that was returned by CPLEX ($\boldsymbol{TC}$); (viii) "Delta" – the objective gap estimated as the difference in the objective function value of the solution returned by CPLEX when a given number of segments is selected for the linear static secant approximation and the objective function value of the solution returned by CPLEX when the maximum number of segments (i.e., 10 segments) is selected for the linear static secant approximation; and (ix) the CPU time that was incurred by CPLEX when a given number of segments is selected for the linear static secant approximation.

Table 6: Design of the linearized product decay function.

Instance	$\|D\|$	$\|T\|$	Segments	NumVar	NumCon	$\boldsymbol{TC}$ (USD)	Delta	CPU time, sec
1	2	6	2	2675	4077	21182.41	19.05%	4.33
1	2	6	3	3395	5157	20753.65	16.64%	4.06
1	2	6	4	4115	6237	20330.25	14.26%	5.03
1	2	6	5	4835	7317	19904.09	11.87%	5.24
1	2	6	6	5555	8397	19476.97	9.47%	5.31
1	2	6	7	6275	9477	19052.28	7.08%	5.14
1	2	6	8	6995	10557	18626.24	4.68%	5.34
1	2	6	9	7715	11637	18199.52	2.29%	7.83
1	2	6	10	8435	12717	17792.83	0.00%	17.91
2	2	7	2	3651	5581	25220.13	16.27%	73.13
2	2	7	3	4631	7051	24771.29	14.20%	78.36
2	2	7	4	5611	8521	24328.07	12.15%	79.41
2	2	7	5	6591	9991	23881.95	10.10%	78.64
2	2	7	6	7571	11461	23434.84	8.03%	80.47
2	2	7	7	8551	12931	22990.26	5.99%	85.39
2	2	7	8	9531	14401	22544.27	3.93%	86.41
2	2	7	9	10511	15871	22097.58	1.87%	89.39
2	2	7	10	11491	17341	21691.91	0.00%	124.58
3	2	8	2	4731	7255	28470.56	13.20%	290.77
3	2	8	3	6011	9175	28038.93	11.49%	292.78
3	2	8	4	7291	11095	27612.71	9.79%	295.88
3	2	8	5	8571	13015	27183.70	8.09%	310.67
3	2	8	6	9851	14935	26753.73	6.38%	314.70
3	2	8	7	11131	16855	26326.21	4.68%	325.33
3	2	8	8	12411	18775	25924.28	3.08%	341.17
3	2	8	9	13691	20695	25537.53	1.54%	348.22
3	2	8	10	14971	22615	25150.08	0.00%	349.70
4	2	9	2	6005	9223	37751.42	11.11%	3600.02
4	2	9	3	7625	11653	37256.02	9.65%	3600.05
4	2	9	4	9245	14083	36766.83	8.21%	3600.05
4	2	9	5	10865	16513	36274.44	6.76%	3600.05
4	2	9	6	12485	18943	35780.94	5.31%	3600.05
4	2	9	7	14105	21373	35309.46	3.92%	3600.08
4	2	9	8	15725	23803	34865.68	2.62%	3600.06
4	2	9	9	17345	26233	34421.79	1.31%	3600.08
4	2	9	10	18965	28663	33977.10	0.00%	3600.08
5	2	10	2	7371	11345	38382.96	9.35%	3600.03
5	2	10	3	9371	14345	37943.22	8.10%	3600.06
5	2	10	4	11371	17345	37508.98	6.86%	3600.06
5	2	10	5	13371	20345	37071.90	5.62%	3600.08
5	2	10	6	15371	23345	36677.71	4.49%	3600.06
5	2	10	7	17371	26345	36282.82	3.37%	3600.09
5	2	10	8	19371	29345	35888.91	2.25%	3600.08
5	2	10	9	21371	32345	35494.88	1.12%	3600.08
5	2	10	10	23371	35345	35100.15	0.00%	3600.08

As expected, the number of variables and constraints in the **TSP-CDTL** model substantially increases with increasing number of segments in the linear static secant approximation, which further caused an increase in the CPU time required by CPLEX. However, the CPU time increase can be viewed as acceptable for the small-size problem instances “1”, “2”, and “3” (e.g., the CPU time increased from 290.77 sec to 349.70 sec for the small-size problem instance “3” after increasing the number of segments in the linear static secant approximation from 2 segments to 10 segments).

The small-size problem instances “4” and “5” could not be solved with CPLEX within the computational time limit imposed (i.e., 3,600 sec with the relative optimality gap of 1.00%) even when 2 segments were used in the linear static secant approximation for each perishable product type. The latter highlights the computational complexity of the **TSP-CDTL** model. Furthermore, the average values of the objective function were more promising when more segments were used in the linear static secant approximation. Based on the conducted analysis, the linear static secant approximation with 10 segments will be further used within CPLEX for each perishable product type throughout the computational experiments.

6.2.2 Detailed comparison of the solution approaches for the small-size problem instances

The accuracy of the solution approaches (i.e., EA, VNS, TA, and SA metaheuristic algorithms) was further analyzed by comparison with CPLEX. A total of 4 replications were performed for each problem instance, and the average CPU times were estimated. The objective function values remained the same for all the replications of each problem instance. Throughout the numerical experiments, the CPU time restriction imposed on CPLEX was 3,600 sec (i.e., 1 hour), while the relative optimality gap was set to 1.00%. In particular, CPLEX was stopped if one of the following criteria has been met: (1) the CPU time reached 3,600 sec or more; and (2) the relative optimality gap reached 1.00% or less.

Table 7 shows the outputs of the conducted comparative analysis of the metaheuristics considered, which comprise the following outlined data for each of the small-size problem instances: (i) the number of the problem instance; (ii) the number of the available CDT doors – $|D|$; (iii) the number of the trucks that arrive for service at the CDT – $|T|$; (iv) the objective function value of the solution that was returned by CPLEX; (v) the CPU time that was incurred by CPLEX; (vi) the values of objective function of the solutions that were returned by each metaheuristic algorithm (average of ten replications); and (vii) the values of the CPU time that was incurred by each metaheuristic algorithm (average of ten replications). The analysis showed that the size of the problem instances after applying the linearization technique on the original

TSP-CDT mathematical model still significantly affected the CPU time of CPLEX. It can be observed that, for the problem instances with 9 trucks or more (i.e., the problem instances "4" – "10"; "14" – "20"; and "24" – "30"), CPLEX could not solve the **TSP-CDT** model to global optimality within the time limit of 3,600 sec (i.e., 1 hour) which was set as the CPU time constraint. The latter finding shows that the CDT truck scheduling problem has a high computational complexity and, hence, requires the deployment of metaheuristics.

For the problem instances "1", "2", "3", "11", "12", "13", "21", "22", and "23", the considered metaheuristics obtained optimal solutions which were identical to those obtained by CPLEX. In addition, the EA, VNS, TS, and SA metaheuristics required significantly lower CPU times with an average of 18.02 sec, 14.95 sec, 12.85 sec, and 16.13 sec, respectively, over the small-size problem instances that were generated; while CPLEX required a CPU time of 2,757.92 sec on average. The EA, VNS, TS, and SA metaheuristics showed the average objective function values of 35,599.6 USD, 36,101.3 USD, 35,786.1 USD, and 36,373.3 USD, respectively; while the average objective function value obtained by CPLEX was 37,446.5 USD (see Table 7). Therefore, the considered metaheuristics yielded solutions which had an improved quality for the majority of the small-size problem instances that were generated when comparing to CPLEX.

This numerical analysis step included the computation of the optimality gaps for the considered metaheuristics, which was computed using the following relationship: $Gap_{alg} = \frac{TC_{alg} - TC_{cplex}}{TC_{cplex}}$, where TC_{alg} is the objective function value obtained by algorithm alg; TC_{cplex} is the objective function value obtained by CPLEX. The average optimality gaps for EA, VNS, TS and SA were -3.35%, -2.51%, -3.05% and -1.96%, respectively. Hence, this finding confirms the accuracy of the considered metaheuristics in terms of solution quality and their superior performance as compared to CPLEX. Since the analyzed problem instances can be classified as small-size (i.e., the number of CDT doors did not surpass 4, and the number of trucks did not surpass 15), the difference among the values of the objective function returned by the four metaheuristics was not significant.

Table 7: Comparative analysis of CPLEX vs. metaheuristic algorithms for small-size problem instances.

Instance	$\lvert D \rvert$	$\lvert T \rvert$	CPLEX		EA		VNS		TS		SA	
			$\boldsymbol{TC}$ (USD)	CPU (sec)	$\boldsymbol{TC}$ (USD)	CPU (sec)	$\boldsymbol{TC}$ (USD)	CPU (sec)	$\boldsymbol{TC}$ (USD)	CPU (sec)	$\boldsymbol{TC}$ (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

*The bold font is adopted for the best values of the objective function that were identified.

6.3 Detailed Comparison of the Metaheuristics

The main objective of this section in the numerical experiments is the detailed evaluation of the metaheuristics for the realistic-size problem instances. The algorithms were compared in terms of important performance indicators, such as the objective function values at termination, CPU times, and the stability of the algorithms. The numerical analysis comprised of a total of 10 algorithmic runs using the realistic-size problem instances, and the average values of the objective function and the CPU time were estimated. This section includes the following subsections: (1) the values of objective function and CPU time; (2) the variations in values of objective function and CPU time; and (3) the patterns of convergence of the algorithms. First, the values of the objective function and the CPU time were evaluated and discussed for all the realistic-size problem instances. Second, the stability of the metaheuristics at termination was evaluated due to the stochastic nature of the algorithmic runs (i.e., the final solution may vary from one replication to the other for each of the metaheuristics considered). Third, the patterns of convergence were investigated in detail for all the metaheuristics. The next subsections elaborate on these analyses in detail.

6.3.1 The values of the objective function and CPU time

The first step in the evaluation of the computational performance is the analysis of the values of the objective function and CPU time for the considered metaheuristics. It is generally expected that good-quality solutions (i.e., solutions that have acceptable values of the objective function and an acceptable CPU time) are returned by an efficient metaheuristic algorithm (unlike CPLEX that requires high CPU time to produce good-quality or even optimal solutions – see Table 7). Since a CDT manager ideally desires a truck schedule that minimizes the total cost ($\boldsymbol{TC}$), a solution approach that requires a high CPU time to return a good-quality solution cannot be used. Therefore, each metaheuristic algorithm is evaluated based on the tradeoff between the values of the objective function and the CPU time. Table 8 shows the comparative analysis outputs for the metaheuristics considered in terms of the values of the objective function as well as the CPU time, which comprise of the following data for each of the large-size problem instances: (i) the number of the problem instance; (ii) the number of the available CDT doors – $|D|$; (iii) the number of the trucks that arrive for service at the CDT – $|T|$; (iv) the value of the objective function of the truck schedule that was returned by each metaheuristic (average of ten replications); (v) the value of the CPU time that was incurred by each metaheuristic (average of ten replications).

Table 8: Comparative analysis of the metaheuristic algorithms for realistic-size problem instances.

Instance	$\|D\|$	$\|T\|$	EA		VNS		TS		SA	
			$\boldsymbol{TC}$ (10^3 USD)	CPU (sec)	$\boldsymbol{TC}$ (10^3 USD)	CPU (sec)	$\boldsymbol{TC}$ (10^3 USD)	CPU (sec)	$\boldsymbol{TC}$ (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

*The bold font is adopted for the best values of the objective function that were identified.

The results of the analysis show that the average values of the objective function were 1,657.16 10^3 USD, 2,358.26 10^3 USD, 1,821.03 10^3 USD, and 3,220.59 10^3 USD for the EA, VNS, TS, and SA metaheuristics, respectively. In particular, the average objective function value of EA was lower than VNS, TS, and SA by 37.71%, 10.99% and 92.65%, respectively. The latter finding shows that the EA metaheuristic, which is a population-based metaheuristic, is capable of exploring more promising domains and obtaining solutions with good quality within the solution search space. The other metaheuristics (i.e., VNS, TS, and SA) are single-solution-based metaheuristics that utilize local search methods, which are limited in their exploration of promising domains and obtaining solutions with good quality within the solution search space. The analysis also included the assessment of the substantial difference among the average values of objective function obtained by EA and each of the other metaheuristics (i.e., VNS, TS, and SA). The paired z-test was used for the assessment. In particular, each of the alternative metaheuristics was paired against EA for the paired z-test for each large-size problem instance: (1) "EA vs. VNS"; (2) "EA vs. TS"; and (3) "EA vs. SA".

Based on the null hypothesis (H_0), it was presumed that there was no any statistically significant difference between the average of the objective function values obtained by EA and those obtained by the alternative metaheuristics. On the other hand, based on the alternative hypothesis (H_1), it was presumed that there was a statistically significant difference between the average of the objective function values obtained by EA and those obtained by the alternative metaheuristics. The z-statistic was calculated for each paired z-test based on the following relationship: $z = \frac{mean(\boldsymbol{TC}_m) - mean(\boldsymbol{TC}_{EA})}{\sqrt{\frac{(std(\boldsymbol{TC}_m))^2}{n} + \frac{(std(\boldsymbol{TC}_{EA}))^2}{n}}}$, where $mean(\boldsymbol{TC}_m)$ is the average of the values of the objective function that were returned by a given metaheuristic m; $mean(\boldsymbol{TC}_{EA})$ is the average of the values of the objective function that were returned by EA; $std(\boldsymbol{TC}_m)$ is the standard deviation of the values of the objective function for a given metaheuristic m; $std(\boldsymbol{TC}_{EA})$ is the standard deviation of the values of the objective function that were returned by EA; and n is the number of observations (=2 as two metaheuristics that have ten replications each were evaluated for each z-test).

Table 9 shows the analysis outputs for the conducted paired z-tests, which comprise of the following data for each of the large-size problem instances: (i) the number of the problem instance; (ii) the number of the available CDT doors – $|D|$; (iii) the number of the trucks that arrive for service at the CDT – $|T|$; (iv) the value of the z-statistic for the z-test "EA vs. VNS"; (v) the value of the z-statistic for the z-test "EA vs. TS"; and (vi) the value of the z-statistic for the z-test "EA vs. SA".

Table 9: The results of the conducted paired z-test.

Instance	$\|D\|$	$\|T\|$	EA vs. VNS	EA vs. TS	EA vs. SA
31	4	84	27.38	14.70	14.95
32	4	88	12.64	12.75	31.17
33	4	92	40.91	9.83	20.02
34	4	96	65.22	11.57	30.16
35	4	100	39.15	16.61	39.39
36	4	104	30.51	3.33	26.81
37	4	108	30.93	4.61	36.11
38	4	112	63.01	1.30	45.56
39	4	116	54.06	1.53	38.00
40	4	120	39.03	1.83	59.44
41	6	84	12.24	7.36	28.79
42	6	88	20.89	7.49	17.42
43	6	92	16.23	6.60	18.79
44	6	96	44.75	6.51	20.47
45	6	100	31.24	6.65	32.57
46	6	104	33.52	0.56	84.26
47	6	108	21.06	1.76	42.66
48	6	112	40.62	0.86	67.54
49	6	116	29.87	4.73	85.28
50	6	120	48.14	0.10	55.22
51	8	84	15.49	9.52	30.01
52	8	88	24.87	11.41	24.26
53	8	92	40.66	7.59	31.66
54	8	96	16.00	7.78	62.52
55	8	100	43.09	8.01	37.27
56	8	104	17.74	0.05	88.85
57	8	108	6.89	-2.74	62.05
58	8	112	14.04	-0.39	92.78
59	8	116	23.32	0.64	92.01
60	8	120	14.93	-1.63	66.24
Average:			**30.61**	**5.36**	**46.08**

Based on the patterns identified in Table 9, the performed paired z-tests analysis shows that the average z-statistic values over all the realistic-size problem instances that were generated comprised 30.61, 5.36, and 46.08 for the "EA vs. VNS", "EA vs. TS", and "EA vs. SA" tests, respectively. Thus, the null hypothesis which assumed no statistically significant difference can be rejected at 0.01% as the level of significance and 3.819 as the critical z-value. The latter finding demonstrates that the objective function values obtained by the EA metaheuristic are statistically superior to the values obtained by the alternative metaheuristic algorithms considered (i.e., VNS, TS, and SA).

Regarding the average CPU time values required, the EA, VNS, TS, and SA metaheuristics incurred the following CPU times: 132.78 sec, 134.37 sec, 103.02 sec, and 132.70 sec, respectively, over the realistic-size problem instances that were generated. This can be explained by the number of additional computations, such as crossover operation, that EA performs which require additional CPU time. The maximum CPU time required for the EA metaheuristic was not greater than 180 sec (or 3 min) which is a reasonable time to produce a good-quality solution for a CDT manager. Note that the realistic-size problem instances evaluated in this section had CDT doors which ranged from a minimum of 4 to a maximum of 8 doors, and the arriving trucks ranged from a minimum of 84 to a maximum of 120 trucks. Therefore, fairly large-size instances from the practical perspective were captured in the conducted analysis.

6.3.2 Objective function and CPU time variations

The considered metaheuristics deploy a variety of operators that are stochastic in nature in order to obtain good-quality solutions (e.g., the developed EA deploys a crossover operator to produce the chromosomes of the offspring and a mutation operator to mutate the chromosomes). The main aim of these stochastic operators is to allow the metaheuristics change from current solutions in order to find superior solutions. However, the application of these operators produces variations in the values of the objective function at termination. Furthermore, the CPU time that is required for the execution of the same metaheuristic on the same problem instance for a given CPU may vary from one replication to the other. Although these variations could also be as a result of additional system processes that are running concurrently on a given CPU during the execution of the metaheuristic (e.g., scheduled updates, system scans). Generally, metaheuristics which return solutions that have a significant variation are considered as unreliable for realistic applications. Therefore, in addition to the conducted analyses in these numerical experiments, the variations in the values of the objective function and the CPU time were also analyzed.

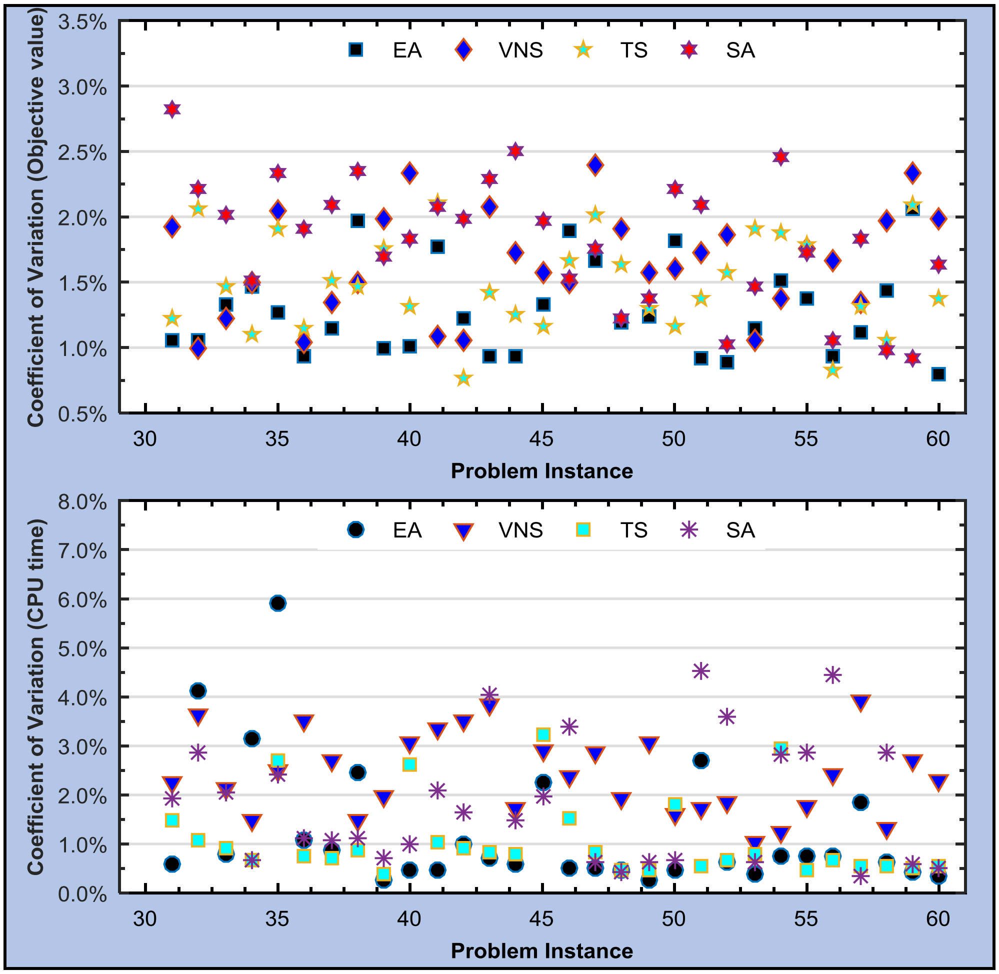


Figure 38: The coefficient of variation in the values of the objective function and CPU time for the considered algorithms.

The study adopts the coefficient of variation metric as a measure of the variation in values of the objective function and the CPU time. The coefficients of variation were then computed for each of the considered metaheuristics and each realistic-size problem instance over the ten replications that were performed. Figure 38 presents the results of the coefficients of variation analysis that was conducted. The results showed that the coefficient of variation of the values of the objective function obtained by the metaheuristics was not greater than 2.82% for the realistic-size problem instances that were generated. Also, the coefficient of variation of the CPU time that was incurred by the metaheuristics was less than 5.91% for the realistic-size problem instances that were generated. Therefore, the stability of the metaheuristics (both EA and the alternative metaheuristics) is confirmed for both objective function and the CPU time values. The EA metaheuristic typically had the lowest coefficients of variation of the objective function for the realistic-size problem instances that were generated, followed by TS. On the other hand, the SA metaheuristic typically had the highest coefficients of variation of the objective function for the realistic-size problem instances that were generated.

6.3.3 Convergence patterns of the algorithms

The convergence pattern analysis is another critical step in the evaluation of the efficiency of metaheuristics developed. The convergence patterns of the considered metaheuristics are presented in this section (see Figure 39 for details). The patterns of convergence for the algorithms showed that each metaheuristic was able to generate improved solutions from one generation (in case of EA) or iteration (in case of VNS, TS, and SA) to the other. This pattern can be used to compare the performance of metaheuristics in terms of how quickly an improved solution is obtained. In this analysis, the patterns of convergence were obtained from the algorithmic runs selected from the realistic-size problem instances (specifically, problem instances “51” – “60”) and are presented in Figure 39.

The problem instances that were selected represent a portion of the realistic-size problem instances with the highest number of CDT doors and arriving trucks both in- and outbound. It can be observed from the convergence patterns that the algorithms begin their exploration from the same chromosome (or solution). This is due to the fact that FCFS-TSC heuristic was used for the chromosome/solution initialization phase of the considered metaheuristic algorithms. The convergence patterns show that within the number of generations/iterations set as termination criteria for the algorithms, EA and TS achieved solutions of good quality more quickly than VNS and SA. It can be also observed that the SA metaheuristic had the worst convergence patterns (i.e., it converged quickly in one of the local optima and was not able to discover any other promising search space domains). For the problem instances “57”, “58”, and “60”, TS was able to achieve improved solutions faster than EA. However, for most of the evaluated realistic-size problem instances, the EA metaheuristic outperformed the VNS, TS, and SA metaheuristics.

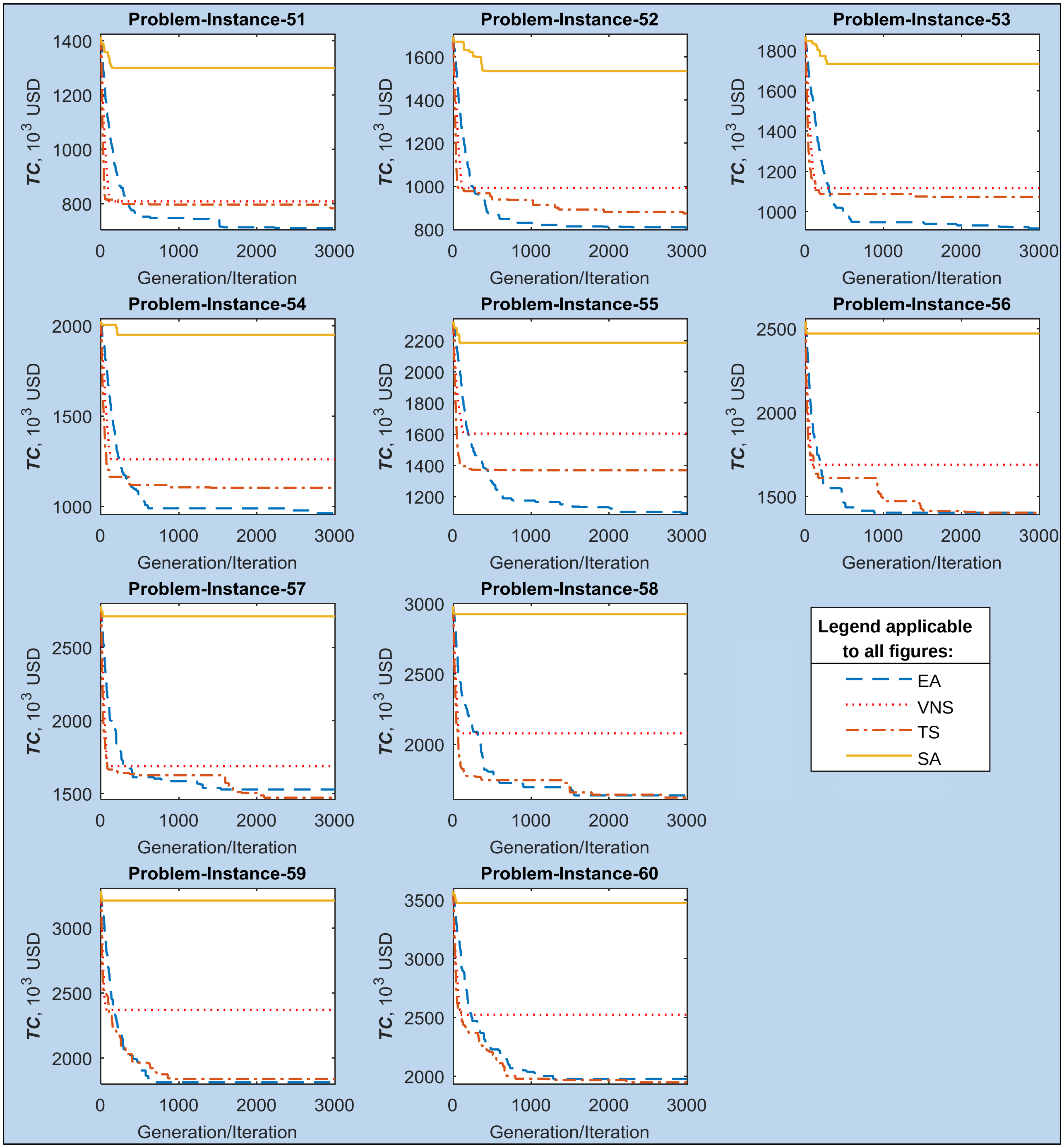


Figure 39: Convergence patterns of the algorithms (problem instances “51” – “60”).

Moreover, the use of the tabu list data structure in the TS algorithm provided a comparative advantage in the algorithmic search process, since the supplementary restrictions were forced on the algorithm for revisiting exactly the same solutions. These restrictions allowed the TS algorithm to discover new search space domains that were more promising rather than evaluating solutions that were of similar quality near the local optima. In particular, TS demonstrated a competitive performance against EA for a number of realistic-size problem instances (i.e., especially for the instances "56" through "60"). Based on the overall convergence pattern analysis results presented in Figure 39, the EA metaheuristic performed better than all the other metaheuristic algorithms (VNS, TS, and SA) by discovering superior solutions faster in at least 7 out of 10 times. The latter finding proves the superiority of the EA algorithm for the considered **TSP-CDT** mathematical model.

6.4 Managerial Insights

This section comprises of the set of sensitivity analyses that were performed using the proposed **TSP-CDT** mathematical model and the superior metaheuristic (i.e., the EA metaheuristic). The previous sections (see sections 6.3.1 – 6.3.3) of the numerical analysis contain the comparative analyses of the considered metaheuristics. It was observed that the EA metaheuristic offered results with the best balance between the solution quality at convergence and the incurred CPU time. Therefore, all subsequent sensitivity analyses will be conducted using the developed EA metaheuristic, and important managerial insights will be drawn. These would provide the CDT manager with valuable assistance in making decisions regarding the CDT operations.

Table 10: Sensitivity analysis ranges.

Sensitivity analysis	Scenario range
Product decay rate	$U[0.010; 0.030] - U[0.052; 0.155]$
Unit truck waiting cost	$U[68.30;\ 136.60] - U[161.05;\ 322.10]$
Unit truck service cost	$U[102.45;\ 204.90] - U[241.57;\ 483.14]$
Unit cost of product storage	$U[25.66;\ 51.32] - U[60.50;\ 121.00]$
Delayed truck departure cost	$U[179.61;\ 359.22] - U[423.50;\ 847.00]$

Specifically, the sensitivity of the CDT truck scheduling decisions to the following attributes will be analyzed: (1) product decay rate; (2) the unit truck waiting cost; (3) the unit truck service cost; (4) the unit cost of product storage; (5) unit delayed truck departure cost; and (6) truck arrival patterns and the CDT door capacity. The ranges of the product decay rate, unit truck waiting cost, unit truck service cost, unit cost of product storage, and unit delayed truck departure cost that were used in the analyses are reported in Table 10. Sections 6.4.1 – 6.4.6 of the dissertation discuss the details regarding the managerial implications using the developed EA algorithm.

6.4.1 **Assessment of the impact of the rate of product decay on the scheduling of trucks**

Ten scenarios of the product decay rate were developed. These scenarios were generated by increasing the value of the base product decay rate, based on uniform distribution $U[0.010; 0.030]$. Each successive scenario was set 20% greater than the previous scenario (i.e., the decay rate of products was set as $U[0.010; 0.030]$ in scenario "1", while the decay rate of products was set as $[0.052; 0.155]$ in scenario "10"). The largest among the realistic-size problem instances was used for the sensitivity analysis. Note that the problem instance "60", which is the largest one, has a total of 8 CDT doors (i.e., inbound and outbound doors) and 120 arriving trucks (i.e., trucks both in- and outbound).

The other **TSP-CDT** mathematical model parameters were assumed to remain constant, as specified in section 6.1.1, while the rate of product decay was varied. The results of the sensitivity analysis that was conducted are presented in Figure 40 (A-G) and Figure 41. These results show the changes in waiting time of trucks, truck delayed departure time, total storage time, product decay, service time of trucks both in- and outbound (denoted as "Service Time - 1" for the total service time of the inbound trucks and "Service Time - 2" for the total service time of the outbound trucks), individual cost components of the objective function, and product decay cost.

In order to account for probable stochasticity in the results, a total of 10 algorithmic replications of each scenario were run, and the average values were recorded. The observation of the results shows that as the product decay rate increased, it caused a reduction in the waiting time of the arriving trucks and the total delayed departure time. Therefore, the arriving outbound trucks were aimed to be loaded in a timely fashion with the perishable products to reduce the overall total decay, so that the shelf life of perishable products could be preserved and these products could be then delivered to the designated end customers on time.

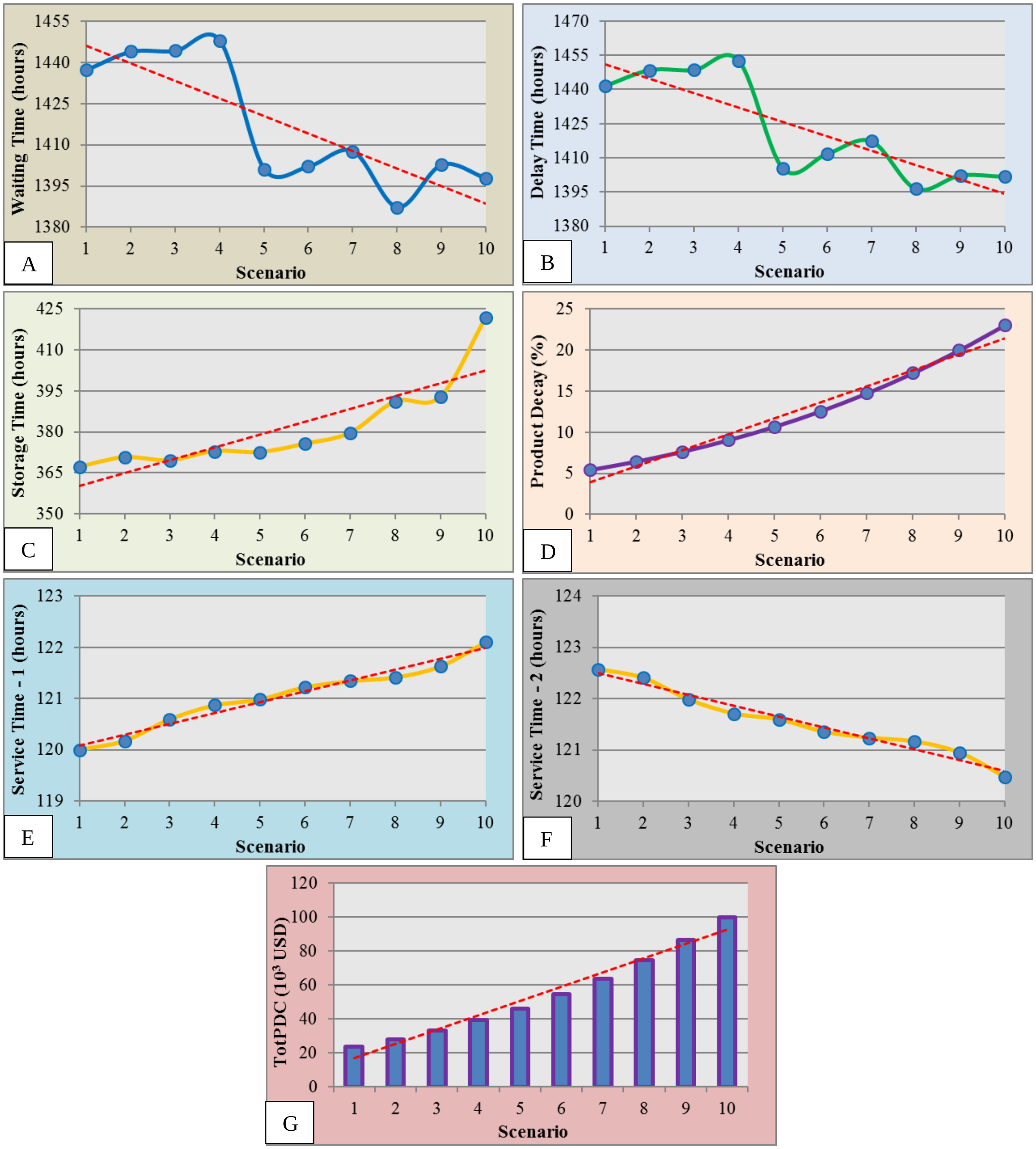
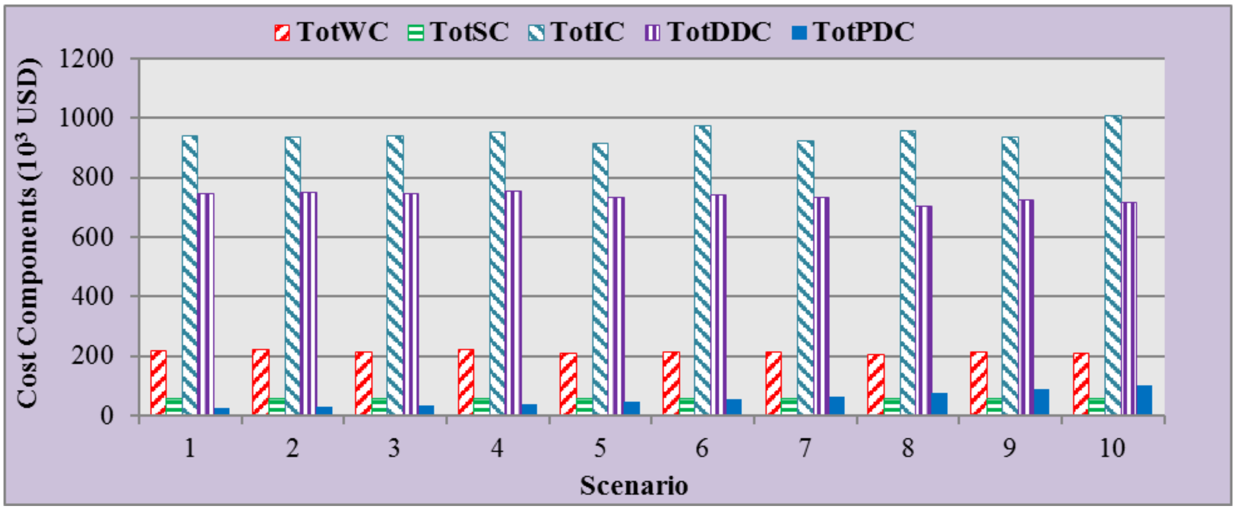


Figure 40 (A-G): Sensitivity analysis results for the product decay rate.

Although in some cases, the perishable products had to be transferred to the temporary locations of storage areas because of the variations in terms of the service start times of the arriving inbound trucks delivering the products and the arriving outbound trucks receiving the products. The latter further caused an increase in the total storage time of the delivered products. However, since this dissertation models temperature-controlled locations of storage areas for perishable products, the decay throughout the storage of products could be considered as insignificant. Throughout the numerical experiments, it was noticed that the total cost of perishable product decay increased with increasing product decay rate.



Note: TotWC - total cost of waiting; TotSC - total cost of service; TotIC - total cost of product storage; TotDDC - total cost of delay on the departure of trucks; TotPDC - total cost of perishable product decay

Figure 41: Cost components for the product decay rate sensitivity analysis.

In order to decrease the waiting time, the arriving inbound trucks were assigned for service at the available CDT doors which were not necessarily closer to the assigned temporary locations of storage areas. The latter action caused an increase in the service time at the CDT inbound doors. On the other hand, the arriving outbound trucks were typically assigned for service at the available CDT doors which were closer to the assigned temporary locations of storage areas (hence, a reduction in the service time at the CDT outbound doors was observed). Moreover, it was observed that the decay rate increased from 2.25% to 11.62% on average for the considered product types. Furthermore, the total cost (i.e., $\boldsymbol{TC}$ - the **TSP-CDT** objective function value), associated with the service of the trucks arriving at the CDT, was significantly affected by an increase in the product decay rate. In particular, the total cost of serving trucks increased from 1,972.21 ($\times 10^3$ USD) in scenario "1" to 2,085.51 ($\times 10^3$ USD) in scenario "10".

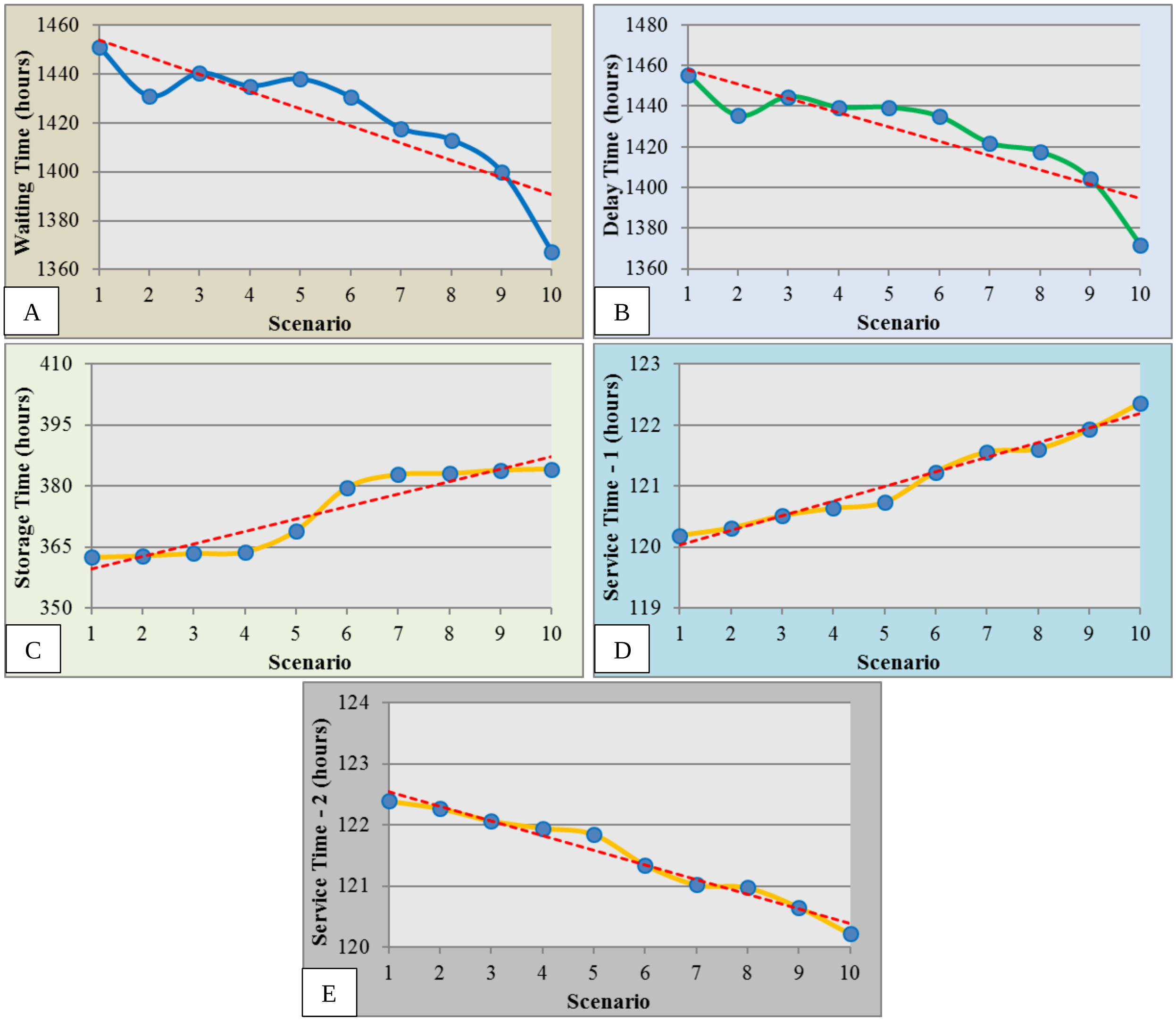
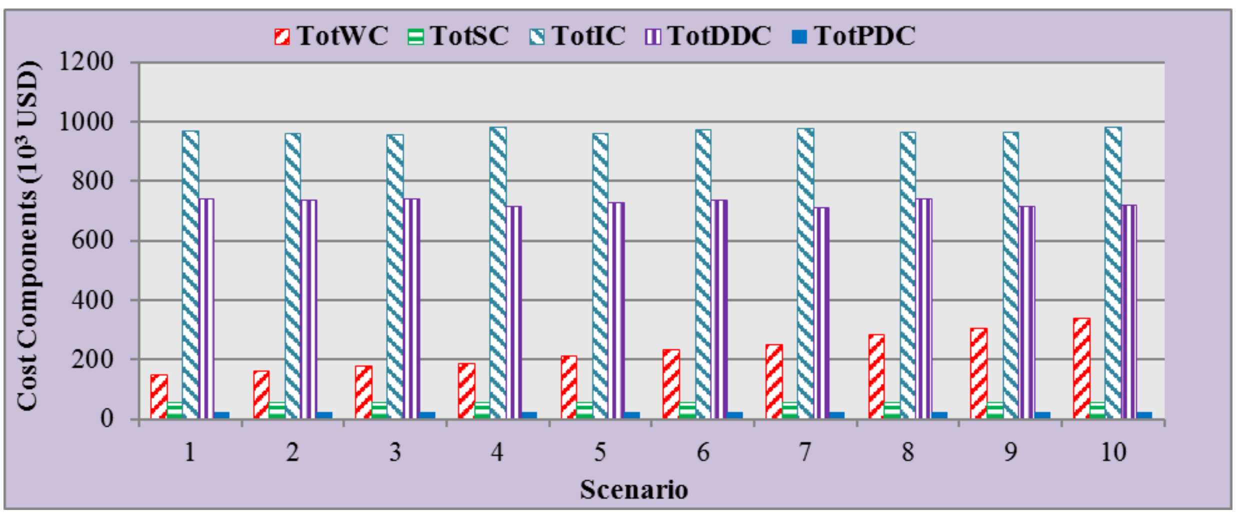


Figure 42 (A-E): Sensitivity analysis for the unit waiting cost of trucks.

6.4.2 Assessment of the impact of the unit waiting cost of trucks on the scheduling of trucks

Ten scenarios of the unit waiting cost of trucks were developed. These scenarios were generated by increasing the value of the base unit waiting cost, based on uniform distribution $U[68.30;\ 136.60]$. Each successive scenario was set 10% greater than the previous scenario (i.e., the unit waiting cost of trucks was set as $U[68.30;\ 136.60]$ in scenario "1", while the unit waiting cost of trucks was set as $U[161.05;\ 322.10]$ in scenario "10"). The largest among the realistic-size problem instances was used for the sensitivity analysis. Note that the problem instance "60", which is the largest one, has a total of 8 CDT doors (i.e., inbound and outbound doors) and 120 arriving trucks (i.e., trucks both in- and outbound). The other **TSP-CDT** mathematical model parameters were assumed to remain constant, as specified in section 6.1.1, while the unit waiting cost of trucks was varied.

The results of the sensitivity analysis that was conducted are presented in Figure 42 (A-E) and Figure 43. These results show the changes in waiting time of trucks, truck delayed departure time, total storage time, service time of trucks both in- and outbound (denoted as "Service Time - 1" for the total service time of the inbound trucks and "Service Time - 2" for the total service time of the outbound trucks), and individual cost components of the objective function. In order to account for probable stochasticity in the results, a total of 10 algorithmic replications of each scenario were run, and the average values were recorded.



Note: TotWC - total cost of waiting; TotSC - total cost of service; TotIC - total cost of product storage; TotDDC - total cost of delay on the departure of trucks; TotPDC - total cost of perishable product decay

Figure 43: Cost components for the unit truck waiting cost sensitivity analysis.

It can be observed that as the unit waiting cost of trucks increased, the waiting time and the delayed departure time reduced. The latter can be explained by the fact that an increase in the unit waiting cost could prohibitively increase the total cost of serving the arriving trucks; therefore, the arriving trucks were assigned to the first available CDT doors (especially, inbound doors) to reduce the waiting time. The latter generally caused the storage time to increase, as the first available CDT doors were located farther away from the designated temporary locations of storage areas for the delivered products. Such finding also explains the fact that the total service time for the inbound trucks generally increased with increasing unit waiting cost of trucks. Moreover, it was observed that the total waiting time of trucks decreased from 1,451.1 hours in scenario “1” to 1,367.3 hours in scenario “10”. Furthermore, the total cost (i.e., $\boldsymbol{TC}$ - the **TSP-CDT** objective function value), associated with the service of the trucks arriving at the CDT, was significantly affected by an increase in the unit waiting cost of trucks. In particular, the total cost of serving trucks increased from 1,937.83 ($\times 10^3$ USD) in scenario “1” to 2,116.57 ($\times 10^3$ USD) in scenario “10”.

6.4.3 Assessment of the impact of the unit service cost of trucks on the scheduling of trucks

Ten scenarios of the unit service cost of trucks were developed. These scenarios were generated by increasing the value of the base unit service cost, based on uniform distribution $U[102.45;\ 204.90]$. Each successive scenario was set 10% greater than the previous scenario (i.e., the unit service cost of trucks was set as $U[102.45;\ 204.90]$ in scenario “1”, while the unit service cost of trucks was set as $U[241.57;\ 483.14]$ in scenario “10”). The largest among the realistic-size problem instances was used for the sensitivity analysis. Note that the problem instance “60”, which is the largest one, has a total of 8 CDT doors (i.e., inbound and outbound doors) and 120 arriving trucks (i.e., trucks both in- and outbound). The other **TSP-CDT** mathematical model parameters were assumed to remain constant, as specified in section 6.1.1, while the unit service cost of trucks was varied.

The results of the sensitivity analysis that was conducted are presented in Figure 44 (A-E) and Figure 45. These results show the changes in waiting time of trucks, truck delayed departure time, total storage time, service time of trucks both in- and outbound (denoted as “Service Time - 1” for the total service time of the inbound trucks and “Service Time - 2” for the total service time of the outbound trucks), and individual cost components of the objective function. In order to account for probable stochasticity in the results, a total of 10 algorithmic replications of each scenario were run, and the average values were recorded.

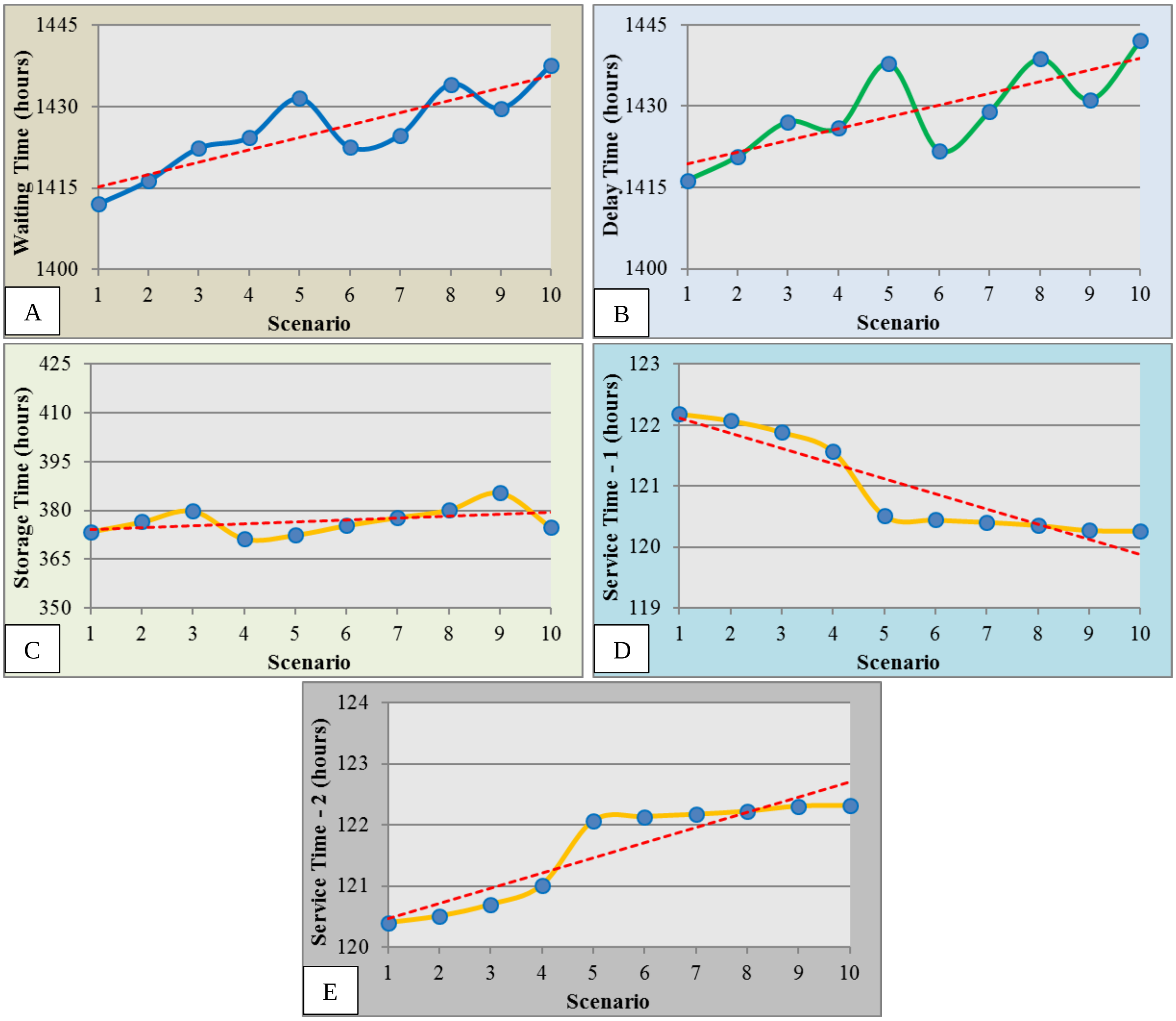
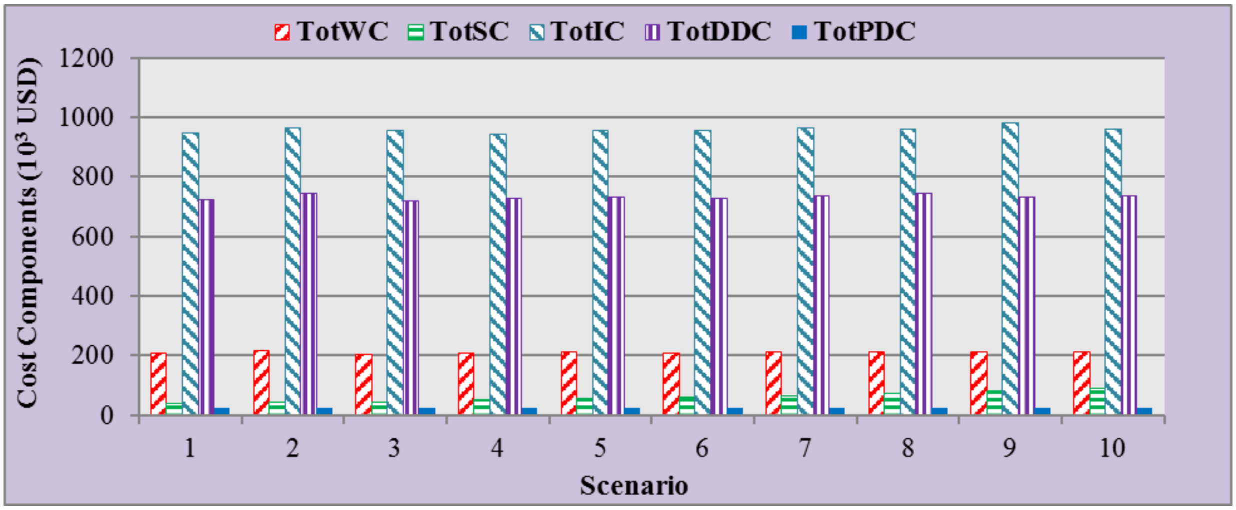


Figure 44 (A-E): Sensitivity analysis for the unit service cost of trucks.

It can be observed that as the unit service cost of trucks increased, the total service time of the arriving inbound trucks decreased by assigning these trucks to the inbound CDT doors that were located closer to the designated temporary locations of storage areas for the delivered products. A reduction in the total service time of the arriving inbound trucks allowed decreasing the associated truck service cost. However, in many instances, the assigned inbound CDT doors were not available immediately upon arrival of the inbound trucks, which further increased the total time of truck waiting as well as the total delayed departure time of trucks. The storage time of the delivered products did not fluctuate substantially from one scenario to another. Moreover, it was observed that the total service time of inbound trucks decreased from 122.2 hours in scenario “1” to 120.3 hours in scenario “10”. Furthermore, the total cost (i.e., $\boldsymbol{TC}$ - the **TSP-CDT** objective function value), associated with the service of the trucks arriving at the CDT, was significantly affected by an increase in the unit service cost of trucks. In particular, the total cost of serving trucks increased from 1,940.52 ($\times 10^3$ USD) in scenario "1" to 2,026.76 ($\times 10^3$ USD) in scenario "10".



Note: TotWC - total cost of waiting; TotSC - total cost of service; TotIC - total cost of product storage; TotDDC - total cost of delay on the departure of trucks; TotPDC - total cost of perishable product decay

Figure 45: Cost components for the unit truck service cost sensitivity analysis.

6.4.4 Assessment of the impact of the unit cost of product storage on the scheduling of trucks

Ten scenarios of the unit cost of product storage were developed. These scenarios were generated by increasing the value of the base unit cost of product storage, based on uniform distribution $U[25.66;\ 51.32]$. Each successive scenario was set 10% greater than the previous scenario (i.e., the unit cost of product storage was set as $U[25.66;\ 51.32]$ in scenario "1", while the unit cost of product storage was set as $U[60.50;\ 121.00]$ in scenario "10"). The largest among the realistic-size problem instances was used for the sensitivity analysis. Note that the problem instance "60", which is the largest one, has a total of 8 CDT doors (i.e., inbound and outbound doors) and 120 arriving trucks (i.e., trucks both in- and outbound). The other **TSP-CDT** mathematical model parameters were assumed to remain constant, as specified in section 6.1.1, while the unit cost of product storage was varied.

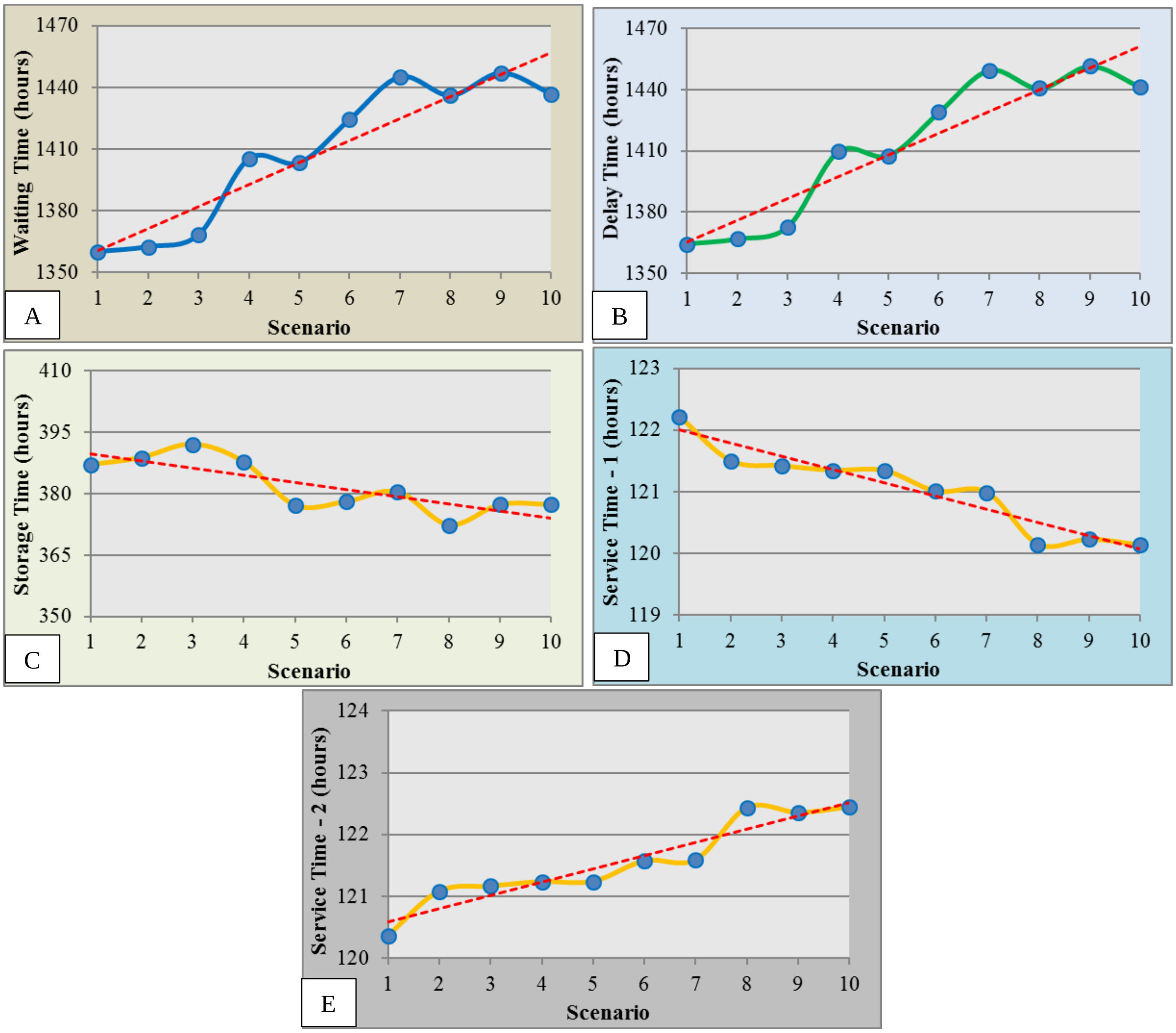
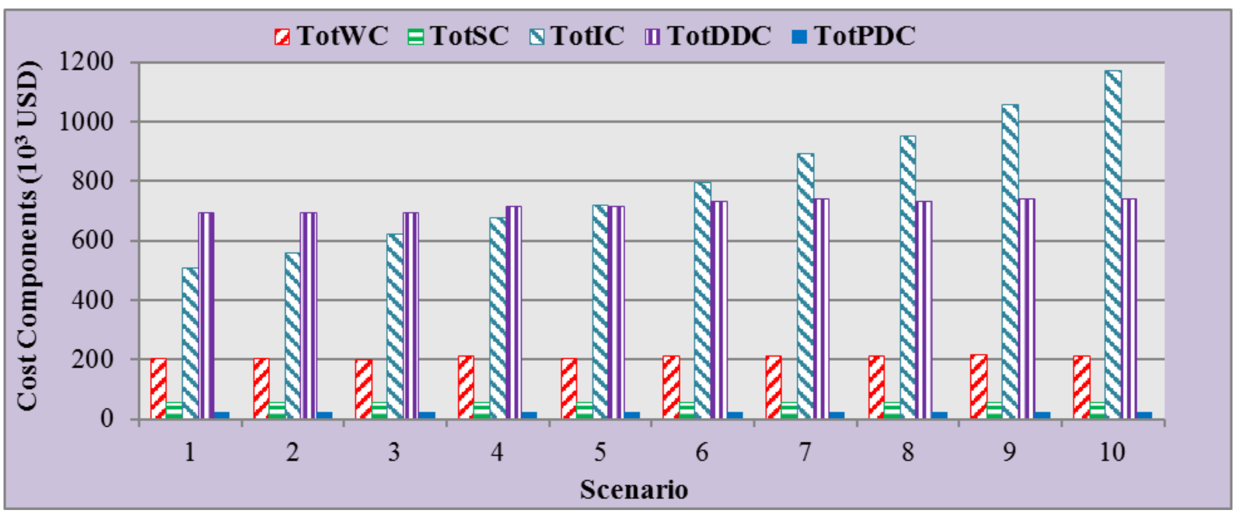


Figure 46 (A-E): Sensitivity analysis for the unit cost of product storage.

The results of the sensitivity analysis that was conducted are presented in Figure 46 (A-E) and Figure 47. These results show the changes in waiting time of trucks, truck delayed departure time, total storage time, service time of trucks both in- and outbound (denoted as "Service Time - 1" for the total service time of the inbound trucks and "Service Time - 2" for the total service time of the outbound trucks), and individual cost components of the objective function. In order to account for probable stochasticity in the results, a total of 10 algorithmic replications of each scenario were run, and the average values were recorded.

It can be observed that as the unit cost of product storage increased, the total storage time of the products delivered by the arriving inbound trucks decreased by synchronizing the start service times of the arriving inbound trucks and corresponding outbound trucks (where the products would be loaded). A reduction in the storage time of the products allowed decreasing the associated product storage cost. However, in many instances, the assigned CDT doors were not available immediately upon arrival of the inbound and outbound trucks, which further increased the total time of truck waiting as well as the total delayed departure time of trucks. It was also noticed that the total service time of the arriving inbound trucks decreased from one scenario to another by assigning these trucks to the inbound CDT doors that were located closer to the designated temporary locations of storage areas for the delivered products (which further allowed more effective transfer of the delivered products to the designated temporary locations of storage areas).



Note: TotWC - total cost of waiting; TotSC - total cost of service; TotIC - total cost of product storage; TotDDC - total cost of delay on the departure of trucks; TotPDC - total cost of perishable product decay

Figure 47: Cost components for the unit cost of product storage sensitivity analysis.

Moreover, it was observed that the total storage time of the delivered products decreased from 387.1 hours in scenario “1” to 377.3 hours in scenario “10”. Furthermore, the total cost (i.e., $\boldsymbol{TC}$ - the **TSP-CDT** objective function value), associated with the service of the trucks arriving at the CDT, was significantly affected by an increase in the unit cost of product storage. In particular, the total cost of serving trucks increased from 1,484.41 (×10^3 USD) in scenario “1” to 2,204.59 (×10^3 USD) in scenario “10”.

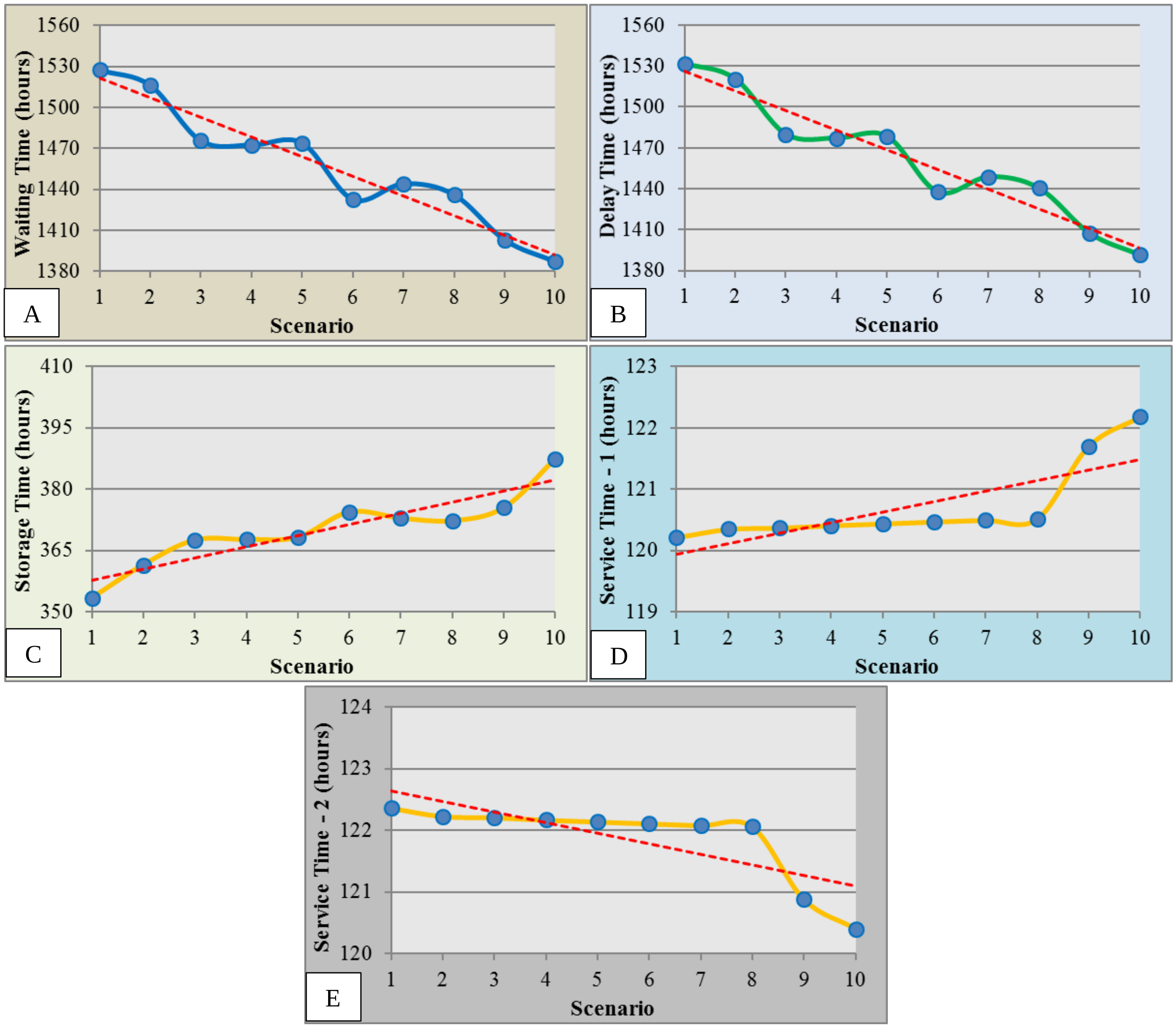


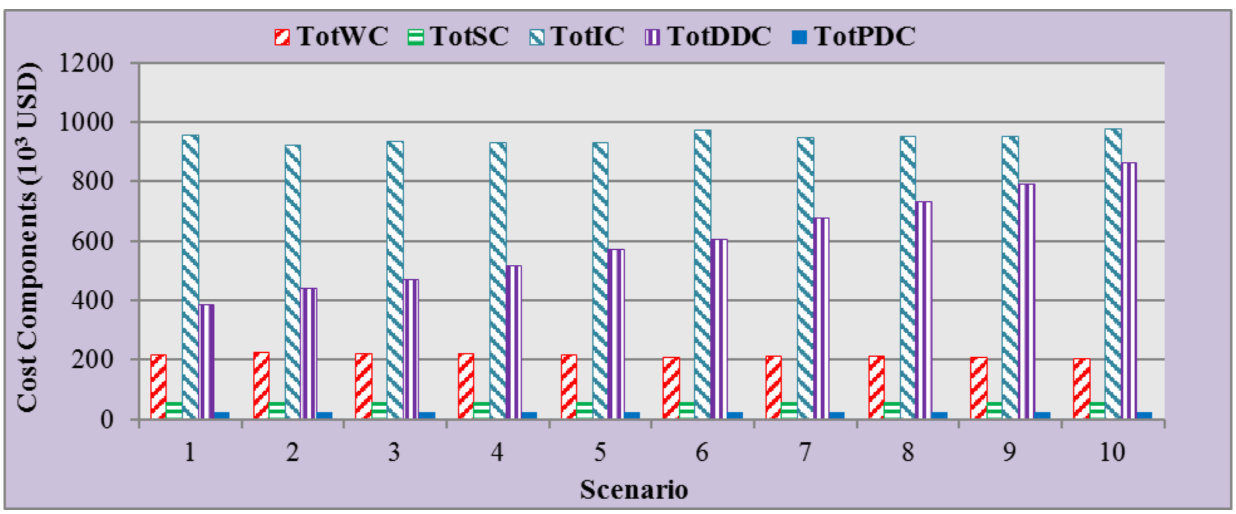
Figure 48 (A-E): Sensitivity analysis for the unit delayed departure cost of trucks.

6.4.5 Assessment of the impact of unit cost of the delayed departure on the scheduling of trucks

Ten scenarios of the unit delayed departure cost of trucks were developed. These scenarios were generated by increasing the value of the base unit delayed departure cost, based on uniform distribution $U[179.61;\ 359.22]$. Each successive scenario was set 10% greater than the previous scenario (i.e., the unit delayed departure cost of trucks was set as $U[179.61;\ 359.22]$ in scenario “1”, while the unit delayed departure cost of trucks was set as $U[423.50;\ 847.00]$ in scenario

"10"). The largest among the realistic-size problem instances was used for the sensitivity analysis. Note that the problem instance "60", which is the largest one, has a total of 8 CDT doors (i.e., inbound and outbound doors) and 120 arriving trucks (i.e., trucks both in- and outbound). The other **TSP-CDT** mathematical model parameters were assumed to remain constant, as specified in section 6.1.1, while the unit delayed departure cost of trucks was varied.

The results of the sensitivity analysis that was conducted are presented in Figure 48 (A-E) and Figure 49. These results show the changes in waiting time of trucks, truck delayed departure time, total storage time, service time of trucks both in- and outbound (denoted as "Service Time - 1" for the total service time of the inbound trucks and "Service Time - 2" for the total service time of the outbound trucks), and individual cost components of the objective function. In order to account for probable stochasticity in the results, a total of 10 algorithmic replications of each scenario were run, and the average values were recorded.



Note: TotWC - total cost of waiting; TotSC - total cost of service; TotIC - total cost of product storage; TotDDC - total cost of delay on the departure of trucks; TotPDC - total cost of perishable product decay

Figure 49: Cost components for the unit delayed truck departure cost sensitivity analysis.

It can be observed that as the unit delayed departure cost of trucks increased, the waiting time and the delayed departure time reduced. The latter can be explained by the fact that an increase in the unit delayed departure cost could prohibitively increase the total cost of serving the arriving trucks; therefore, the arriving trucks were assigned to the first available CDT doors (especially, inbound doors) to reduce the delayed departure time. The latter generally caused the storage time to increase, as the first available CDT doors were located farther away from the designated temporary locations of storage areas for the delivered products. Such finding also explains the fact that the total service time for the inbound trucks generally increased with increasing unit delayed departure cost of trucks. Moreover, it was observed that the total delayed departure time of trucks decreased from 1,531.6 hours in scenario "1" to 1,391.6 hours in scenario "10". Furthermore, the total cost (i.e., $\boldsymbol{TC}$ - the **TSP-CDT** objective function value), associated with the service of the trucks arriving at the CDT, was significantly affected by an increase in the unit delayed departure cost of trucks. In particular, the total cost of serving trucks increased from 1,635.61 ($\times 10^3$ USD) in scenario "1" to 2,122.21 ($\times 10^3$ USD) in scenario "10".

6.4.6 Assessment of the impact of the truck arrival patterns and the CDT door capacity on the scheduling of trucks

Thirty scenarios were considered throughout the analysis of realistic-size problem instances, where the number of available CDT doors was changed from a minimum of 4 to a maximum of 8 (with an increment of 2 doors), and the number of arriving trucks was changed from a minimum of 84 to a maximum of 120 (with an increment of 4 trucks). The truck arrival patterns (i.e., the number of arriving trucks both in- and outbound) and the CDT door capacity (i.e., the number of available CDT doors both in- and outbound) are expected to impact the scheduling of trucks. For example, if there are a lot of trucks arriving for service at the CDT and there are only a few CDT doors available for service, the arriving trucks both in- and outbound are expected to experience significant waiting time as well as significant delays in their departure from the facility.

Figure 50 (A-F) shows the changes in waiting time of trucks, truck delayed departure time, total storage time, service time of trucks both in- and outbound, decay of the perishable products handled at the CDT, and the objective function (i.e., the total cost that is associated with service of the arriving trucks both in- and outbound) for different combinations of the truck arrival patterns and the CDT door capacities. Furthermore, Figure 51 (A-F) shows the changes in waiting time of trucks, truck delayed departure time, total storage time, service time for trucks both in- and outbound, decay of the perishable products handled at the CDT, and the objective function (i.e., the total cost that is associated with service of the arriving trucks both in- and outbound) for different CDT door capacities, aiming to specifically identify the impacts of the number of available CDT doors on the scheduling of trucks.

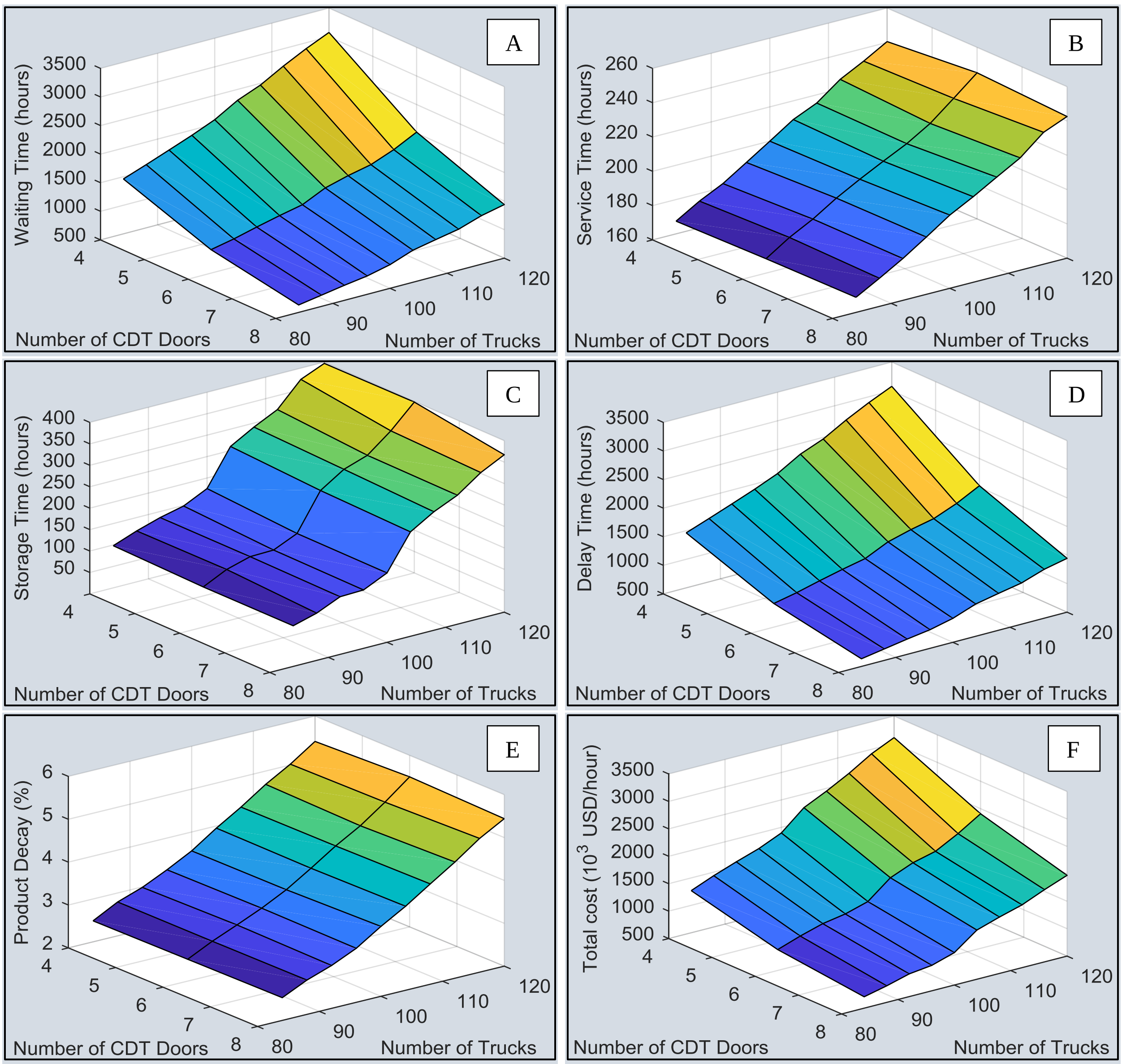


Figure 50 (A-F): Sensitivity analysis for the CDT capacity and the number of arriving trucks.

It can be observed that Figure 50 (A-F) and Figure 51 (A-F) confirm the initial expectations (i.e., the truck arrival patterns and the CDT door capacity may substantially impact the scheduling of trucks). The conducted numerical experiments show that the total waiting time of trucks can be effectively reduced by increasing the CDT door capacity (e.g., in case of 120 trucks to be served at the CDT, the total waiting time of trucks can be reduced from 3,074.6 hours to 1,437.4 hours by increasing the number of the CDT doors available from 4 to 8). Similarly, the total delayed departure time of trucks can be effectively reduced by increasing the CDT door capacity (e.g., in case of 120 trucks to be served at the CDT, the total delayed departure time of trucks can be reduced from 3,076.7 hours to 1,441.6 hours by increasing the number of CDT doors available from 4 to 8).

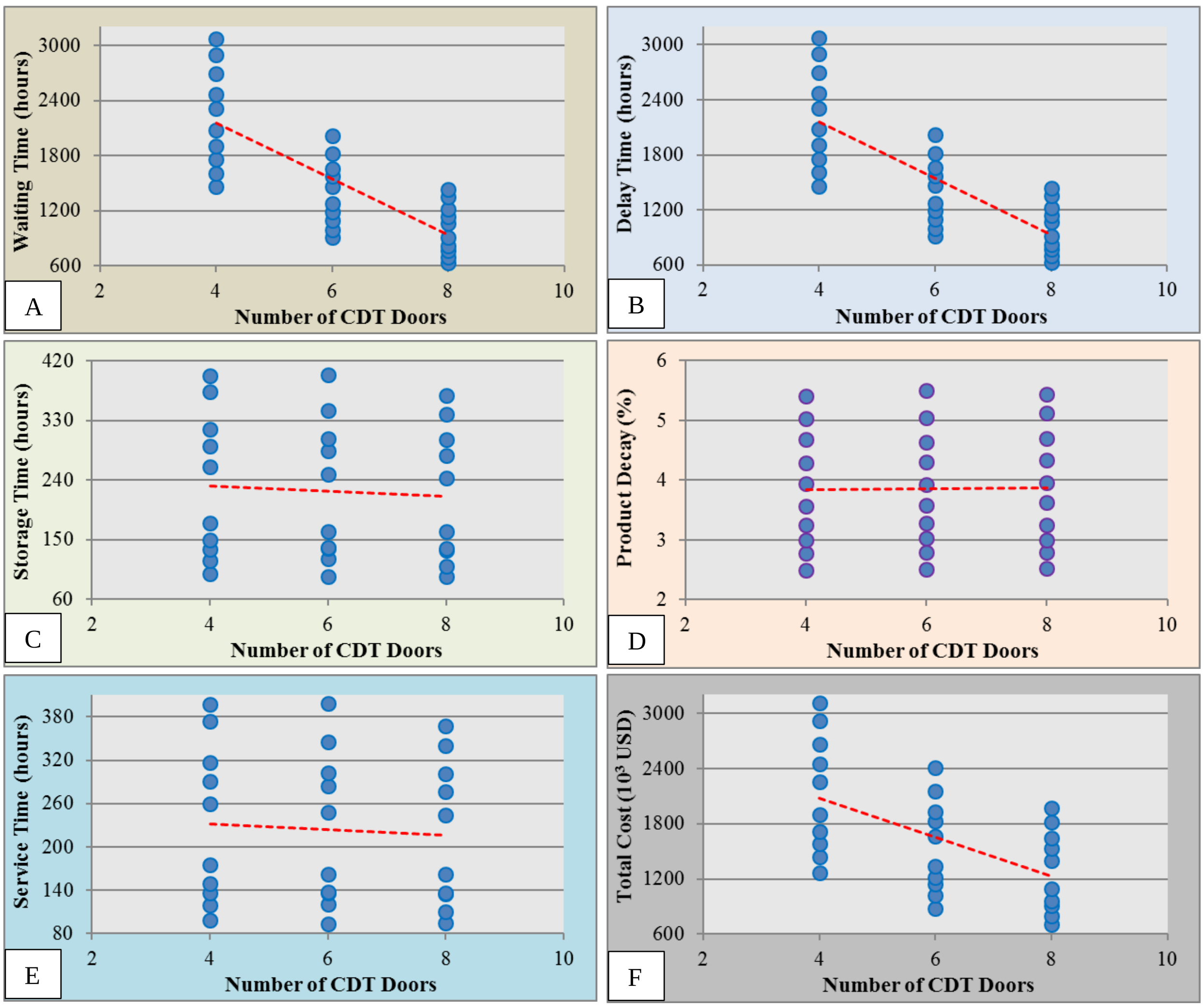


Figure 51 (A-F): Sensitivity analysis for the number of CDT doors.

Moreover, the numerical experiments show that the total storage time of the delivered products, the total service time of the arriving trucks, and the total decay of the perishable products handled at the CDT were primarily affected by truck arrival patterns. No significant changes in the total storage time of the delivered products, the total service time of the arriving trucks, and the total decay of the perishable products handled at the CDT were noticed after increasing the number of CDT doors available. However, it can be clearly seen that both truck arrival patterns and the CDT door capacity substantially influence the total cost (i.e., $\boldsymbol{TC}$ - the **TSP-CDT** objective function value), associated with the service of the trucks arriving at the CDT. In particular, in case of 120 trucks to be served at the CDT, the total cost that is associated with the service of the trucks can be reduced from 3,108.67 ($\times 10^3$ USD) to 1,972.21 ($\times 10^3$ USD) by increasing the number of the CDT doors available from 4 to 8. Furthermore, in case the CDT had only 4 doors available, the total cost that is associated with the service of trucks increased from 1,268.34 ($\times 10^3$ USD) to 3,108.67 ($\times 10^3$ USD) after increasing the number of arriving trucks from 84 to 120 over the considered planning horizon.

CHAPTER SEVEN

7 CONCLUSIONS

The efficiency of the operations of a supply chain depends on the product distribution strategy adopted. Supply chain managers are charged with the responsibility of making a choice among several product distribution strategies. The goals of an efficient supply chain include among others: cost reduction, improvement of customers satisfaction, maximization of profit and minimization of waste in terms of perishable products. Supply chain managers are, therefore, in constant search for promising methods that can ensure that the aforementioned supply chain goals are attained through the process of planning, analysis, and collaborative efforts of stakeholders. An important strategy that has been identified for efficient supply chain management is the cross-docking strategy. It is a form of product distribution strategy where the cross-docking terminal (CDT) or facility is entirely dedicated to the transshipment and distribution of products and container loads.

The cross-docking strategy is based on the logistics process, which consists in the reception of inbound products that have been transported by trucks/trailers, unloading and transferring the products across the dock to outbound doors (with or without temporary storage), where the outbound trucks/trailers are docked to pick up the consolidated products, which are ready to be shipped out. This process enhances the efficiency of the supply chain operations, reduces handling times and operational cost using various handling equipment types (such as sorters, combiners, conveyor belts, and others). Cross-docking has become an integral part of numerous supply chains over the recent years.

As a part of this dissertation, a comprehensive review of the relevant studies on CDT truck scheduling (i.e., assignment of trucks to the CDT doors and their service order determination) was carried out. The studies were grouped into various categories, such as: (1) general CDT truck scheduling; (2) multi-objective CDT truck scheduling; (3) uncertainty modeling in CDT truck scheduling; and (4) miscellaneous studies. A literature summary was provided based on the specific features of each CDT study reviewed. From the detailed review of studies, it was found that although several models were formulated to address the key problems of operations at

CDTs, most of the proposed models were not developed to effectively handle cold-chain CDTs (i.e., CDTs that handle perishable products).

It was also found that some studies focused on detailed modeling of behavior of perishable products throughout various supply chain operations. However, these studies did not explicitly capture cross-docking operations. Hence, this research focused on developing a novel mixed integer programming model that captured the major CDT operations and also the decay of products that are perishable in nature throughout the CDT operations with a particular focus on the CDT truck scheduling. The proposed mathematical model was named as a truck scheduling problem at the CDT with perishable products (**TSP-CDT**).

The **TSP-CDT** was initially formulated as a non-linear mixed integer programming model due to some non-linear components of the model. The model aimed to minimize the total cost of CDT operations which comprised of: (i) the total cost of waiting; (ii) the total cost of service; (iii) the total cost of product storage; (iv) the total cost of delay on the departure of trucks; and (v) the total cost of perishable product decay. The model was linearized using the linear static secant approximation in order to reduce the complexity involved in the model’s computation. The reformulated model, referred to as **TSP-CDTL**, was solved using CPLEX, and optimal solutions were reached. The assumptions and data used for the model were adopted from the previous studies on the scheduling of trucks at a CDT. Furthermore, the estimation of the product decay was done by adopting the exponential decay functions for the considered perishable product types, which is in line with the studies on perishability of products.

The complexity of the proposed model was examined and was found to be NP-hard. Therefore, a realistic problem instance with a significant number of arriving trucks and available CDT doors might require a fairly high computational time. Alternative solution methodologies were proposed to address the realistic-size problems and obtain solutions with good quality within a reasonable computational time. This dissertation considered four primary metaheuristic algorithms which were (1) Evolutionary Algorithm (EA); (2) Variable Neighborhood Search (VNS); (3) Tabu Search (TS); and (4) Simulated Annealing (SA) as alternative solution methodologies. These metaheuristic algorithms have been widely used for optimizing various operations, including the CDT truck scheduling, as stated in the previous studies on this topic.

EA is recognized as a population-based metaheuristic algorithm, where the EA operators are deployed to search for good-quality solutions in a population of generated solutions, referred to as chromosomes. Unlike the canonical EAs, the proposed EA used a special heuristic (the First Come First Served (FCFS) policy with truck sequence considerations that was named as "FCFS-TSC") for a half of the chromosome initialized, while the other half was initialized based on the random mechanism. The VNS metaheuristic searches for an improved solution within a finite number of neighborhood structures. On the other hand, the TS metaheuristic algorithm performs its search for an improved solution by moving from one solution to another, while prohibiting the visit of previous solutions (hence, the word "tabu" is applied).

Moreover, the SA metaheuristic algorithm performs its search for a more promising solution by using the annealing concept, which prevents the acceptance of poor-quality solutions as the number of iterations increases. The canonical VNS, TS, and SA are all single-solution-based metaheuristic algorithms. However, the proposed SA was designed to generate multiple neighbor solutions in order to enhance its explorative capabilities. The algorithmic parameter tuning analysis was performed in this dissertation for all the metaheuristic algorithms considered to enhance their performance by setting the appropriate parameter values from a set of candidate values for each one of the parameters.

The numerical experiments started with a comparative evaluation of the proposed metaheuristics with CPLEX. This analysis was conducted using the small-size problem instances, where the number of arriving trucks ranged from 6 to 15 and the number of CDT doors ranged from 2 to 4. It was revealed that the metaheuristic algorithms considered achieved good-quality solutions in a reasonable computational time. CPLEX, on the other hand, required higher computation time to achieve the optimal solution as the problem size increased. Moreover, the performance of metaheuristic algorithms was evaluated using the balance between the values of the objective function and the CPU time for the realistic-size problem instances as well, where the number of arriving trucks ranged from 84 to 120 and the number of CDT doors ranged from 4 to 8.

It was revealed that the proposed EA was able to achieve the best objective function values in a reasonable computational time. Moreover, the developed EA demonstrated acceptable stability levels in terms of the objective function values at termination and CPU time required. The numerical experiments were concluded with a set of sensitivity analyses that provided some managerial insights. These sensitivity analyses were performed using the EA metaheuristic, which was found to be the most promising one.

These analyses included: (1) assessment of the impact of the rate of product decay on the scheduling of trucks; (2) assessment of the impact of the unit waiting cost of trucks on the scheduling of trucks; (3) assessment of the impact of the unit service cost of trucks on the scheduling of trucks; (4) assessment of the impact of the unit cost of temporary storage on the scheduling of trucks; (5) assessment of the impact of unit cost of the delayed departure on the scheduling of trucks; and (6) assessment of the impact of the truck arrival patterns (i.e., the number of arriving trucks both in- and outbound) and the CDT door capacity (i.e., the number of available CDT doors both in- and outbound) on the scheduling of trucks.

The results of the sensitivity analyses showed how the total waiting time of trucks, total delayed departure time of trucks, total storage time of the delivered products, total product decay, total service time of inbound trucks, total service time of outbound trucks, and different cost components (i.e., the components of the adopted objective function) varied with the unit costs, truck arrival patterns (i.e., the number of arriving trucks both in- and outbound), and the CDT door capacity (i.e., the number of available CDT doors both in- and outbound). It was observed that the scheduling of trucks could be significantly affected by the rate of product decay, unit truck waiting cost, unit truck service cost, unit cost of product storage, unit delayed truck departure cost, truck arrival patterns, and the CDT capacity. Therefore, the proposed mixed integer **TSP-CDT** mathematical model, the EA metaheuristic deployed, and the managerial insights drawn from the sensitivity analyses could assist CDT managers in making decisions regarding perishable product distribution and facilitate supply chain operations.

CHAPTER EIGHT

8 FUTURE RESEARCH NEEDS

This dissertation has proposed a mathematical formulation **TSP-CDT** for the CDT truck scheduling problem. In addition to the modeling of the operations of the CDT, the developed model directly incorporated the estimation and minimization of the product decay throughout the truck service. However, certain assumptions were made in order to address the complexity of the problem under consideration. These assumptions could be further reexamined as a part of future studies. In particular, the following extensions could be further investigated:

- The CDT modeled in this study was assumed to be I-shaped, and the future research could be conducted to examine the effect of a CDT shape on the scheduling decisions of trucks.
- The CDT door service mode considered throughout this study assumed the segregated mode, and the future research could examine other modes, such as a mixed door service mode or a hybrid of segregated and mixed door service modes.
- The CDT truck scheduling model in this study did not allow preemption (i.e., a given truck could leave the door, so another truck could dock at that door for service). The future research could incorporate preemption as it could potentially lead to some operational improvements for the considered CDT.
- The arrival pattern of the trucks in real world is associated with some uncertainties, which can be due to several unforeseen circumstances (e.g., inclement weather during the transportation process). However, this study modeled the arrival times of trucks as deterministic (i.e., based on the known schedules that were previously established with the logistics companies). Therefore, uncertainty in the arrival times of trucks can be considered as a part of the future research.
- The handling times of the trucks in real world are associated with some uncertainties, which can be due to several unforeseen circumstances (e.g., breakdowns in a conveyor belt system). However, this study modeled the handling times of trucks as deterministic.

Therefore, uncertainty in the handling times of trucks could be considered as a part of the future research.

- The temporary storage capacity of a CDT in real world is limited. However, this study does not account for this limitation. Therefore, the future research could impose certain realistic limitations on the temporary storage capacity of the CDT, which would make the developed optimization model more accurate.
- The resource handling capacity in real world is limited. Therefore, certain realistic constraints could be imposed on the resource handling capacity, which would make the developed optimization model more accurate.
- The decay of perishable products is product-specific in the real world. As a part of the future research, a number of additional cases studies can be conducted for specific products to determine the changes in the truck scheduling decisions.
- This study adopted a linear static secant-approximation method, which is one of several common approaches for approximating non-linear functions. The future research could critically compare the result with alternative approximation methods, such as "quadratic static outer-approximation method" and "linear branch-and-bound outer-approximation method".
- The future research could consider the development of a multi-objective framework in order to critically examine the conflicting objectives (i.e., objectives that are negatively correlated).
- This study considered four metaheuristics, including (1) Evolutionary Algorithm (EA); (2) Variable Neighborhood Search (VNS); (3) Tabu Search (TS); (4) Simulated Annealing (SA). The future research could compare the best performed metaheuristic (which is EA) with alternative metaheuristics (e.g., Imperialist Competitive Algorithm, Differential Evolution, Red Deer Algorithm, Particle Swarm Optimization, and Grey Wolf Optimizer).
- The future research could also deploy local search heuristics (or even exact optimization procedures) within the developed EA to further improve the fitness of the offspring chromosomes produced.

8.1 Consideration of Uncertainties in Truck Scheduling at CDT

There are several possible uncertainties in the truck scheduling at a CDT that could be addressed in the future research. Factors, such as weather, traffic conditions, and other external factors, could lead to uncertainty in the arrival of trucks both in- and outbound. According to Konur and Golias (2013a,b), the first come first served (FCFS) policy could be used to model the uncertainty in the arrival of trucks (which is more of a canonical approach). In this case, the truck schedule is incomplete until the arrival of the last truck. This may result in high waiting cost for the trucks that have already arrived (e.g., the truck that arrived last would be scheduled for service before the other trucks that are already present at the facility).

Also, the handling times and waiting times of the arriving trucks are not fixed and subject to changes. Therefore, these uncertainties could also be addressed as a part of the future research. Several methods could be used in addressing the aforementioned uncertainties, such as minimax regret method, application of upper and lower bounds, cardinality-constrained method, sample average approximation method, scenario analysis method, and others. Furthermore, the future research could conduct a comparative evaluation of these methods while incorporating uncertainties in the proposed **TSP-CDT** mathematical model. The following sections contain a detailed discussion of the approaches that could be potentially used in modeling the uncertainties in the truck scheduling at a CDT.

8.1.1 Minimax regret method

The minimax regret method is used to minimize the potential loss that can result from the worst-case scenario. This method relies on the accurate estimation of the regret or potential loss. The regret is estimated as the difference of the payoff of a given outcome to the payoff of the optimal outcome. This method is advantageous since it is independent of the probabilities of different outcomes. Therefore, it can be used in addressing uncertainties. The areas of application of this method include hypothesis testing, prediction, project scheduling, among others. The following illustration shows a case, where the arrival times of trucks at a given CDT are uncertain. The arrival times of trucks could be early, on time (as originally scheduled), or late (e.g., these truck arrival alternatives would be referred to as $Arr1$, $Arr2$, and $Arr3$, respectively).

Depending on the existing CDT capacity for a given planning horizon (e.g., changes in the number of available CDT doors for a given planning horizon), changes in the arrival time could result in higher of lower payoffs. For example, if a truck arrives before its scheduled arrival time, and the CDT does not have any available doors, the truck might be required to wait for an extended time to be served, which may cause substantial truck service costs (therefore, will reduce the total payoff). On the other hand, arriving later than the scheduled arrival time could be beneficial for a given truck, as some CDT doors could be available by the time the truck arrives, which would reduce the truck service costs and increase the total payoff. However, in certain instances, late truck arrivals could significantly disrupt the CDT operations and incur excessive truck service costs and lower payoffs. The minimax regret method estimates the regret that results from the deviation of any outcome from the optimal outcome.

An example of different payoff values for different truck arrival decisions and CDT capacity scenarios is presented in Table 11. It could be observed that the arrival decision $Arr2$ would have the highest payoff under the CDT capacity scenario $C1$ (i.e., the payoff value of \$21). On the other hand, the arrival decision $Arr3$ would have the lowest payoff under the CDT capacity scenario $C1$ (i.e., the payoff value of \$11). Therefore, the regret value for making the arrival decision $Arr3$ under the CDT capacity scenario $C1$ would be (\$21 - \$11) = \$10. In the meantime, the regret value for making the arrival decision $Arr2$ under the CDT capacity scenario $C1$ would be (\$21 - \$21) = \$0, since the arrival decision $Arr2$ is the optimal decision that has the highest payoff.

Table 11: Payoff values for each arrival decision and CDT capacity scenario.

	CDT Capacity Scenario		
Arrival Decisions	$C1$	$C2$	$C3$
$Arr1$	\$18	\$1	\$10
$Arr2$	\$21	\$17	-\$3
$Arr3$	\$11	\$24	\$1

Table 12 provides the estimated regret values for each arrival decision and CDT capacity scenario. The results from the conducted analysis show that the arrival decision $Arr3$ (i.e., arriving later that it was originally scheduled) would be the most promising over all the considered CDT capacity scenarios (i.e., the total regret would be \$10 + \$0 + \$9 = \$19 over all the considered CDT capacity scenarios). On the contrary, the arrival decision $Arr1$ (i.e., arriving earlier that it was originally scheduled) would be the least promising over all the considered CDT capacity scenarios (i.e., the total regret would be \$3 + \$23 + \$0 = \$26 over all the considered CDT capacity scenarios).

Table 12: Regret values for each arrival decision and CDT capacity scenario.

	CDT Capacity Scenario		
Arrival Decisions	$C1$	$C2$	$C3$
$Arr1$	\$3	\$23	\$0
$Arr2$	\$0	\$7	\$13
$Arr3$	\$10	\$0	\$9

8.1.2 Application of upper and lower bounds

When modeling factors that may have high variability (such as arrival times of trucks), the application of upper and lower bound could be useful. Since the exact arrival time of trucks cannot be determined with certainty, a time window could be assumed. The time window would have a lower bound, which corresponds to the earliest possible time of arrival, and an upper bound, which corresponds to the latest possible arrival time for each truck. Solutions could, therefore, be developed for the arrival times within the assumed arrival time window (Konur and Golias, 2013b). Let Arr_t^l be the lower bound of the arrival time of truck t, and Arr_t^u be the upper bound of the arrival time of truck t:

$$Arr_t \in [Arr_t^l, Arr_t^u] \; \forall t \in T \tag{31}$$

where $[Arr_t^l, Arr_t^u]$ is the arrival time window of truck t. Since the CDT manager does not know the exact arrival time of a truck, the arrival time of each given truck is, therefore, could be sampled from a probability distribution that has a lower bound of $Arr_t^l, t \in T$ and an upped bound of $Arr_t^u, t \in T$.

8.1.3 **Cardinality-constrained method**

In the future research, uncertainty in the components of the **TSP-CDT** mathematical model could be addressed using the cardinality-constrained method. A parameter in a mathematical model may be uncertain; however, such parameter does not necessarily have to have the worst value. For example, the arrival time of trucks at the CDT may be uncertain due to several factors. However, it does not mean that every truck will violate its expected arrival time at the CDT that will result in the worst-case scenario truck arrival times, which would further increase the total cost associated with the service of trucks. The cardinality-constrained method addresses uncertainty with the assumption that only a fraction of the total set of parameters may be adversely affected by the values of the uncertain parameters (Raith et al., 2018). This method could offer a promising approach to addressing uncertainty in the **TSP-CDT** mathematical model (e.g., only a portion of the arriving trucks would have the worst-case scenario arrival times at the CDT due to the associated uncertainties).

8.1.4 **Scenario analysis**

This method could be used in the future research to evaluate each one of the pre-defined scenarios in order to address the uncertainties in the **TSP-CDT** mathematical model. Although, this method is somewhat similar to making a prediction, it mainly focuses on possible outcomes of uncertain factors. For example, if the arrival time of trucks is assumed to be uncertain, several scenarios could be developed to evaluate the outcome of the model as the arrival time changes from one value to the other. This method is, therefore, effective for evaluating several values of the uncertain parameter. It could present several alternative outcomes of the function, containing the uncertain parameter, as well as to emulate possible future implications (Quiceno et al., 2019). Moreover, the usefulness of this method in modeling is limited to the range of values assumed for the uncertain parameter. Defining a pre-determined set of potential values for a given parameter that is subject to uncertainties may be challenging in some cases.

8.1.5 **Sample average approximation**

The sample average approximation method uses the Monte Carlo Simulation to address stochasticity in a given optimization problem. This method could be used in the future research for the modeling of uncertainty in some of the parameters of the **TSP-CDT** mathematical model. Note that the sample average approximation method is appropriate for the use when the approximating function f_n has a form that can be determined by a deterministic optimization algorithm (Kim et al., 2015). The application of the sample average approximation method involves a number of the key steps that are further described next. First, the uncertain parameter in the optimization problem is identified. Second, n random replications of the parameter are generated. Third, the average value of the function that comprises of the uncertain parameter is taken. Therefore, the value of the uncertain parameter in the model is replaced by the average value of the random replications. Let's consider the following stochastic optimization problem, where γ is the parameter that is subject to uncertainty and $\boldsymbol{z}$ is a decision variable.

$$min\ F(\boldsymbol{z}) = c^T\boldsymbol{z} - \mathbb{E}(Q(\boldsymbol{z},\gamma)) \tag{32}$$

Subject to:

$$A\boldsymbol{z} = b \tag{33}$$

$$\boldsymbol{z} \in \mathbb{R}^+ \tag{34}$$

When the sample average approximation method is applied, it is required to generate a total of n random replications of the uncertain parameter γ, generally following a pre-determined probability distribution. After that, the expected value of the function that contains the uncertain parameter γ can be approximated based on the following relationship:

$$\mathbb{E}(Q(\boldsymbol{z},\gamma)) = \frac{1}{n}\sum_{i=1}^{n} Q(\boldsymbol{z},\gamma_i) \tag{35}$$

Hence, the original stochastic optimization problem with the uncertain parameter γ can be reformulated as follows:

$$min\ F(\boldsymbol{z}) = c^T\boldsymbol{z} - \frac{1}{n}\sum_{i=1}^{n} Q(\boldsymbol{z},\gamma_i) \tag{36}$$

Subject to:

$$Az = b \tag{37}$$

$$\boldsymbol{z} \in \mathbb{R}^{+} \tag{38}$$

The reformulated model can be further solved using a deterministic approach, since the stochasticity has been removed. Furthermore, the sample average approximation method could be performed repeatedly with many samples, as long as the probability distribution of the uncertain parameter is known. As for the **TSP-CDT** mathematical model, the uncertain parameter γ could represent the arrival time of trucks, the handling time of trucks, the decay rate of perishable products, along with other parameters.

8.2 Multi-Objective Optimization Approaches

The **TSP-CDT** mathematical model in this study was formulated as a single-objective optimization model. The objective, which is the total cost associated with the service of arriving trucks both in- and outbound, comprises of five cost components, including the following: (i) the total cost of waiting; (ii) the total cost of service; (iii) the total cost of product storage; (iv) the total cost of delay on the departure of trucks; and (v) the total cost of perishable product decay. Each of these cost components can be further examined. The future research could consider reformulating the **TSP-CDT** mathematical model as a multi-objective optimization model. Generally, the goal of a multi-objective model is to find a good tradeoff for several objectives (Edgeworth, 1881). The notion of optimality in multi-objective optimization was conceptualized by Pareto in 1896. Therefore, the term Pareto optimum was used to refer to a vector of decision variables $\vec{x}^* \in T$ if there does not exist another $\vec{x} \in T$ such that $f_i(\vec{x}) \leq f_i(\vec{x}^*)$ for all $i = 1, \dots, k$ and $f_j(\vec{x}) \leq f_j(\vec{x}^*)$ for at least one j.

A solution is referred to as Pareto optimal if there exists no feasible vector of decision variables that would cause a reduction in some criterion without simultaneously increasing in at least one other criterion. Usually, the concept of Pareto optimality does not result in a single solution but a set of solutions referred to as a Pareto optimal set (or Pareto Front). Therefore, the solution vector, which is among the Pareto optimal set, is also referred to as “non-dominated”.

The multi-objective optimization method is particularly useful when an optimization problem has conflicting objectives. An example of conflicting objectives in the **TSP-CDT** mathematical model is reducing the waiting time of trucks and reducing the service time of trucks simultaneously. In particular, in order to reduce the waiting time of trucks, the CDT manager would have to assign the arriving trucks both in- and outbound to the first available CDT doors. The latter would cause an increase in the service time of trucks, as the first available CDT doors may be located far away from the designated temporary locations of storage areas for the delivered products. On the contrary, the CDT manager can reduce the service time of trucks by assigning them the CDT doors that are located closer to the designated temporary locations of storage areas for the delivered products. However, the latter decision would lead to an increase in the waiting time of trucks, as these CDT doors may not be available immediately upon arrival of certain trucks. Therefore, minimization of the waiting time of trucks and minimization of the service time of trucks are the conflicting objective functions within the **TSP-CDT** mathematical model.

The following sections provide more information regarding different multi-objective concepts that could be further used in the **TSP-CDT** mathematical model as a part of the future research. In particular, different multi-objective optimization methods would be classified into two major groups, including the following: (i) Non-Pareto methods; and (ii) Approximate Pareto Front-based methods. Furthermore, an illustrative example of a Pareto Front for a multi-objective problem would be provided as well.

8.2.1 Non-Pareto methods

The Non-Pareto methods are used for addressing multi-objective mathematical models in a manner that does not use the Pareto optimality concept explicitly. These methods are aggregating functions that are relatively easy to implement (e.g., use weights to assign the importance to each one of the objective functions and aggregate the weighed functions afterwards) or treating these functions separately (e.g., one objective function is optimized at a time). Examples of Non-Pareto methods include the following:

- Aggregating function method (or weighted sum method);
- Epsilon-constraint method;
- Augmented epsilon-constraint method;
- Lexicographic goal programming method;
- Target-vector method;
- And others.

These methods are fairly easy to implement. However, they may not be suitable for handling multi-objective problems with too many objectives, and application of the alternative multi-objective methods might be required. More details regarding the epsilon-constraint method and the lexicographic goal programming method that could be applied to the **TSP-CDT** mathematical model are presented next.

8.2.1.1 Epsilon-constraint method

The epsilon-constraint method could be effectively used to address a multi-objective **TSP-CDT** mathematical model with conflicting objectives. Based on the epsilon-constraint method, one of the considered objective functions (generally, the most important objective function) is optimized, while constraint sets are imposed on the values of the other objective functions. The constraint sets bound the values of the other objective functions by certain levels ε_i. Once a single-objective optimization problem is solved for a given set of ε_i levels, the procedure is repeated again by adjusting the values of ε_i for each one of the considered objective functions (except the one that is optimized). Depending on the number of required iterations, the epsilon-constraint method may incur quite a substantial computational time. In case of bi-objective settings for the **TSP-CDT** mathematical model (e.g., minimize of the waiting time of trucks vs. minimize of the service time of trucks), the waiting time of trucks could be used as an objective, while an upper bound could be imposed on the service time of trucks.

8.2.1.2 Lexicographic goal programming method

Charnes and Cooper (1975) originally proposed goal programming as a method that could solve decision problems that involve multiple objectives. Lexicographic goal programming is a specialized goal programming method developed to solve multi-objective problems when the objectives are arranged in the lexicographic order. This method is, therefore, appropriate when multiple objectives have clear priority. Each of the objectives in a multi-objective problem has its importance, and some objectives are more important than others. Hence, there is a need for ranking these objectives. Although, this method is easy to implement, it requires additional comparative evaluation to put the objectives in order. This method can be used to address the multi-objective reformulation of the **TSP-CDT** mathematical model in the future research. In this case, the objectives would be ranked, and then the optimum solution would be obtained by minimizing each objective in the order of importance from the most important to the least important (e.g., the minimization of the waiting time of trucks could be prioritized over the minimization of the service time of trucks).

8.2.2 **Approximate Pareto Front-based methods**

These are the methods, referred to as heuristics or metaheuristics, that were developed to address multi-objective mathematical models. Such methods generally deploy stochastic optimization operators. Therefore, the Pareto Fronts that are returned by these methods after one run may differ from the Pareto Fronts that are returned by these methods after another run. Hence, such methods are named as approximate Pareto Front-based methods (i.e., the Pareto Front can be obtained but it may not be the optimal Pareto Front). A large variety of different Pareto Front-based methods have been used over the past years, including the following (Deb et al., 2000; Dulebenets, 2018d; Dulebenets et al., 2020):

- Non-dominated Sorting Genetic Algorithm (NSGA);
- Non-dominated Sorting Genetic Algorithm II (NSGA-II);
- Niched-Pareto Genetic Algorithm (NPGA);
- Pareto Archived Evolution Strategy (PAES);
- Pareto Envelope-based Selection Algorithm (PESA);
- And others.

The NSGA-II method was proposed by Deb et al. (2000) as an improved version of the initially proposed Non-dominated Sorting Genetic Algorithm (NSGA) by Srinivas and Deb (1994). The

NSGA earlier proposed was criticized as follows: 1) it has a fairly high computational complexity; 2) it does not use the elitism strategy (the elitism strategy preserves the best solution in each generation of the NSGA algorithm); and 3) it relies on a traditional mechanism for insuring diversity in the population; therefore, the sharing parameter has to be specified. The purpose of developing the NSGA-II algorithm was to address some of the major limitations of the NSGA algorithm. The NSGA-II algorithm relies on specialized crossover and mutation operators to produce offspring. Moreover, the next generation of individuals are selected using the non-dominated sorting and crowding distance comparison procedures. Furthermore, NSGA-II directly deploys the elitism strategy to ensure that the best solution will be presented in each generation of the algorithm. The NSGA-II method has been widely used in solving multi-objective optimization problems for a variety of different domains.

The NPGA method was originally proposed by Horn et al. (1994). The reformulation of the **TSP-CDT** model as multi-objective could be addressed using the NPGA algorithm. The NPGA method is based on Pareto dominance using the tournament selection. Two individuals are selected from the population of individuals and compared with a selected portion of the entire population. The key advantages of the NPGA algorithm include the following: (i) efficient as Pareto ranking does not have to be applied for the entire population; (ii) fairly easy to implement; and (iii) good performance for a wide array of different decision problems. However, along with the sharing factor, the NPGA algorithm also requires setting another parameter to perform the tournament selection.

The PAES method was originally introduced by Knowles and Corne (2000). This method uses the archive of the record of non-dominated vectors that were previously identified throughout the search process. In order to maintain diversity in the population of individuals, the adaptive grid is used by the PAES algorithm. Lastly, Corne et al. (2000) proposed the PESA method, which utilizes a small internal population of individuals, along with a larger external (i.e., secondary) population of individuals. Similar to the PAES method, PESA relies on the hyper-grid division of phenotype, aiming to maintain the population diversity. However, it also deploys the crowding distance measure. The crowding distance measure is used by the PESA method to decide on the solutions that will be moved to the secondary population of individuals (which represents the archive of non-dominated vectors that have been identified throughout the evolutionary process).

8.2.3 **Illustration of a Pareto Front**

A Pareto Front is a set of solutions in a system (usually, a system with multiple objectives) where no change in any component of the system could lead to an improvement in the system without worsening at least one other component of the same system. For example, assuming there is a cost T1 and a cost T2, which are conflicting costs in a system (e.g., the total waiting cost of trucks and the total service cost of trucks in the **TSP-CDT** mathematical model). There is a need to minimize the total cost of the system, which is dependent on the two aforementioned costs. The Pareto Front can be effectively used to demonstrate how changes in one cost component of the system would affect changes in the other cost component of the same system. A schematic illustration of a Pareto Front with two conflicting objectives (i.e., T1 and T2) is provided in Figure 52.

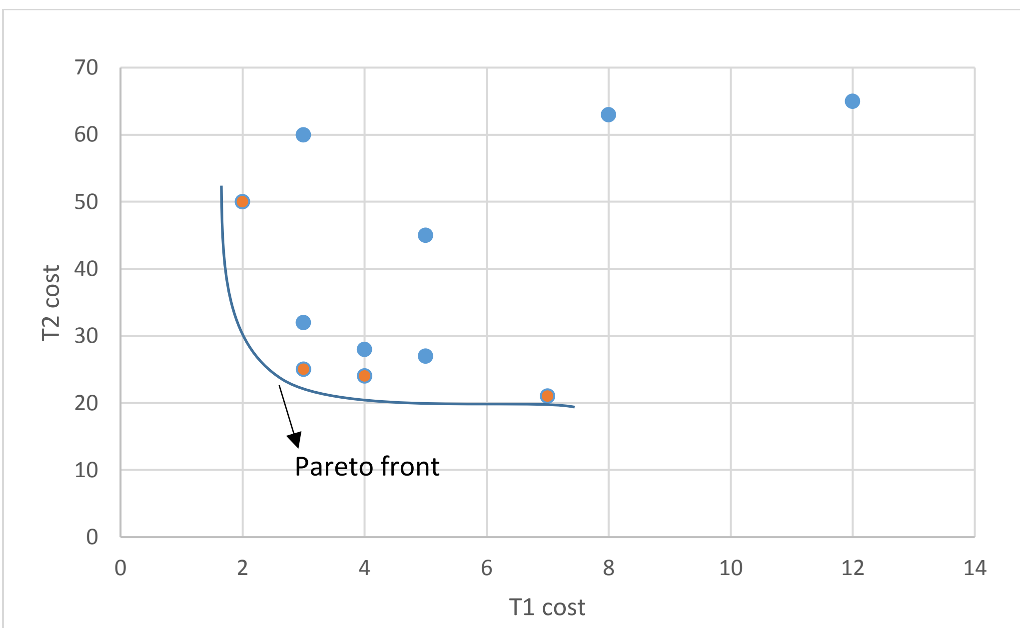


Figure 52: Pareto Front for conflicting objectives T1 and T2.

It can be observed that non-dominated solutions {7, 21}, {4, 24}, {3, 25}, and {2, 50} form a Pareto Front. The Pareto Front points {7, 21} and {2, 50} can be considered as "corner points" that have the best values of objective functions T2 and T1, respectively. On the other hand, the Pareto Front points {4, 24} and {3, 25} show a reasonable tradeoff between objectives T1 and T2.

REFERENCES

Abioye, O.F., Dulebenets, M.A., Pasha, J., & Kavoosi, M., 2019. A Vessel Schedule Recovery Problem at the Liner Shipping Route with Emission Control Areas. Energies 12 (12), p. 2380.

Abioye, O.F., Dulebenets, M.A., Kavoosi, M., Pasha, J., & Theophilus, O., 2020. Vessel Schedule Recovery in Liner Shipping: Modeling Alternative Recovery Options. IEEE Transactions on Intelligent Transportation Systems, pp. 1-15.

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.

Agustina, D., Lee, C.K.M., Piplani, R., 2010. A review: mathematical models for cross-docking planning. International Journal of Engineering Business Management 2, 47–54.

Ahumada, O., Villalobos, J., 2009. Application of planning models in the agri-food supply chain: A review. European Journal of Operational Research 196(1), 1-20.

Alpan, G., Ladier, A., Larbi, R., Penz, B. 2011. Heuristic solutions for transshipment problems in a multiple door cross. Computers & Industrial Engineering 61, 402–408.

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

Arkat, J., Qods, P., Ahmadizar, F. 2016. Truck scheduling problem in a cross-docking system with a release time constraint. Journal of Industrial and Systems Engineering 9, 1–16.

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

Bala, K., 2014. Supply Chain Management: Some Issues and Challenges - A Review. Supply Chain Management 8.

Barak Y., Gat A., 2016. Trends in Supply Chain Management. Tefen and Israeli Supply Chain Management Association.

Bartholdi, J., Gue, K., 2004. The best shape for a cross-dock. Transportation Science 38(2), 235-244.

Behnamian, J., Fatemi Ghomi, S.M.T., Jolai, F., Heidary, P., 2018. Location-Allocation and Scheduling of Inbound and Outbound Trucks in Multiple Cross-Dockings Considering Breakdown Trucks. Journal of Optimization in Industrial Engineering 11, 51-65.

Brimberg, J., 1996. A variable neighborhood algorithm for solving the continuous location-allocation problem. Studies in Locational Analysis 10, 1-12.

Bodnar, P., Azadeh, K., Koster, R., 2015. Scheduling Trucks in a Cross-Dock with Mixed Service Mode Dock Doors. Transportation Science 51(1), 112-131.

Boloori Arabani, A.R., Zandieh, M., Fatemi Ghomi, S.M.T., 2011. Meta-Heuristics Implementation for Scheduling of Trucks in a Cross-Docking System with Temporary Storage. Expert Systems with Applications 38 (3), 1964–79.

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

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

Boysen, N., 2010. Truck scheduling at zero-inventory cross-docking terminals. Computer & Operations Research 37, 32–41.

Bureau of Statistics, 2012. Commodity Flow Survey Public Use Microdata File. Retrieved from: https://www.bts.gov/cfs/2012-cfs-public-use-microdata-visuals

Bureau of Statistics, 2012. U.S. Department of Transportation, Bureau of Statistics and U.S. Department of Commerce, U.S. Census Bureau. https://www.bts.gov/cfs/2012-cfs-public-use-microdata-visuals

Bureau of Transportation Statistics and Federal Highway Administration, 2017. Freight Analysis Framework, Version 4.3.1.

Cekala, T., Telec, Z., Trawiński, B. 2015. Truck loading schedule optimization using a genetic algorithm for yard management. Asian Conference on Intelligent Information and Database Systems 536–548.

Charnes, A., Cooper, W.W. (1975) Goal Programming and Multiple objective optimizations (Part I). Center for cybernetic studies, Research report CCS 250.

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

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

Chmielewski, A., Naujoks, B., Janas, M., Clausen, U., 2009. Optimizing the Door Assignment in LTL-Terminals. Transportation Science 43, 198–210.

Choy, K.L., Chow, H.K.H., Poon, T.C., Ho, G., 2012. Cross-dock job assignment problem in space constrained industrial logistics distribution hubs with a single docking zone. International Journal of Production Research 50(9), 2439-2450.

Christopher, Martin L., 1992. Logistics and Supply Chain Management.

Cooper, M. C., Lambert, D. M., & Pagh, J. D., 1997. Supply Chain Management: More Than a New Name for Logistics. The International Journal of Logistics Management 8(1), 1-14.

Corne, D.W., Knowles, J.D., Oates, M.J., 2000. The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization. In: Schoenauer M. et al. (eds) Parallel Problem Solving from Nature PPSN VI. PPSN 2000. Lecture Notes in Computer Science, vol 1917. Springer, Berlin, Heidelberg

Cota, P.M., Gimenez, B.M.R., Araújo, D.P.M., Nogueira, T.H., de Souza, M.C., Ravetti, M.G. 2016. Time-indexed formulation and polynomial time heuristic for a multi-dock truck scheduling problem in a cross-docking centre. Computers & Industrial Engineering 95, 135–143.

Council of Supply Chain Management Professionals, 2013. Supply chain management terms and glossary.

Cross-docking trend report, 2011. Whitepaper series. Saddle creek corp.

CSCMP's Annual State Logistic Annual Report, 2011. Retrieved from: http://www.areadevelopment.com/logisticsInfrastructure/Q3-2013/CRE-supply-chain-strategy-logistics-labor-love-2272118.shtml

CSCMP's Global Supply Chain Statistics (2015) Retrieved from: https://project44.com/headline/global-supply-chain-statistics/

Darwin, C., 1859. On the Origin of species by means of natural selection, or the preservation of favored races in the struggle for life.

Deb, K., Agrawal, S., Pratap, A., Meyarivan, T., 2000. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In: Schoenauer M. et al. (eds) Parallel Problem Solving from Nature PPSN VI. PPSN 2000. Lecture Notes in Computer Science, vol 1917. Springer, Berlin, Heidelberg.

Deep, K., Mebrahtu, H., 2012. Variant of partially mapped crossover for the Travelling Salesman problems. International Journal of Combinatorial Optimization Problems and Informatics 3(1), 47-69.

Deveshwar, A., Rathee, R., 2010. Challenges for supply chain management in today's global competitive environment, 10.

Dulebenets, M.A., 2015a. Models and Solution Algorithms for Improving Operations in Marine Transportation. Dissertation Submitted to the Graduate School of the University of Memphis, May 2015.

Dulebenets, M.A., 2015b. Bunker Consumption Optimization in Liner Shipping: A Metaheuristic Approach. International Journal on Recent and Innovation Trends in Computing and Communication 3 (6), 3766-3776.

Dulebenets, M.A., 2016a. A New Simulation Model for a Comprehensive Evaluation of Yard Truck Deployment Strategies at Marine Container Terminals. Open Science Journal 1 (3), 1-28.

Dulebenets, M.A., 2016b. Advantages and Disadvantages from Enforcing Emission Restrictions within Emission Control Areas. Maritime Business Review 1 (2), 107-132.

Dulebenets, M. A., Kavoosi, M., Abioye, O., & Pasha, J. 2018. A Self-adaptive Evolutionary Algorithm for the Berth Scheduling Problem: Towards Efficient Parameter Control. Algorithms 11(7), p.100.

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 Diploid Evolutionary Algorithm for Sustainable Truck Scheduling at a Cross-Docking Facility. Sustainability 10 (5), p.1333.

Dulebenets, M.A., 2018b. 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., 2018c. Application of Evolutionary Computation for berth scheduling at marine container terminals: Parameter tuning versus parameter control. Transactions on Intelligent Transportation Systems 19 (1), 25–37.

Dulebenets, M.A., 2018d. A Comprehensive Multi-Objective Optimization Model for the Vessel Scheduling Problem in Liner Shipping. International Journal of Production Economics 196, 293-318.

Dulebenets, M.A., 2018e. The Vessel Scheduling Problem in a Liner Shipping Route with Heterogeneous Fleet. International Journal of Civil Engineering 16 (1), 19-32.

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.

Dulebenets, M.A., 2020a. An Adaptive Island Evolutionary Algorithm for the berth scheduling problem. Memetic Computing 12 (1), 51-72.

Dulebenets, M.A., 2020b. Archived Elitism in Evolutionary Computation: Towards Improving Solution Quality and Population Diversity. International Journal of Bio-Inspired Computation 15 (3), 135-146.

Dulebenets, M.A., Pasha, J., Kavoosi, M., Abioye, O.F., Ozguven, E.E., Moses, R., Boot, W.R. and Sando, T., 2020. Multiobjective optimization model for emergency evacuation planning in geographical locations with vulnerable population groups. Journal of Management in Engineering 36(2), p.04019043.

Edelkamp, S., Schrodl, S., 2012. Heuristic search - Selective search (chapter 14). Theory and Applications, 633-669.

Edgeworth, F.Y., Mathematical Psychics, P. Keagan, London, England, 1881.

Eiben, A. E., Smith, J.E., 2015. Introduction to Evolutionary Computing. Springer International Publishing.

Fazel Zarandi, M., Khorshidian, H., Akbarpour Shirazi, M., 2016. A constraint programming model for the scheduling of JIT cross-docking systems with preemption. Journal of Intelligent Manufacturing 27, 297-313.

Forouharfard, S., and Zandieh, M., 2010. An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems. The International Journal of Advanced Manufacturing Technology 51, 1179-1193.

Gelareh, S., Goncalves, G., Monemi, R.N., 2013. On Truck dock assignment problem with operational time constraint within cross-docks. https://pdfs.semanticscholar.org/e842/6970cd13bcad7694b5540c61710a1a5d33e6.pdf.

Geoff W., (2017) Supply Chain Management Statistics. Retrieved from: https://explorewms.com/key-supply-chain-management-statistics.html

Glossary of the Material Handling Industry of America (MHIA), 2011.

Glover, F., 1986. Future Paths for Integer Programming and Links to Artificial Intelligence. Computers and Operations Research 13, 533-549.

Goldberg, D.E. and Lingle, R., 1985. Alleles, Loci and the Traveling Salesman Problem. Proceedings of the First International Conference on Genetic Algorithms 154-159.

Golias, M.M., Saharidis, K.D.S., Ivey S., Haralambides H.E., 2013. Advances in Truck Scheduling at a Cross-Dock Facility. International Journal of Information Systems and Supply Chain Management (IJISSCM). 6 (3), 40–62.

Golias, M.M., Stephanie, I., Kaifeng, J., Lipinski, M.E., 2010. A bi-objective model to minimize service and storage time at a cross-dock facility. 51st Annual Transportation Research Forum 2010.

Golias, M.M., Georgios K. D. S., Maria B., Sotirios T., 2012. Scheduling of Inbound Trucks at a Cross-Docking Facility: Bi-Objective VS Bi-Level Modeling Approaches. International Journal of Information Systems and Supply Chain Management (IJISSCM) 5(1), 20–37.

Golshahi-Roudbaneh, A., Hajiaghaei-Keshteli, M., Paydar, M.M., 2017. Developing a Lower Bound and Strong Heuristics for a Truck Scheduling Problem in a Cross-Docking Center. Knowledge-Based Systems 129, 17–38.

Goodarzi, A.H., Nahavandi, N., Zegordi, S.H., 2018. A Multi-Objective Imperialist Competitive Algorithm for Vehicle Routing Problem in Cross-Docking Networks with Time Windows. Journal of Industrial and Systems Engineering 11 (1), 1-23.

Groznika, A., Trkman, P., 2012. Current Issues and Challenges of Supply Chain Management. Economic Research-Ekonomska Istraživanja. 25, 1101–1112.

Gue, K., 2011. Supply Chain News: Where does cross-docking have the best operational fit.

Guignard, M., Hahn, P., Zhang, H. 2014. Dynamic vs. Static Optimization of Cross-docking Operations. Simposio Brasileiro De Pesquisa Operacional 16, 3711-3716.

Handfield, R., Nichols, E., 1999. Introduction to Supply Chain Management.

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

He, Y., Huang, H., Li, D., Shi, C., Wu, S.J. 2018. Quality and Operations Management in Food Supply Chains: A Literature Review. Journal of Food Quality 1–14.

Horn, J., Nafpliotis, N., and Goldberg, D.E. 1994. A Niched Pareto Genetic Algorithm for Multi-Objective Optimization. Proceedings of the 1st IEEE Conference on Computation Evolutionary 1, 82 – 87.

Jayant R., 2016. Supply Chains: Definitions & Basic Concepts. Department of Industrial Engineering, University of Pittsburg.

Jiang, Z., Wang, H., Dulebenets, M.A., & Pasha, J., 2019. Assembly System Configuration Design for Reconfigurability under Uncertain Production Evolution. Journal of Manufacturing Science and Engineering 141 (7), 1-12.

Jones, T., Riley D. W. 1985. "Using Inventory for Competitive Advantage through Supply Chain Management," International Journal of Physical Distribution and Materials Management 15(5), 16-26.

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

Kargari Esfand Abad, H., Vahdani, B., Sharifi, M., Etebari, F., 2018. A bi-objective model for pickup and delivery pollution-routing problem with integration and consolidation shipments in cross-docking system. Journal of Cleaner Production 193, 784–801.

Kavoosi, M., Dulebenets, M.A., Abioye, O.F., Pasha, J., Wang, H., & Chi, H., 2019. An Augmented Self-Adaptive Parameter Control in Evolutionary Computation: A Case Study for the Berth Scheduling Problem. Advanced Engineering Informatics 42, p. 100972.

Kavoosi, M., Dulebenets, M.A., Abioye, O.F., Pasha, J., Theophilus, O., Wang, H., Kampmann, R., & Mikijeljevic, M., 2020. Berth Scheduling at Marine Container Terminals: A Universal Island-based Metaheuristic Approach. Maritime Business Review 5 (1), 30-66.

Khalili-Damghani, K., Tavana, M., Santos-Arteaga, F.J., Ghanbarzad-Dashti, M. A. (2017) A customized genetic algorithm for solving multi-period cross-dock truck scheduling problems. Measurement 108, 101–118.

Khosla S., (2014) World Export Map. GlobalPost.

Kim S., Pasupathy R., Henderson S.G. (2015) A Guide to Sample Average Approximation. In: Fu M. (eds) Handbook of Simulation Optimization. International Series in Operations Research & Management Science, vol 216. Springer, New York, NY.

Kirkpatrick, S., Gelatt, J., Vecchi, M.P., 1983. Optimization by simulated annealing. Science 220(4598), 671-680.

Knowles, J. D., Corne, D.W., 2000. Approximating the nondominated front using the Pareto Archived Evolution Strategy. Evolutionary Computing 8(2), 149-172.

Konur, D., Golias, M.M., 2013a. Analysis of Different Approaches to Cross-Dock Truck Scheduling with Truck Arrival Time Uncertainty. Computers & Industrial Engineering 65(4), 663–72.

Konur, D., Golias, M.M., 2013b. Cost-Stable Truck Scheduling at a Cross-Dock Facility with Unknown Truck Arrivals: A Meta-Heuristic Approach. Transportation Research Part E: Logistics and Transportation Review 49 (1), 71–91.

Kozlenkova, I. V., Hult, G., Thomas M., Lund, D. J., Mena, J. A., Kekec, P., 2015. The Role of Marketing Channels in Supply Chain Management. Journal of Retailing 91(4), 586-609.

Kulwiec, R., 2004. Cross-docking as a Supply Chain Strategy 20, 8.

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

Ladier, A., Alpan, G. 2016. Cross-Docking Operations: Current Research versus Industry Practice. Omega 62, 145–62.

Ladier, A., Alpan, G. 2018. Cross-dock truck scheduling with time windows: earliness, tardiness, and storage policies. Journal of Intelligent Manufacturing 29, 569-583.

Ladier, A., Alpan, G. 2016. Robust Cross-Dock Scheduling with Time Windows. Computers & Industrial Engineering 99, 16–28.

Ley, S., Elfayoumy, S. 2007. Cross-Dock Scheduling Using Genetic Algorithms. IEEE International Symposium on Computational Intelligence in Robotics and Automation. Jacksonville, FL, USA. 416–20.

Li, J., Cai, X., Zeng, Y., 2016. Cost allocation for less-than-truckload collaboration among perishable product retailers. OR Spectrum 38, 81-117.

Li, Y., Lim A., Rodrigues, B. 2004. Crossdocking: JIT Scheduling with Time Windows. Journal of Operation Research Society 55, 1342–51.

Li, Z., Sim, C.H., Low, M.Y.H., Lim, Y.G. (2008) Optimal product allocation for cross-docking and warehousing operations in FMCG supply chain, in Service Operations and Logistics, and Informatics. IEEE International Conference on Service Operations and Logistics, and Informatics 2963–2968.

Liao, T.W., Egbelu, P.J., Chang, P.C. 2012. Two-hybrid differential evolution algorithms for optimal inbound and outbound truck sequencing in cross-docking operations. Applied Soft Computing 12, 3683-3697.

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, 212-229.

Lim, A., Miao, Z., Rodrigues, B., Xu, Z., 2005. Transshipment through cross-docks with inventory and time windows. Naval Research Logistics 52, 724–733.

Luo, C., Ting, C. 2017. An Ant Colony Optimization for the Multi-Dock Truck Scheduling Problem with Cross-Docking. International Journal of Operation research 14(3), 77-87.

Madani-Isfahani, M., Tavakkoli-Moghaddam, R., Naderi, B. (2014) Multiple cross-docks scheduling using two meta-heuristic algorithms. Computers & Industrial Engineering 74, 129–138.

Maknoon, M.Y., Baptiste P. 2009. Cross-Docking: Increasing Platform Efficiency by Sequencing Incoming and Outgoing Semi-Trailers. International Journal of Logistics Research and Applications 12 (4), 249–61.

Margolis, J., Sullivan, K., Mason, S., Magagnotti, M. 2018. A multi-objective optimization model for designing resilient supply chain networks. International Journal of Production Economics – in press.

McWilliams, D.L. 2009. Genetic-Based Scheduling to Solve the Parcel Hub Scheduling Problem. Computers & Industrial Engineering 56 (4), 1607–16.

McWilliams, D.L., Stanfield, P.M., Geiger, C.D. 2005. The parcel hub scheduling problem: A simulation-based solution approach. Computers & Industrial Engineering 49, 393–412.

Mentzer, J., DeWitt, W., Keebler, J.S., Nix, N.W., Smith, C.D. 2001. Journal of Business Logistics 22(2).

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

Mladenović, N., Hansen, P., 1997. Variable neighborhood search. Computers & Operations Research 24(11), 1097-1100.

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

Montana, D., 1998. Introduction to the special issue: Evolutionary algorithms for scheduling. MIT Press.

Morais, V.W.C., Mateus, G.R., Noronha, T.F., 2014. Iterated local search heuristics for Vehicle Routing Problem with Cross-Docking. Expert Systems with Applications 41, 7495-7506.

Murray B., 2019. Warehouse Logistics: Cross-docking. Retrieved from: https://www.thebalancesmb.com/cross-docking-in-the-warehouse-2221394

Naderi, B., Rahmani, S., Rahmani, S., 2014. A Multiobjective Iterated Greedy Algorithm for Truck Scheduling in Cross-Dock Problems. Journal of Industrial Engineering 1–12.

Naskaris, S., Pudukadan, R., Smith, K., Maher, M., 2014. Cost Analysis for a Cross-Dock Alternative to Supply Local Produce from Food Hub to Grocery Store via a Large Regional

Distribution Center. Center for Environmental Farming Systems, North Carolina State University, Raleigh, NC.

Nitasha, S. Tapas, K., 2014. Study of Various Mutation Operators in Genetic Algorithm. International Journal of Computer Science and Information Technologies 5(3), 4519-4521.

Oliver K., Webber, M., 1982. Supply Chain Management: logistics catches up with strategy.

Pal, A., Kant, K. 2017. A Food Transportation Framework for an Efficient and Worker-Friendly Fresh Food Physical Internet. Logistics 1, 10.

Pareto, V., Manuale di Economia Politica, Societa Editrice Libraria, Milano, Italy, 1906. Translated into English by A.S. Schwier as Manual of Political Economy, Macmillan, New York, 1971.

Pasha, J., Dulebenets, M.A., Kavoosi, M., Abioye, O.F., Theophilus, O., Wang, H., Kampmann, R., & Guo, W., 2020a. Holistic Tactical-Level Planning in Liner Shipping: An Exact Optimization Approach. Journal of Shipping and Trade 5, 1-35.

Pasha, J., Dulebenets, M.A., Kavoosi, M., Abioye, O.F., Wang, H., & Guo, W., 2020b. An Optimization Model and Solution Algorithms for the Vehicle Routing Problem with a "Factory-in-a-Box". IEEE Access 8, 134743-134763.

Patil, M. 2015. Challenges for Supply Chain Management in Today's Global Competitive Environment. European Journal of Business Management 4.

Pawar, U., Hanchate, D., 2014. Personnel Scheduling with Heuristic Search Approach. International Journal of Engineering Sciences & Research Technology 3, 88-94

Pinchera, D., Stefano Perna, S., and Migliore, M.D., A Lexicographic Approach for Multi-Objective Optimization in Antenna Array Design. Progress in Electromagnetic Research 59, 85-102.

Quiceno, G., Álvarez, C., Ávila, R., Fernández, Ó., Franco, C., Kunc, M., and Dyner, I., 2019. Scenario analysis for strategy design: A case study of the Colombian electricity industry. Energy Strategy Reviews 23, 57-68.

Raith, A., Schmidt, M., Schöbel, A., and Thom, L., 2018. Multi-objective minmax robust combinatorial optimization with cardinality-constrained uncertainty. European Journal of Operational Research 267(2), 628-642.

Rajabi, M., Shirazi, M.A., 2016. Truck scheduling in a cross-dock system with multiple doors and uncertainty in availability of trucks. Journal of Applied Environmental and Biological Sciences 6, 101–109.

Rob, 2012. Supply Chain Management, Cross-docking.

Ross, A., Jayaraman, V., 2008. An evaluation of new heuristics for the location of cross-docks distribution centers in supply chain network design. Computers & Industrial Engineering 55, 64–79.

Sathasivan, K., 2011. Optimizing Cross-dock Operations under Uncertainty. University of Texas at Austin.

Schwerdfeger, S., Boysen, N., Briskorn, D., 2018. Just-in-time logistics for far-distant suppliers: scheduling truck departures from an intermediate cross-docking terminal. OR Spectrum 40(1), 1-21.

SCOR., 2012. Supply Chain Council, Inc. SCOR: The Supply Chain Reference.

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

Shakeri, M., Low, M.Y.H.., Li, Z., 2008. A generic model for cross-dock truck scheduling and truck-to-door assignment problems. 6th IEEE International Conference on Industrial Informatics 857–864.

Sheikholeslam, M.N., Emamian, S., 2016 A Review and Classification of Cross-docking Concept 10.

Shiekholeslam, M. N., Emamian, S., 2016. A Review and Classification of Cross-Docking Concept. International Journal of Learning Management Systems 4(1), 25-33.

Shiguemoto, A.L., Netto, U.S.C., Bauab, G.H.S., 2014. An efficient hybrid meta-heuristic for a cross-docking system with temporary storage. International Journal of Production Research 52(4), 1231-1239.

Shuib, A., Fatthi, W. N. A. W. A., 2012. A review on Quantitative approach for Dock Door Assignment in Cross-docking. International Journal on Advanced Science, Engineering and Information Technology 2(5).

Sivanandam, S., Deepa, S., 2008. Introduction to Genetic Algorithms, Springer International Publishing: Berlin, Germany.

Skinner B., 2006. Systems and Operations Management. Retrieved from: https://slideplayer.com/slide/4273206/ MBA 8452 Systems and Operations Management.

Soltani, R., Sadjadi, S.J. 2010. Scheduling trucks in cross-docking systems: A robust meta-heuristics approach. Transportation Research Part E: Logistics and Transportation Review 46, 650–666.

Song, K., Chen, F., 2007. Scheduling cross-docking logistics optimization problem with multiple inbound vehicles and one outbound vehicle, in Automation and Logistics. IEEE International Conference on Automation and Logistics 3089–3094.

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.

Srinivas, N., and Deb, K., 1994. Multi-objective Optimization Using Nondominated Sorting in Genetic Algorithms," in Evolutionary Computation 2(3), 221-248.

Stalk, G., Evans, P., Sgulman, L. E., 1992. Competing on capabilities: the new rules of corporate strategy. Harvard Business Review 63.

Stevens, Graham C., 1989. "Integrating the Supply Chains," International Journal of Physical Distribution and Materials Management 8(8), 3-8.

Tang, S.L., Yan, H., 2010. Pre-distribution vs. post-distribution for cross-docking with transshipments. Omega 38, 192–202.

Thapa, G.B., Dhamala, T.N., Pant, S.R., 1970. Cross-docking Operations for Supply Chain Logistics in JIT Production and Distribution Systems. Journal of Industrial Engineering 8, 219–230.

Theophilus, O., Dulebenets, M.A., Pasha, J., Abioye, O.F. and Kavoosi, M., 2019. Truck Scheduling at Cross-Docking Terminals: A Follow-Up State-Of-The-Art Review. Sustainability 11(19), p.5245.

Theophilus, O., Dulebenets, M.A., Pasha, J., Lau, Y.Y., Fathollahi-Fard, A.M., & Mazaheri, A., 2021 Truck Scheduling Optimization at a Cold-Chain Cross-Docking Terminal with Product Perishability Considerations – under review.

Tsui, L.Y., and Chang, C.-H. 1990. A microcomputer-based decision support tool for assigning dock doors in freight yards. Computers & Industrial Engineering 19(1-4), 309-312.

Vahdani, B., Soltani, R., Zandieh, M. 2010. Scheduling the truck holdover recurrent dock cross-dock problem using robust meta-heuristics. The International Journal of Advanced Manufacturing Technology 46, 769–783.

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

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

Vis, I.F.A., Roodbergen, K.J., 2008. Positioning of goods in a cross-docking environment. Computers & Industrial Engineering 54, 677–689.

Vogt, J., 2010. The successful cross-dock based supply chain. Journal of Business Logistics 31, 99–119.

Walha, F., Chaabane, S., Bekrar, A., Loukil, T., 2014. The cross-docking under uncertainty: State of the art. International Conference on Advanced Logistics and Transport (ICALT) 330-335.

Wang, J., Ranganathan Jagannathan, A.K., Zuo, X., Murray, C.C., 2017. Two-layer simulated annealing and tabu search heuristics for a vehicle routing problem with cross-docks and split deliveries. Computers & Industrial Engineering 112, 84–98.

Wang, S., Meng, Q., Liu, Z., 2013. Bunker consumption optimization methods in shipping: a critical review and extensions. Transport Research Part E 53, 49-62.

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

Wang, Z., Lu, Y., Zhao, L., Cao, N., 2018. Improved Harmony Search Algorithm for Truck Scheduling Problem in Multiple-Door Cross-Docking Systems. Discrete Dynamics in Nature and Society 1–15.

Wen, M., Larsen, J., Clausen, J., Cordeau, J-F, Larporte, G. 2009. Vehicle routing with cross-docking. Journal of Operation Research Society 60(12), 1708-1718.

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

Xin, J., Negenborn, R.R., Lin, X., 2017 Piecewise affine approximations for quality modeling and control of perishable foods: A demonstration of the class file. Optimal Control Applications and Methods 39, 860–872.

Yan, W. 2014. Heuristics for Truck scheduling at Cross-docking Terminals. Concordia University Montreal, Quebec, Canada.

Yang, S., Xiao, Y., Kuo, Y.-H., 2017. The Supply Chain Design for Perishable Food with Stochastic Demand. Sustainability 9, 1195.

Yazdani, M., Naderi, B., Mousakhani, M., 2015. A Model and Metaheuristic for Truck Scheduling in Multi-door Cross-dock Problems. Intelligent Automation & Soft Computing 21 (4), 633-644.

Yu, W., 2002. Operational strategies for cross-docking systems Iowa State University, Iowa.

Yu, W., Egbelu, P.J., 2008. Scheduling of inbound and outbound trucks in cross-docking systems with temporary storage. European Journal of Operational Research 184, 377–396.

BIOGRAPHICAL SKETCH

Oluwatosin Theophilus is a Doctoral Candidate and a Graduate Research Assistant in the Department of Civil & Environmental Engineering at FAMU-FSU College of Engineering. He holds a baccalaureate degree in Transport Management Technology from the Federal University of Technology Akure (FUTA), Nigeria. He also holds an M.S. degree in civil engineering from Florida State University. His research interests include transport engineering, optimization, mathematical programming, traffic simulation and modeling, and evolutionary computation. His doctoral dissertation which is part of the requirement for his doctorate degree at FAMU-FSU College of Engineering is focused on the development of algorithms for scheduling at distribution facilities.