MULTI-OBJECTIVE SHIPMENT CONSOLIDATION AND DISPATCHING PROBLEM

ÖZGE BÜYÜKDEVECİ

Master's Thesis

Graduate School

Izmir University of Economics

Izmir

2022

MULTI-OBJECTIVE SHIPMENT CONSOLIDATION AND DISPATCHING PROBLEM

ÖZGE BÜYÜKDEVECİ

A Thesis Submitted to
The Graduate School of Izmir University of Economics
Master Program in Industrial Engineering

Izmir
2022

ABSTRACT

MULTI-OBJECTIVE SHIPMENT CONSOLIDATION AND DISPATCHING PROBLEM

Büyükdeveci, Özge

Master Program in Industrial Engineering

Advisor: Prof. Dr. Selin Özpeynirci
Co-Advisor: Assoc. Prof. Dr. Özgür Özpeynirci

June, 2022

In recent years, with the increase in global production and demand, transportation problems have become a widely studied area to provide high-quality service at the lowest cost. In this thesis, a bi-objective shipment consolidation and dispatching problem is considered where one of the objectives is to minimize the total cost and the other is to minimize the total distance. In order to create a non-dominated solution set, a multi-objective mixed integer linear programming model is developed and the augmented-ε constraint method is used to generate the efficient frontier. However, since this approach is not capable of finding the non-dominated solution set in a reasonable time even for small-sized instances, we propose a multi-objective variable neighborhood search heuristic. To measure the performance of the proposed approach, a computational experiment is conducted on randomly generated instances available in the literature. The experimental results indicate that the multi-objective variable neighborhood search heuristic performs efficiently in reasonable time.

Keywords: Shipment Consolidation, Multi-objective Decision Making, Augmented ε-Constraint Method, Variable Neighborhood Search.

ÖZET

ÇOK AMAÇLI YÜK BİRLEŞTİRME VE SEVKİYAT PROBLEMİ

Büyükdeveci, Özge

Endüstri Mühendisliği Yüksek Lisans Programı

Tez Danışmanı: Prof. Dr. Selin Özpeynirci
İkinci Tez Danışmanı: Doç. Dr. Özgür Özpeynirci

Haziran, 2022

Son yıllarda, küresel üretim ve talebin artmasıyla birlikte, ulaşım sorunları, yüksek kaliteli hizmeti en düşük maliyetle sunmak için yaygın olarak çalışılan bir alan haline gelmiştir. Bu tezde, amaçlarından birinin toplam maliyeti en aza indirmek ve diğerinin toplam mesafeyi en aza indirmek olduğu iki amaçlı bir yük birleştirme ve sevkiyat problemi ele alınmıştır. Domine edilemeyen sonuçlar kümesi oluşturmak için, çok amaçlı karma tamsayılı doğrusal programlama modeli önerilmiş ve etkin sınırı oluşturmak için modifiye edilmiş ε-kısıt yöntemi kullanılmıştır. Ancak bu yaklaşım, küçük boyutlu örnekler için bile makul bir sürede bir domine edilemeyen sonuçlar kümesi bulamadığı için, çok amaçlı değişken komşuluk arama sezgisel yöntemi önerilmiştir. Önerilen yaklaşımın performansını ölçmek için literatürde yer alan rassal olarak üretilmiş örnekler ile bir hesaplamalı deney gerçekleştirilmiştir. Deneysel sonuçlar, çok amaçlı değişken komşuluk arama sezgisel yönteminin verimli bir şekilde çalıştığını ve hesaplama süresinin makul olduğunu göstermektedir.

Anahtar Kelimeler: Yük Birleştirme, Çok Amaçlı Karar Verme, Modifiye Edilmiş ε-kısıtı yöntemi, Değişken Komşuluk Arama.

ACKNOWLEDGEMENTS

Above all, I would like to express my sincere gratitude to my academic advisers, Prof. Dr. Selin Özpeynirci and Assoc. Prof. Dr. Özgür Özpeynirci for their invaluable guidance, advice, and encouragement. I feel extremely lucky to have them as my thesis advisors. I hope we will be able to work together in the future.

My deepest thanks to my husband, my best friend, Onur Büyükdeveci. He was my greatest support and was always there for me. I felt his love and encouragement all the time. Without a doubt, this process would have been extremely difficult for me without him.

I would like to thank my role model, my mother, Ayşegül Kubilay, who loves me unconditionally in every stage of my life; my father, Ercan Kubilay, who always encourages me to be better; and my lovely sister, Elif Kubilay, who is my source of joy.

I am very grateful to Sude Pehlivan, Serdar Babaç, Mesut Abedinifar, Serdar Hakan Argüz, Dünya Kulavuz, Barış Dal, and Mehmet Faruk Cengiz, who have given me their friendship, put up with my odd hours, and their never-ending support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

Due to the increase in global production and demand in recent years, long-haul freight transportation operations have intensified. Consequently, international transportation has an increasing need for improvements in consolidation systems and transportation operations. Considering the high logistics costs, the fact that the carriers aim to achieve the highest quality service with the minimum cost has made transportation a subject that has been extensively studied by researchers.

The daily operation plans of the freight forwarders aim to deliver different types of demands of their customers within certain time-windows while minimizing the total cost. Moreover, they can often use transshipment terminals to save time and money. In real life, carriers often make their daily operation plans manually. This can cause them to implement dispatching plans that are not the optimal solution. Also, the planning process might take several hours. By using analytical solution approaches, we aim to find the most efficient solutions for freight forwarders. With this objective, we examine a problem faced by carriers in real life, namely the Shipment Consolidation and Dispatching Problem (SCDP) under some presumptions. The order information such as volume, weight, length, destination point, release date, and deadline is known to be deterministic. Secondly, the cost is computed according to the farthest delivery point from the depot. This is a common practice in real life that eases the planning process, also Koca and Yıldırım (2012) described routes characterized by their location with the greatest distance and the fixed cost associated with that location. There is a fixed number of stops allowed to be charged as a fixed cost and a maximum number of extra stops. An additional cost is charged to the vehicle's overall cost for each extra stop. Thus, the cost for each vehicle is calculated. Finally, two options for the delivery structure are examined. The orders could be transferred directly to their delivery point or through a transshipment terminal. Instead of being delivered to their final address, the orders are delivered at the transshipment terminal at a certain cost. Figure 1 depicts an illustrative example of a hybrid system with direct delivery and delivery from a transshipment terminal. As seen in Figure 1, the routes are open, i.e., return of the vehicles to the depot is not included in the problem.

Both SCDP and network design problems deal with freight processes of consolidation (Crainic, 2000). However, there are some distinctions between the two problems, such

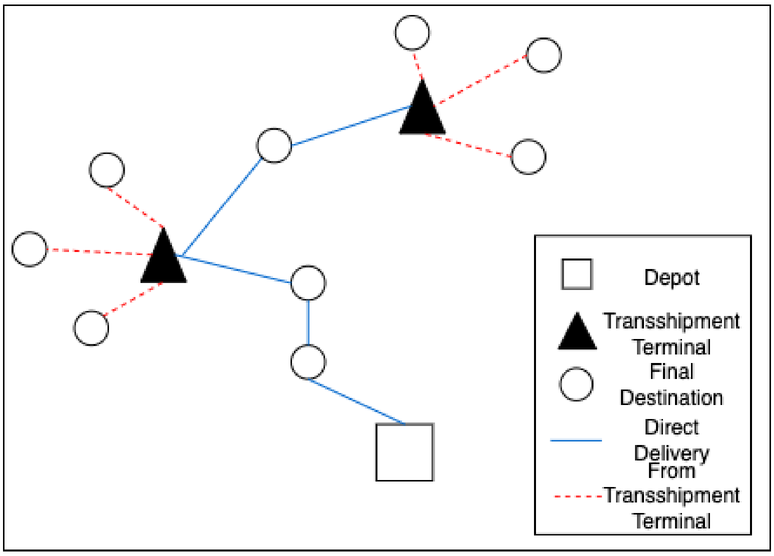

Figure 1. Delivery System

as cost structures. For network design problems, the total cost is computed from node to hub or hub to node, but for SCDP, the cost is calculated for each route. Also, network problems use the direct delivery and transshipment terminals rarely. Instead, a network is designed using many intermediate facilities (Guastaroba et al., 2016). Another transportation planning problem, the Vehicle Routing Problem (VRP), has similarities with the SCDP. Both SCDP and VRP extensions can optimize vehicle routes while taking capacity into account, as well as provide solutions with time-windows and different delivery options. On the other hand, the cost structures of VRP and SCDP are different. While the cost scheme of VRP is varied according to the distance between the locations visited, SCDP uses a fixed cost for each route according to the farthest stop from the depot and an additional charge for extra stops. In this regard, the problem is not modeled as a VRP extension.

The aim of this thesis is to provide efficient solutions to the introduced SCDP with two separate goals; the overall cost and the overall distance minimization. The cost structure of the problem is generated according to the real-life business environment. Vehicle returns to the depot are not considered for freight forwarders because they only rent vehicles for one-way trips. Additionally, long-haul transportation trips are typically much longer than short-haul transportation, thus the cost structure of a route does not need to be as accurate as it would be in short-haul vehicle routing. Moreover, the dynamic real-life cases require fast decision-making, so freight forwarders prefer to use simple procedures to support easy cost calculations with minimum data requirements. This means that the main route of a vehicle is defined by its farthest stop, and additional stops along the same route can be allowed. Even if there is an additional stop in the route, it adds to travel time and might cause delivery to be delayed. As a result, fewer stops are preferable, and only a limited number of extra stops are permitted. An additional cost must be paid for each stop that exceeds that number. Consequently, the total cost for each route is calculated by summing the fixed cost calculated depending on the farthest stop from the depot, the additional cost for each extra stop, and the transshipment terminal usage cost. Considering this particular cost structure, it is understood that the total cost to be calculated is not directly related to the total distance. In this manner, the two objective functions, which are total cost and total cost, conflict. For example, while the total cost of the route created by delivering the orders whose delivery point is close to the depot directly to their own destinations is a minimum, it can be predicted that the total distance will be smaller if the same orders are delivered from the same transshipment terminal.

An exact solution approach is developed to create a Pareto optimal set for the instances that need less amount of order to be delivered. Since this exact solution method is not capable of generating the Pareto set in an acceptable time for large-sized problems, a heuristic approach is proposed. The multi-objective Variable Neighborhood Search (VNS) heuristic is developed to attain approximate points to the optimal in short computational times. We conduct computational test to asses the performance of the proposed approaches and discuss the experimental results.

The thesis is organized as follows: In Chapter 2, we review the related studies in the literature in four sections; (*i*) SCDP and its variations, (*ii*) multi-objective decision making, (*iii*) VNS algorithms and (*iv*) multi-objective VNS. Additionally, we highlight our contribution to the literature. In Chapter 3, we present the problem definition, mathematical model formulation, the methods for generating the efficient frontier, and the VNS algorithm. In Chapter 4, we report the results of the multi-objective mixed integer linear programming model (MOMILP) and the multi-objective VNS algorithm. Finally, in Chapter 5 we present a general review and discussion of the study, as well as some potential future research areas.

CHAPTER 2 : LITERATURE REVIEW

In this chapter, we examine the related literature to our problem and the relevant solution methods. We present the literature in four sections. In the first section, we examine the SCDP and its different versions. In the second section, we discuss the studies on multi-objective decision making. Next, we give a review of corresponding studies on VNS. In the last section, we examine the multi-objective VNS in the literature. Finally, we discuss our contribution to the literature.

2.1 *Shipment Consolidation and Dispatching Problems*

Transportation problems are divided into three classes: strategic, tactical, and operational planning problems (Crainic and Laporte, 1997; Schmidt and Wilhelm, 2000). Long-term decisions or decisions that need big investments, such as facility location issues, are related to strategic decision problems. Tactical decision problems consider medium-term decisions such as network design problems. Operational planning problems concern short-term decisions. SCDP is an operational planning problem, since it considers short-term decisions.

Shipment consolidation is a crucial logistics application that merges many small-sized orders into a single big load, lowering the shipper's total transportation costs (Higginson and Bookbinder, 1994). Because the consolidation problem is so diverse and complicated, it is critical to choose the correct consolidation technique (Min, 1996). Brennan (1982) classified the consolidation strategies into three groups: spatial, temporal, and product consolidation strategies. We discuss only spatial and temporal consolidation since the literature on product consolidation is very limited.

The spatial consolidation strategy is related to geographical decisions. The strategy consolidates the orders according to their geographic features. Thus, it creates routes for the consolidated orders. We can say that the main objective of spatial consolidation is similar to that of the shortest-path problem (Min and Cooper, 1990). So the studies that use this spatial consolidation strategy focus on the cost-effectiveness of the consolidation. Furthermore, they discuss the advantages of consolidation compared with direct deliveries. Generally, early studies only consider the spatial strategy (Daganzo, 1988; Campbell, 1990; Min, 1996). More recent studies consider the problem with both temporal and spatial strategies.

On the other hand, the main focus of the temporal consolidation strategy is time. While considering the release and due date of the orders, the aim is to determine the best departure time for the consolidated orders. When determining this appropriate time, two main questions are important; i) when the vehicle must leave the depot to meet all orders on time, and ii) how large should the shipment quantity be (Cetinkaya and Lee, 2000). To apply the temporal strategy, some studies use simulation models (Masters, 1980; Higginson and Bookbinder, 1994). Others prefered to use analytical models, such as dynamic programming and Markov decision model (Powell, 1985; Gupta and Bagchi, 1987; Minkoff, 1993).

In recent years, studies have considered both spatial and temporal strategies. The approaches were applied not only to SCDP, but also to network design problems and vehicle routing problems as operational planning problems. Attanasio et al. (2007) study a SCDP, which aims to determine on a regular basis the optimal route to deliver a group of orders across a multi-day planning horizon. In their study, they create an algorithm which is a cutting plane framework that solves a simplified Integer Linear Program (ILP). Computational findings reveal that their method allows them to save the cost. Ülkü (2012) examine SCDP with the goal of reducing carbon and energy waste. They develop an optimization model to maximize savings for the environment. Another related research is Yücel et al. (2022), where they study a vehicle loading and dispatching problem. They provide a Mixed Integer Linear Program (MILP), and for large-sized instances, they develop a heuristic method which is an extension of Large Neighborhood Search. The results found using real-world data are presented to show the proposed algorithm's efficiency.

Tokcaer (2018) propose two variations of SCDP based on different assumptions. In the first problem, the cost of each route is determined by annual contracts. For the second one, each route has a fixed cost, and the total cost is determined by adding the charge for the extra stops. They first propose a mathematical model and then different heuristic methods to find a solution for the problems, with the minimization of the total cost as an objective function. Then, they analyze their methods by testing them with real-world data. The computational results are demonstrated to show the provided savings. Our study considers a similar case with the second problem of Tokcaer (2018). The main difference is that in our study, solution methods are developed for two objectives, not for a single objective. One of these objectives is to minimize the total cost and the other one is to minimize the total distance. Additionally, we do not use fixed routes for the objective of minimizing the total distance. Instead, our solution methods generate the possible feasible routes themselves. Since we handle the multi-objective optimization problem, the solution strategies utilized in the literature are explored in the following section.

2.2 *Multi-Objective Decision-Making*

Multi-objective optimization involves optimizing many competing objective functions at the same time. We cannot find a single optimal solution to these problems since the objective functions conflict with each other. In other words, improving one of the objective functions may worsen the other ones. So, the decision maker seeks the most preferred solution or the most efficient solution instead of the optimal solution. As a result, the idea of optimality is altered by Pareto optimality (Steuer, 1986). Recently, multi-objective decision making has become one of the popular study fields in operation research. It combines different study areas such as mathematics, software engineering, and decision support systems (Köksalan and Wallenius, 2012).

Transportation problems have also been frequently addressed as multi-objective in the literature. Conflicting objectives make the problems more complex. Nevertheless, there are some commonly used exact solution techniques to generate the Pareto optimal solutions to solve problems with multi-objective. Hwang and Masud (2012) introduced these solution methods by dividing them into three classes: the priori methods, the interactive methods, and the posteriori (generation) methods. The definition of the preference of the decision maker is the first step in the priori procedures. It is mostly defined by the weights of the objective functions. Then, according to these weighting combinations, the objective functions are optimized. The disadvantage of this strategy is that determining the weights according to the preferences is not an easy task. On the other hand, for interactive methods, the decision maker is active throughout the entire decision making procedure. After finding each new solution, the decision maker is asked to make his/her choice. However, in this technique, only some of the non-dominated points can be found. The best solution is selected among the solutions that have been obtained so far. The decision maker may not even be aware of most of the non-dominated solutions. Finally, the generation methods start with obtaining the Pareto optimal set. After that, the decision maker selects a solution in the obtained Pareto solution set. The biggest disadvantage of this method is that it requires significant computing effort. On the other hand, the decision maker has much higher confidence as they have all the efficient solutions. One of the most widely used posteriori methods is ε-constraint algorithm. With Mavrotas (2009) introducing the augmented ε-constraint algorithm, it has become a highly preferred solution method, which is an advanced version of the ε-constraint approach. Since it avoids getting weak Pareto-optimal solutions, this approach is utilised for many different optimization problems, such as location-routing problems (Yu and Solvang, 2016), staff scheduling problems (Sadjadi et al., 2014), supply chain network problems (Resat and Unsal, 2019) and traveling salesperson problems (Bouziaren and Aghezzaf, 2018). Azadnia et al. (2015) applied both the weighted sum algorithm and the augmented ε-constraint algorithm to solve a multi-objective lot sizing problem. In their study, they emphasized that their results show that the augmented ε-constraint algorithm performs better. Zhu and Zhu (2020) investigate the service network problem with four objectives. To generate the non-dominated solutions, they use the augmented ε-constraint method. Tamby and Vanderpooten (2021) use an extended version of the ε-constraint approach for problems with two or more objectives. Their algorithms perform much better than the previous algorithms for the discrete optimization problem instances. Zhu (2022) also prefer to obtain the non-dominated solutions for multi-objective route planning problems using the augmented ε-constraint approach.

Another important point is the decision-maker's choice of which solution to choose after obtaining the Pareto-optimal set. There are various multi-criteria decision support methods. We do not investigate this subject in the scope of our study. Our objective is to use posteriori methods to obtain a Pareto optimal solution. The decision maker's selection can be examined in future studies.

As we mentioned earlier, considering multiple objectives has made our NP-hard problem even more difficult to solve. Therefore, a heuristic method is used for large-sized instances. In the next section, studies on this specific method are reviewed.

2.3 *Variable Neighborhood Search*

For most large-scale real-life problems, the mathematical model cannot produce a solution of the desired quality in an acceptable time. For this reason, heuristic methods have been developed which could find good solutions in a shorter time, although they do not promise to find an optimal solution. In our study, we apply variable neighborhood search, a popular heuristic, to solve the problem. The variable neighborhood search heuristic was first presented by Hansen and Mladenović (1997), and its application areas and methods have been rapidly developed since then. In this heuristic method, a local minimum is found by systematically changing the used neighborhoods, and the local optimum is avoided by the perturbation step (Hansen and Mladenović, 1997). The effectiveness of the variable neighborhood search algorithm for different optimization problems, such as knapsack and packing problem (Puchinger and Raidl, 2008), job shop scheduling (Zobolas et al., 2009), pharmacy duty scheduling (Kocatürk and Özpeynirci, 2014) and travelling salesperson problem (Hu and Raidl, 2008) has been demonstrated. Furthermore, for some problems, basic VNS may give insufficient results. For these cases, advanced versions of VNS have been introduced in the literature. For example, to solve very large-sized instances, reduced variable neighborhood search is introduced. The main difference from basic VNS is that local search is not implemented. Considering that the local search is the part that needs the most time in the whole procedure, it aims to save computational time. Another example of a version of VNS is Variable Neighborhood Descent. In this method, the changing of neighborhoods is done in a deterministic manner. The other variants and the applications of the VNS are explained in detail in Hansen et al. (2010); Hansen et al. (2017).

When we examine the applications of the VNS algorithm for transportation problems in the literature, we see that there are many studies that give very successful results. Hemmelmayr et al. (2012) solve the variable-sized bin packing problem using a VNS heuristic. A hybrid meta-heuristic is developed with the use of lower bound techniques for VNS. Their experimental results show that the suggested method is quite competitive with the current studies in the literature. Sadati and Çatay (2021) study a Multi-Depot Green Vehicle Routing Problem. To solve the problem, they use a hybrid VNS algorithm by combining the general VNS with Tabu Search (TS). They use the Green Vehicle Routing Problem data set from the literature to measure the performance of the technique. The results show that the algorithm gives very effective solutions in a short time. Tokcaer (2018) use VNS to solve a different version of SCDP. The comparison of the exact approach and VNS results indicates that VNS gives approximate results in a much shorter computation time compared to the exact solution method. So far, we have discussed single-objective studies for VNS. The studies of VNS for multi-objective are discussed in the next section.

2.4 *Multi-Objective Variable Neighborhood Search*

The success of metaheuristics in solving single-objective problems has led researchers to develop metaheuristics for solving multi-objective combinatorial problems. Pareto simulated annealing (Czyzżak and Jaszkiewicz, 1998), tabu search algorithm (Alfieri et al., 2020) and the ant colony algorithm (Rivera et al., 2022) are a few examples. The multi-objective VNS is one of the successful metaheuristics. The multi-objective VNS heuristic was first proposed by Geiger (2006). The purpose of the study is to solve the flow shop scheduling problem. It is emphasized that determining the ideal neighborhood structure is one of the most important decisions for the effectiveness of the algorithm. The problem is also solved with a single neighborhood operator. The computational results demonstrate that a single neighborhood is not enough to obtain all efficient frontier points. But VNS provides major improvements on the solution.

Population-based metaheuristics are generally preferred for use on multi-objective problems rather than trajectory-based metaheuristics. Genetic algorithms, harmony search, and differential evolution are examples of population-based algorithms, while ant colony optimization, tabu search, firefly algorithm, and variable neighborhood search are examples of trajectory-based heuristics. The main reason for this is that population-based metaheuristics perform with a set of solutions, that is generally called a population. Meanwhile, trajectory-based metaheuristics perform with a single solution. Using a set of solutions to obtain the Pareto frontier is an advantage. In order to use this advantage in trajectory-based metaheuristics, Duarte et al. (2015) propose using the approximate set of efficient solutions found during the algorithm process as an incumbent solution to multi-objective problems. This approach is also useful for applying other trajectory-based metaheuristics to multi-objective problems. Ripon et al. (2013) present adaptive VNS to handle multi-objective facility layout problems. They compare their algorithm results with genetic algorithm-based approaches. The results show that the proposed algorithm could find more near-optimal solutions. Soylu and Katip (2019) investigate a bi-objective multiple allocation p-hub median problem. While the minimization of the total transportation cost is assumed to be the first objective function, the second one is the minimization of the multiple-transit routes to increase customer satisfaction. They fİnd the exact solutions for small and medium-scaled instances. They present the VNS algorithm to solve large-sized instances. Mrkela and Stanimirović (2022) consider a multi-objective maximal covering location problem. They propose three different multi-objective VNS variants. The results of the applications are compared with the multi-objective evolutionary approaches. The multi-objective VNS implementations give more efficient solutions, especially for large-sized instances. Xu et al. (2021) propose multi-objective VNS to solve the colored traveling salesman problem, that is a specific type of traveling salesman problem. Their method is compared with four existing algorithms, which are two GA and two VNS. The experimental results indicate that their multi-objective VNS performs more efficiently compared with the other existing methods. Özpeynirci et al. (2022) use a mathematical model to generate an efficient frontier for the multi-objective portfolio selection problem. Furthermore, a VNS algorithm for the large-scaled instances of the problem is developed to obtain an approximate Pareto frontier. They utilize various performance measures to evaluate the efficiency of the suggested algorithm, where the results indicate that the heuristic algorithm is able to find adequate solutions in an acceptable time range.

In the previous sections of this chapter, we discussed the studies related to our problem and the solution methods that we used. In the literature, the identification and application of different solution methods for SCDP have already been covered. Several contributions to the SCDP have been made in this thesis, described as follows:

- Predefined fixed routes are not used in SCDP. Instead, we aim to find efficient results by considering all possible routes. Although solving the problem becomes more complicated to solve via this addition, it also causes the freight forwarders to save more in terms of cost and distance.

- To the best of our knowledge, the most important difference between our study and other studies in the literature is that it deals with shipment consolidation and dispatching problems with more than one objective function. In this respect, we create a mathematical model to find the efficient frontier. And for larger-sized instances, we develop a multi-objective variable neighborhood search heuristic.

CHAPTER 3 : METHODOLOGY

The purpose of this study is to find the best routes for Shipment Consolidation and Dispatching Problem (SCDP) with two objectives, where the first one is to minimize the total cost and the second one is to minimize the total distance travelled. For this purpose, we present a Multi-Objective Mixed-Integer Linear Programming (MOMILP) model. Since the computation time is too long due to the difficulty of the problem, the exact solution method is not efficient to solve, especially for the large instances of the problem. Thus, a meta-heuristic algorithm is suggested to obtain near-optimal solutions in acceptable computation time.

In this chapter, we define the problem and discuss the assumptions. We then propose the multi-objective mathematical model for the SCDP that has no predefined routes and present the solution techniques. Finally, we present the Variable Neighborhood Search (VNS) algorithm developed to obtain approximate solutions in short computation time.

3.1 *Problem Definition*

We aim to determine the best route for SCDP considering two objectives where the first one is to minimize the total cost and the second one is to minimize the total distance. The total cost comprises of the fixed cost of each vehicle used, additional cost for extra stops and cost of using transshipment terminals. Furthermore, per kilometer cost for a vehicle is computed according to the distance of the farthest location from the initial location among the destinations assigned to that vehicle. For SCDP based on real-life, we model the problem considering the following assumptions:

- Data about orders and vehicles is deterministic.
- Orders can be delivered with two different options. The first option is direct delivery. The second option is to deliver the order by using one of the contracted transshipment terminals.
- For the orders which are delivered via the transshipment terminal, the terminal is in charge of the delivery of the order to the final destination. The cost is determined based on the size of the order and the transshipment terminal used.
- Routes of vehicles are open routes. All vehicles start their route from the initial location. Then they stop by the destinations that are assigned to them. Routes

end at their last order's final destination. They do not turn back to the starting point.

- It is included in the fixed cost for a limited number of stops. The additional cost is applied for each stop that exceeds the number of stops allowed to be charged as fixed costs.
- According to the vehicles capacity and the time-windows of orders for deliveries, it is determined whether the orders can or cannot be in the same vehicle.
- The transit time of destinations is assumed to be constant and regardless of the assigned vehicle or any other orders assigned to that vehicle.

Since orders have a release date and due date information, the problem has a time dimension. According to the release date, due date and travel time of orders, it is decided whether it is possible for the orders to assign to the same vehicle. Instead of adding the time-windows as new constraint sets in the proposed model, we define a parameter that identifies whether the orders can be in the same vehicle or not. The corresponding parameter is as follows:

$$a_{kl} = \begin{cases} 1 & \text{if orders } k \text{ and } l \text{ can be assigned to the same vehicle} \\ 0 & \text{otherwise} \end{cases}$$

A basic sample is illustrated in Table 1. It is assumed that the transit time of an order is 5 days. The release dates and deadlines of orders are shown in the table. The orders must depart the latest 5 days before their deadlines. This time duration is shown as light grey in the table. The section marked with dark grey represents the available departure dates for each order. Orders (1) and (2) could leave together on day 2 or day 3 in the same vehicle. On the other hand, order (1) cannot be in the same vehicle with order (3), because they do not have any common departure day. Lastly, orders (2) and (3) can depart together on 4^{th} day in the same vehicle. In this manner, the values of a_{12}, a_{13} and a_{23} are 1, 0 and 1, respectively.

Table 1. Time-windows for deliveries

	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10
Order 1										
Order 2										
Order 3										

3.2 *Mathematical Model Formulation*

According to the assumptions that are discussed above, we develop the mathematical model for the problem as follows:

Indices and Sets

$\mathscr{K}$: Set of orders, $k,l \in \mathscr{K}$

$\mathscr{T}$: Set of vehicles, $t \in \mathscr{T}$

$\mathscr{L}$: Set of locations, $i, j \in \mathscr{L}$

L_0 : Initial location, $L_0 \subset \mathscr{L}$

L_q : Set of transshipment terminals, $L_q \subset \mathscr{L}$

L_d : Set of destinations, $L_d \subset \mathscr{L}$

Parameters

v_k : Volume of order k

w_k : Weight of order k

l_k : Length of order k

ν : Volume capacity of vehicle

ω : Weight capacity of vehicle

τ : Length capacity of vehicle

p_k : Delivery location of order k, where $p_k \in \mathscr{L}$

d_{ij} : Distance between locations i and j

pkm: Cost per kilometer

foc: Fixed cost of operating a vehicle

$cost_{ki}$: Cost of using transshipment terminal i to deliver order k

μ : Maximum number of extra stops

ϕ : Number of stops allowed to be charged as fixed costs, where $1 \leq \phi \leq \mu$

ρ : Additional cost for extra stops

$$a_{kl} : \begin{cases} 1 & \text{if orders } k \text{ and } l \text{ can be assigned to the same vehicle} \\ 0 & \text{otherwise} \end{cases}$$

Decision Variables

α_t = Fixed cost of vehicle t

β_t = Number of extra stops for vehicle t

$$x_k^t = \begin{cases} 1 & \text{if order } k \text{ is departed in vehicle } t \\ 0 & \text{otherwise} \end{cases}$$

$$y_k^t = \begin{cases} 1 & \text{if order } k \text{ is delivered directly with vehicle } t \\ 0 & \text{otherwise} \end{cases}$$

$$z_{ki}^t = \begin{cases} 1 & \text{if order } k \text{ is in vehicle } t \text{ and delivered through transshipment terminal } i \\ 0 & \text{otherwise} \end{cases}$$

$$\gamma_{ij}^t = \begin{cases} 1 & \text{if location } j \text{ is visited right after location } i \text{ by vehicle } t \\ 0 & \text{otherwise} \end{cases}$$

$$\delta_i^t = \begin{cases} 1 & \text{if vehicle } t \text{ visits location } i \\ 0 & \text{otherwise} \end{cases}$$

Mathematical Model

$$\text{Min} \quad Z_1 = \sum_{t \in \mathscr{T}} (\alpha_t + p\beta_t + \sum_{i \in L_q} \sum_{k \in \mathscr{K}} cost_{ki} z_{ki}^t) \tag{3.1}$$

$$\text{Min} \quad Z_2 = \sum_{i \in \mathscr{L}} \sum_{j \in \mathscr{L}: j \neq i} \sum_{t \in \mathscr{T}} d_{ij} \gamma_{ij}^t \tag{3.2}$$

Subject to:

$$\alpha_t \geq (d_{0j}.pkm + foc)\delta_j^t \qquad j \in \mathscr{L} \backslash L_0, t \in \mathscr{T} \tag{3.3}$$

$$\delta_{p_k}^t \geq y_k^t \qquad k \in \mathscr{K}, t \in \mathscr{T} \tag{3.4}$$

$$\delta_i^t \geq z_{ki}^t \qquad k \in \mathscr{K}, i \in L_q, t \in \mathscr{T} \tag{3.5}$$

$$\beta_t \geq \sum_{i \in \mathscr{L} \backslash L_0} \delta_i^t - \phi \qquad t \in \mathscr{T} \tag{3.6}$$

$$\beta_t \leq \mu \qquad t \in \mathscr{T} \tag{3.7}$$

$$x_k^t = y_k^t + \sum_{i \in L_q} z_{ki}^t \qquad k \in \mathscr{K}, t \in \mathscr{T} \qquad (3.8)$$

$$\sum_{t \in \mathscr{T}} x_k^t = 1 \qquad k \in \mathscr{K} \qquad (3.9)$$

$$x_k^t + x_l^t \leq 1 \qquad k,l \in \mathscr{K} \mid a_{kl} = 0, t \in \mathscr{T} \qquad (3.10)$$

$$\sum_{k \in \mathscr{K}} v_k x_k^t \leq \nu \qquad t \in \mathscr{T} \qquad (3.11)$$

$$\sum_{k \in \mathscr{K}} w_k x_k^t \leq \omega \qquad t \in \mathscr{T} \qquad (3.12)$$

$$\sum_{k \in \mathscr{K}} l_k x_k^t \leq \tau \qquad t \in \mathscr{T} \qquad (3.13)$$

$$\alpha_t \geq \alpha_{t-1} \qquad t \in \mathscr{T} \mid t \geq 2 \qquad (3.14)$$

$$\sum_{j \in \mathscr{L} \setminus \{i\}} \gamma_{ij}^t \leq \delta_i^t \qquad i \in \mathscr{L}, t \in \mathscr{T} \qquad (3.15)$$

$$\sum_{i \in \mathscr{L} \setminus \{i\}} \gamma_{ij}^t = \delta_j^t \qquad j \in \mathscr{L} \setminus L_0, t \in \mathscr{T} \qquad (3.16)$$

$$u_i^t - u_j^t + 1 \leq (\mu + \phi)(1 - \gamma_{ij}^t) \qquad i,j \in \mathscr{L}, t \in \mathscr{T} \qquad (3.17)$$

$$u_i^t \leq \beta_t + \phi, \qquad i \in \mathscr{L}, t \in \mathscr{T} \qquad (3.18)$$

$$x_k^t, y_k^t, z_{ki}^t, \gamma_{ij}^t, \delta_i^t \in \{0,1\} \qquad i,j \in \mathscr{L}, t \in \mathscr{T} \qquad (3.19)$$

$$\alpha_t, u_i^t \geq 0 \qquad i \in \mathscr{L}, t \in \mathscr{T} \qquad (3.20)$$

$$\beta_t \in \mathbb{Z}^{\geq} \qquad t \in \mathscr{T} \tag{3.21}$$

The objective function (3.1) aims to find the minimum overall cost, which is computed as the summation of the fixed cost of vehicles, the additional cost of extra stops and the cost of delivery from the transshipment terminals. The objective function (3.2) minimizes the overall distance travelled by the vehicles. Constraint set (3.3) defines the fixed cost of each vehicle. The fixed cost includes both the fixed operating cost and the cost per kilometer regarding the farthest destination from the vehicle's initial point. Constraint set (3.4) ensures that if an order is delivered directly, the destination of that order is assigned to it. Similarly, constraint set (3.5) ensures that if an order is delivered from a transshipment terminal by a vehicle, the transshipment terminal location is assigned to that vehicle. Constraint sets (3.6) and (3.7) define the number of additional stops and ensure that it does not exceed the maximum number. Constraint set (3.8) guarantees that if an order is assigned to a vehicle, it is delivered either directly or from the transshipment terminal. Constraint set (3.9) assures each order is assigned a single vehicle. Constraint set (3.10) prevents the assignment of two orders to the same vehicle if those orders cannot have a common departure day due to their release dates and deadlines. Constraint sets (3.11), (3.12) and (3.13) are volume, weight and length capacity constraints for each vehicle, respectively. Constraint set (3.14) is for symmetry breaking, which ensures that the higher index vehicle is used after the lower index vehicle. Constraint sets (3.15) and (3.16) limit that if a vehicle visits a location, it can be only visited right after one location and after that location at most one location can be visited. Constraint sets (3.17) and (3.18) are sub-tour elimination constraints.We used the Miller–Tucker–Zemlin formulation which is proposed by Miller et al. (1960) as sub-tour elimination constraint sets. u_i^t represents the dummy variables that keep track of the number of locations visited counting from the depot, and are bounded by the maximum allowable stops by a truck. Constraint sets (3.19), (3.20) and (3.21) restrict sign and identify types of decision variables.

3.3 *Generating the Efficient Frontier*

In this thesis, the mathematical model for SCDP is a bi-objective optimization problem. Due to the conflict between Z_1 and Z_2 objectives, no unique optimal solution exists. However, by managing the computation between these objective functions, a set of Pareto-optimal solutions can be developed. The idea of optimality is replaced in multi-objective problems with the concept of Pareto optimality or efficiency. If an objective function cannot be improved without degrading at least one other objective function, the solution is considered Pareto optimal (Mavrotas, 2009). Because the model's two objectives are contradictory and cannot be optimized simultaneously, multi-objective solution methods should always be used. The following are some of the most important multi-objective solution methods, which are the ε-constraint approach and the augmented ε-constraint approach.

3.3.1 *ε-Constraint Method*

There are different approaches to obtain efficient solutions for multi-objective optimization problems, as we reviewed in Chapter 2. ε-constraint and weighting approach are two of the most widely used approaches; see Steuer (1986). Furthermore, ε-constraint approach has superiorities over the weighting approach. For instance, there can be a lot of unnecessary runs in the weighting approach because there are a number of weight combinations that produce the same efficient solution (Mavrotas, 2009).

For bi-objective problems, the ε-constraint technique optimizes one of the objective functions while including the other objective function in the model as a constraint and changing the right-hand-side iteratively. In this way, a model with a single objective is constructed from the prior model. Considering the mathematical model introduced on Section 3.1, the ε-constraint approach is performed as below.

$$\text{Min} \quad Z_1 \tag{3.22}$$

Subject to:

$$Z_2 \leq e_2 \tag{3.23}$$

$$\text{Equations } (3.3) - (3.21)$$

The operation principle of the above model is to find the optimal value of Z_1 first by considering the Z_2 as a constraint. The result of Z_1 is noted as Z_1^* and Z_2 as Z_2'. Then, e_2 is updated as Z_2' - 1. To find each non-dominated solution, the model is solved with the updated value of e_2, until the optimal value of Z_2^* is found. The efficient set of the problem is obtained with various variations in e_2. The reason why we prefer to create an efficient frontier by updating Z_2 values instead of Z_1 is that Z_2 is an integer. Instead of trying to find the optimal epsilon value to improve a continuous variable, it has been much more efficient to get a non-dominated solution by subtracting an integer variable by one. Nevertheless, the obtained solutions in the conventional ε-constraint method are not always Pareto-optimal. In other words, some weakly efficient solutions may be produced. Mavrotas (2009) proposed the augmented-constraint method to overcome this drawback of the conventional ε-constraint method.

3.3.2 *Augmented ε-Constraint Method (AUGMECON)*

In this thesis, we use the augmented ε-constraint approach to generate the Pareto front for the MOMILP. The acronym for this procedure is the AUGMECON method. The AUGMECON is an advanced variant of the standard ε-constraint approach that prevents weakly Pareto-optimal solutions from being obtained (Mavrotas, 2009).

One of the drawbacks of the conventional ε-constraint approach is that the generated solutions are not always Pareto optimum. By performing lexicographic optimization to every objective function, the augmented ε-constraint approach aims to overcome this drawback. In this manner, the payoff table contains only Pareto-optimal solutions. In practice, the working principle of lexicographic optimization is to first optimize the first objective function and obtain the minimum value of Z_1 which is Z_1^*. We add a constraint as Z_1=Z_1^* after we find Z_1^* and optimize the second objective function. As a result, we can obtain a minimum value of Z_2 where Z_1 is optimal. By including the relevant non-negative slack variables (s_2), the objective function constraint is turned into equality. Consequently, we apply the formulation of the problem based on AUGMECON as given in the following mathematical model.

$$\text{Min} \quad Z_1 - eps \times (s_2) \tag{3.24}$$

Subject to:

$$Z_2 + s_2 = e_2 \tag{3.25}$$

$$s_2 \geq 0 \tag{3.26}$$

$$\text{Equations } (3.3) - (3.21)$$

Here, e_2 is equal to one less than the last obtained minimal value of Z_2 where Z_1 is optimal, and eps is a very small value. By using the Augmented ε-constraint approach, the Pareto solution is found and reported.

The pseudo-code of the augmented ε-constraint approach is given in Algorithm 1 for a bi-objective minimization problem. The procedure of the method could be explained as follows.

Algorithm 1 Augmented ε-Constraint Method

Create payoff table
Calculate Z_1^{min} and Z_2^{min}
Calculate $Z_1^{min\{Z_2\}} \rightarrow min\{Z_1 : Z_2 = Z_2^{min}\}$ and $Z_2^{min\{Z_1\}} \rightarrow min\{Z_2 : Z_1 = Z_1^{min}\}$
Add $(Z_1^{min}, Z_2^{min\{Z_1\}})$ to the Pareto-optimal set
$e_2 \rightarrow Z_2^{min\{Z_1\}} - \varepsilon$
while $e_2 \geq Z_2^{min}$ **do**
    Solve the following model:

$$\text{Min} \quad Z_1 - eps(s_2)$$

Subject to:

$$Z_2 + s_2 = e_2$$

$$s_2 \geq 0$$

$$Equation(3.3) - (3.21)$$

    Add the optimal values of Z_1^* and Z_2^* to the Pareto-optimal set
    $e_2 \rightarrow Z_2^* - \varepsilon$
end while
Report the Pareto-optimal set

3.4 *Variable Neighborhood Search*

In this section, we first present the basics of the variable neighborhood heuristic. We then discuss the multi-objective version of the heuristic and the details of our implementation to the shipment consolidation and dispatching problem.

3.4.1 *Basics of Variable Neighborhood Search Heuristic*

Considering most real-world problems are NP-hard, the exact solution techniques could be insufficient to solve real-life problems in polynomial time. Optimization and general heuristic methods perform poorly either for too much computation time or stuck in some local optima. When we have a restricted amount of time and the problem is NP-hard, it is a smart idea to utilize metaheuristics. Although metaheuristic algorithms do not promise to find exact solutions, they can find near-optimal solutions within an acceptable time period. Considering that we are trying to solve an NP-hard problem, it is obvious that the performance of the epsilon constraint method will be insufficient, especially for large instances. Therefore, we foresee using heuristic methods to obtain the solutions in an acceptable time range.

One of the most basic and widely used versions of those is the local search algorithms. The general concept of local search algorithms starts with a feasible solution, which is usually found by a simple heuristic and progresses by trying to improve the objective function value by making some local changes. The algorithm eventually finds a local optimal solution, which is not always the global optimal solution. Different metaheuristics have been developed in order to avoid becoming stuck in a local optimal solution. Bee colony, tabu search, and genetic algorithm can be given as examples of the most commonly used metaheuristics. One of the effective metaheuristic methods is the Variable Neighborhood Search (VNS) algorithm, which was introduced by Hansen and Mladenović (1997).

Researchers have applied VNS to many different optimization problems, since it was introduced. Vehicle routing problem (Hemmelmayr et al., 2009), p-median problem (Fleszar and Hindi, 2008) and portfolio selection (Özpeynirci et al., 2022) can be given as examples. As VNS can get very good results for location based problems (Hansen and Mladenović, 1997), we expect VNS to produce near-optimal solutions for our problem as well. Figure 2 illustrates the basic logic of the VNS for minimization problems. The search principle of VNS aims to reach the global optimal point by allowing both intensification and diversification. We can control the balance of them by determining the neighborhood combinations and their frequency.

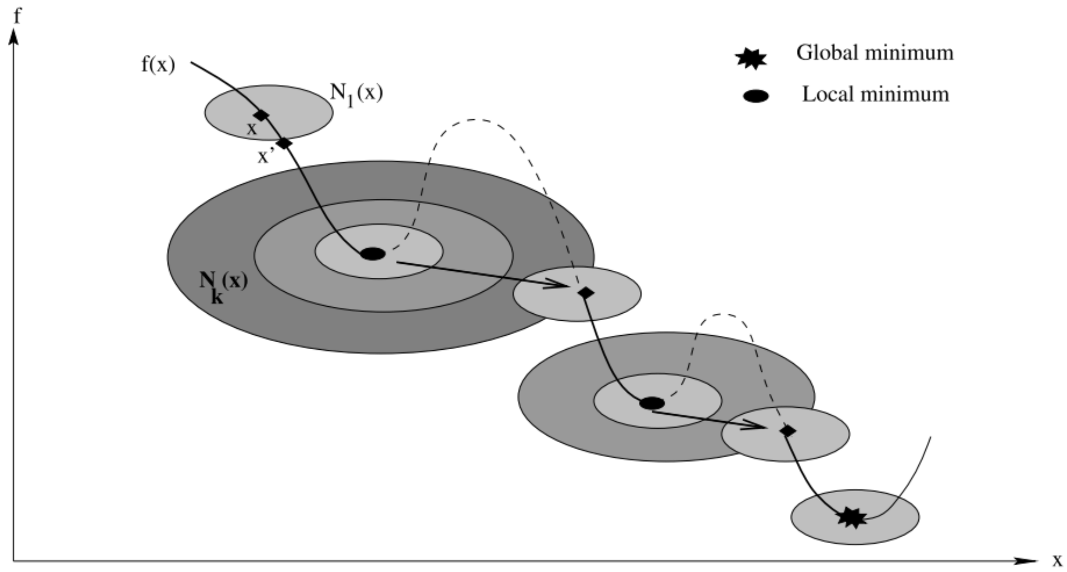

Figure 2. Basic VNS (Source: Hansen et al. 2010)

3.4.2 *Multi-Objective Variable Neighborhood Search for Shipment Consolidation and Dispatching Problem*

The basic principle of the VNS is to search in the solution space while avoiding local optima by changing the neighborhood. When applying this principle, unlike the population-based heurictics, as a trajectory-based metaheuristic, basic VNS uses a single solution instead of a set of solutions. To overcome this problem, Duarte et al. (2015) proposed that the approximate Pareto front discovered throughout the search process can be considered the current solution to a multi-objective problem. Thus, we apply the algorithm to develop an approximate efficient frontier by denoting the solutions in the Pareto frontier.

In this section, we introduce a VNS heuristic to solve the bi-objective shipment consolidation and dispatching problem.

The pseudo-code of the proposed multi-objective VNS is provided in Algorithm 2. The necessary criteria for the algorithm are determined during initialization. By using different heuristic algorithms, the initial efficient frontier set S is created. Then, by using the defined neighborhood structures, the shaking operation starts for each s in set S. Following that, we perform a local search for the found s' and obtain s''. We check whether any of the obtained neighborhoods of s update the set S or not. We can have four different cases for the new neighborhoods:

(i) All of the solutions in S can dominate s' and s''. So, the efficient frontier S is not updated.

(ii) Some of the solutions in S can be dominated by s' or s''. In this situation, s' or s'' is added to set S and the dominated solutions are eliminated from the set S.

(iii) All of the solutions in S can be dominated by s' or s''. If so, all the solutions are removed from current S and the new non-dominated solution is added to the efficient frontier.

(iv) s' or s'' can be a non-dominated solution. For this case, we update the efficient frontier by adding the new non-dominated solution.

For the new solution found in case (i), the algorithm continues with the next neighborhood of s. However, if one of the cases (ii), (iii) or (iv) is valid, k is updated as $k \leftarrow 1$ and the search continues for the newly discovered solution. This procedure continues until all solutions in S are explored. The algorithm is repeated for the current S until the stopping criteria is satisfied. In the proposed algorithm, we set the stopping condition as the maximum number of iterations. Figure 3 shows the flow diagram of our proposed algorithm.

Algorithm 2 Multi-objective Variable Neighborhood Search

Initialization:
Define the set of neighborhood structures. $N_k(k = 1, ..., k_{max})$
Generate a set of non-dominated frontier using different heuristics.
$S \leftarrow$ set of non-dominated solutions
Determine the stopping condition.
$t \leftarrow$ current number of iterations
$t_{max} \leftarrow$ maximum number of iterations
Main step:
for $t = 1 \rightarrow t_{max}$ **do**
    Denote all efficient frontier solutions in S as unexplored
    while S has at least one unexplored solution **do**
        Select an unexplored solution s in S and denote it as explored.
        for $k = 1 \rightarrow k_{max}$ **do**
            Shaking: Generate a random feasible solution s' using k^{th}
            neighborhood of s. $(s' \in N_k(s))$. $S \leftarrow$ update
            Local Search: Solve the proposed multi-objective mathematical
            model for s' to apply local search and find s''.
            if s' or s'' update S **then**
                $k \leftarrow 1$
            else
                $k \leftarrow k + 1$
            end if
        end for
    end while
end for

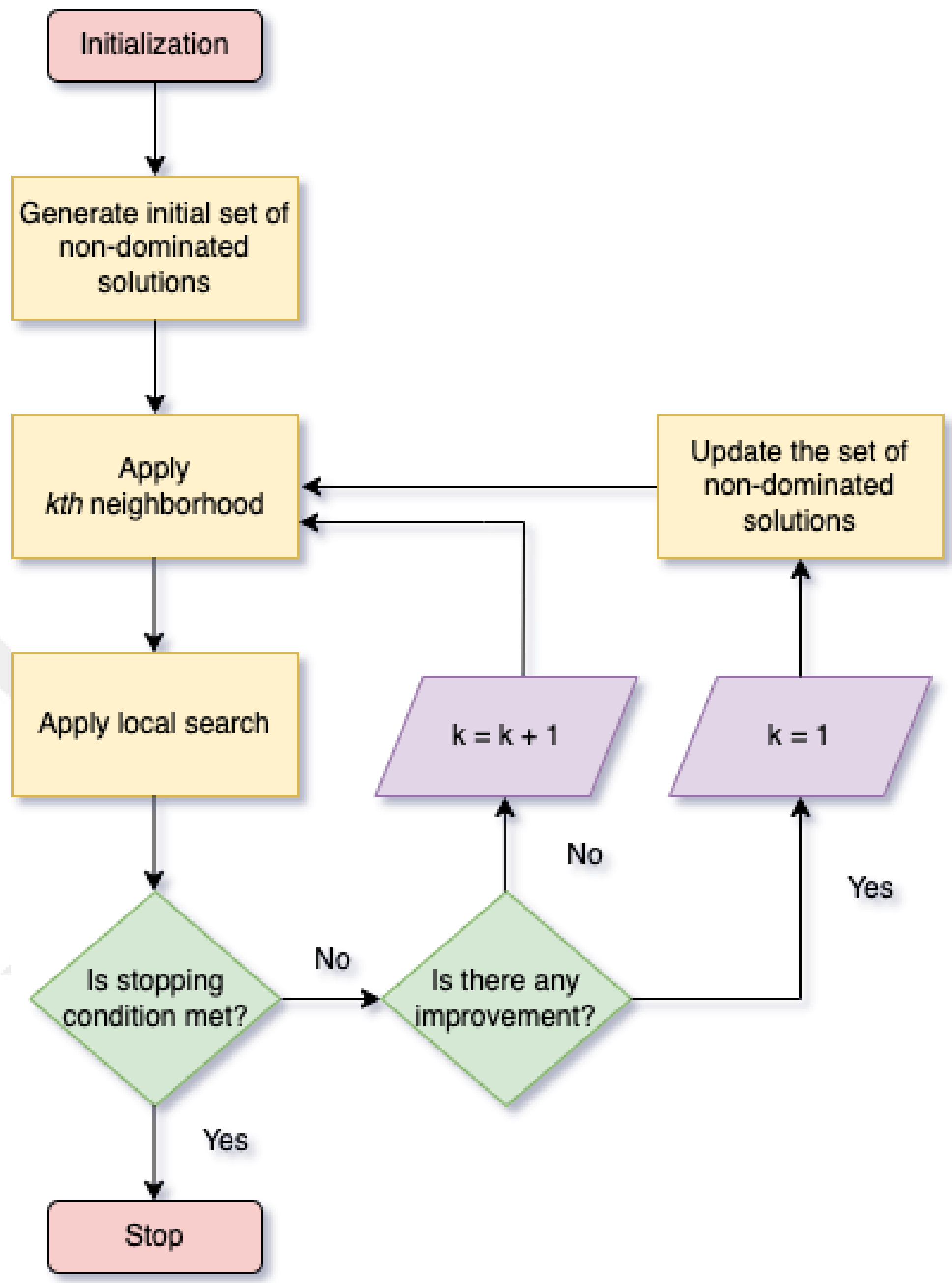

Figure 3. Flowchart of the Proposed Multi-objective Variable Neighborhood Search Algorithm

3.4.2.1 *Initialization*

In order to provide a robust launch of the algorithm, we try different heuristics methods. Instead of starting from a single non-dominated solution, neighborhood structures start to be explored from many different non-dominated points, thanks to the different heuristics used. After we obtain the initial solutions from the heuristics, we produce other feasible solutions by randomly changing the delivery points. The three heuristics that we use are as follows:

Nearest Neighbor Algorithm

To create the initial solution, we first use the nearest neighbor algorithm for m locations and m vehicles. The algorithm determines the number of vehicles needed. While applying the algorithm, it is assumed that all orders are delivered to their direct destinations.

Initially, according to the distance matrix (D) the order with the nearest delivery point to the depot is determined. The solution starts by assigning this order to the selected vehicle (K_t). Then the algorithm assigns the nearest order to the current order's destination. The algorithm checks if the orders are eligible to be in the same vehicle with respect to their release dates and deadlines, which we indicate as a parameter of a_{kl}. It continues to assign the orders to the same vehicle until the vehicle capacity is exceeded. For other vehicles, the same procedure is applied. The algorithm stops when all orders are assigned to a vehicle. Overview of the suggested heuristic is given in Algorithm 3.

Algorithm 3 Nearest Neighbor Algorithm

Input: $D \leftarrow$ distance matrix
    $R \leftarrow$ set of remaining orders
    $c \leftarrow$ current location
    $CT \leftarrow$ capacity of vehicles
Output: Feasible solution
    $S_t \leftarrow$ set of the orders in vehicle t
for $t = 1 \rightarrow m$ **do**
    $c \leftarrow 0$
    for $j = 1 \rightarrow m$ **do**
        $n \leftarrow$ Determine the nearest order for the c
        if n is eligible for S_t and capacity of vehicle $t <= CT$ **then**
            Assign n to S_t
            Remove n from R
            $c \leftarrow n$
        end if
    end for
    if R is empty **then**
        break;
    end if
end for

Nearest Neighbor Algorithm with Randomized Distance Matrix

The distance matrix is randomized by multiplying each distance with a random number between 0.5 and 1.5. According to the varying values of the distance parameter, the very same nearest neighbor algorithm procedure is applied; see Algorithm 3. The non-dominated frontier is enriched by obtaining additional feasible solutions.

Savings Algorithm

One of the other simple heuristics used for location-based problems is the savings algorithm. We present the Savings Algorithm in Algorithm 4. We assume that all orders are delivered to their direct destinations with a different vehicle. At the beginning, we calculate the savings for each possible pair of orders. Instead of two vehicles departing from the depot, a single vehicle departs from the depot and delivers the orders one by one. Savings occur by connecting two locations of orders. The savings are calculated by $SV_{kl} = d_{k0} + d_{0l} - d_k$ for all $k,l >= 1$ and $k \neq l$, where d_{kl} represents the distance between the delivery points of orders k and l. The savings are sorted in descending order. Starting from the maximum savings, we merge the routes into the same vehicle. While the routes are merging, the capacity and eligibility conditions must be satisfied. The algorithm stops when there is no feasible solution that can be produced by saving. We also apply the same procedure to save the total distance instead of saving the total cost. Thus, we get two different solutions for the objectives of cost-saving and distance-saving.

Algorithm 4 Savings Algorithm

Input: $D \leftarrow$ distance matrix
    $RS \leftarrow$ set of remaining pair of orders
    $SV_{kl} \leftarrow$ set of savings for merging the order k and l's delivery points
Output: Feasible solution
    $S_t \leftarrow$ set of the orders in vehicle t
Sort the SV_{kl} in descending order
for $t = 1 \rightarrow m$ **do**
    for $i = 1 \rightarrow m$ **do**
        for $j = 1 \rightarrow m$ **do**
            $n \leftarrow$ Determine the maximum SV_{kl} from current RS
            if i and j is eligible for S_t and capacity of vehicle $k <= CT$ **then**
                Assign i and j to S_t
                Remove the pairs which starts with i and finishes with j from RS
            end if
        end for
    end for
    if RS is empty **then**
        break;
    end if
end for

From the heuristics we mentioned above, we obtain several different feasible solutions; one from the nearest neighbor algorithm, ten from the nearest neighbor algorithm with a randomized distance matrix, one from a savings algorithm using an objective function to minimize the total cost, and one from a savings algorithm using an objective function to minimize the total cost. The obtained solutions comprise only direct deliveries to the final destination of orders. Yet there is another delivery option that is delivered via transshipment terminals. We randomly change the delivery option of each order in order to explore the various feasible solutions for delivery via transshipment terminals. Therefore, for each solution that we obtain by using different heuristics, we create different versions of the solutions with randomly changed delivery options. Finally, we update the efficient frontier by using the new solutions. Thus, we initialize with a robust non-dominated frontier.

3.4.2.2 *Shaking*

Randomized perturbation in VNS, known as shaking, is one approach to escape the local minimum. It is an important part of the VNS algorithm in which a shaking procedure generates a neighborhood solution of the current solution. We identified four neighborhood structures for the shaking procedure. Also, we use two different levels (ℓ=1, ℓ=2) for each neighborhood. The intensity of the change created by the neighborhood is determined by the level. In total, we search through eight neighborhoods for each solution. We define neighborhood structures as follows:

- **Move:** If it is possible, we move ℓ randomly selected orders to randomly selected vehicles. If there is no feasible solution with moving action, continue the procedure with the next neighborhood.
- **Swap:** We select a pair of orders (η_1, η_2) randomly such that $\eta_1 \neq \eta_2$ from different vehicles. We swap the assigned vehicles of a selected pair of orders, if feasible. If there is no feasible solution with swapping action, continue the procedure with the next neighborhood.
- **Perturbation:** We select ℓ vehicles randomly and remove all orders from them. Then we randomly assign orders from different vehicles to the selected vehicles. Lastly, we randomly place removed orders for the vehicles. If there is no feasible solution with perturbation, continue the procedure with the next neighborhood.
- **Remove:** Suppose T is the total number of vehicles for the current solution. We select the vehicle with the largest remaining capacity. Then place all orders in

$T - \ell$ vehicles. If there is no feasible solution with removing action, continue the procedure with the next neighborhood.

3.4.2.3 *Local Search*

We present a mathematical model for the local search procedure. The algorithm has two objectives: to minimize total cost and to minimize the total distance for each vehicle in the current solution. We use the current solution as the input to the mathematical model. The input of the mathematical model is identified as two different sets, as follows:

$\mathscr{T}'$: Set of vehicles that has been updated in the latest iteration, $t \in \mathscr{T}'$

S_t : Set of orders assigned to vehicle t, $t \in \mathscr{T}'$

The model for local search procedure is as follows;

$$\text{Min} \quad Z_1 = \sum_{t \in \mathscr{T}} (\alpha_t + p\beta_t + \sum_{i \in L_q} \sum_{k \in \mathscr{K}} cost_{ki} z_{ki}^t) \tag{3.27}$$

$$\text{Min} \quad Z_2 = \sum_{i \in \mathscr{L}} \sum_{j \in \mathscr{L} \setminus i} \sum_{t \in \mathscr{T}} d_{ij} \gamma_{ij}^t \tag{3.28}$$

Subject to:

$$\alpha_t \geq (d_{0j}.pkm + foc)\delta_j^t \qquad j \in \mathscr{L} \setminus \{0\}, t \in \mathscr{T} \tag{3.29}$$

$$\delta_i^t \geq z_{ki}^t \qquad k \in \mathscr{K}, i \in L_q, t \in \mathscr{T} \tag{3.30}$$

$$\delta_{p_k}^t \geq y_k^t \qquad k \in \mathscr{K}, t \in \mathscr{T} \tag{3.31}$$

$$x_k^t = y_k^t + \sum_{i \in L_q} z_{ki}^t \qquad k \in \mathscr{K}, t \in \mathscr{T} \tag{3.32}$$

$$x_k^t = 1 \qquad k \in S_t, t \in \mathscr{T}' \tag{3.33}$$

$$\sum_{j \in \mathscr{L} \setminus i} \gamma_{ij}^t \leq \delta_i^t \qquad i \in \mathscr{L}, t \in \mathscr{T} \tag{3.34}$$

$$\sum_{i \in \mathscr{L} \setminus i} \gamma_{ij}^t = \delta_j^t, \qquad j \in \mathscr{L} \setminus \{0\}, t \in \mathscr{T} \tag{3.35}$$

$$u_i^t - u_j^t + 1 \leq (\mu + \phi)(1 - \gamma_{ij}^t), \qquad i, j \in \mathscr{L}, t \in \mathscr{T} \tag{3.36}$$

$$u_i^t \leq \beta_t, \qquad i \in \mathscr{L}, t \in \mathscr{T} \tag{3.37}$$

$$x_k^t, y_k^t, z_{ki}^t, \gamma_{ij}^t, \delta_i^t \in \{0,1\} \qquad i, j \in \mathscr{L}, t \in \mathscr{T} \tag{3.38}$$

$$\alpha_t, u_i^t \geq 0 \qquad i \in \mathscr{L}, t \in \mathscr{T} \tag{3.39}$$

$$\beta_t \in \mathbb{Z}^{\geq} \qquad t \in \mathscr{T} \tag{3.40}$$

The objective function (3.27) minimizes the overall cost, which is computed as the sum of the fixed cost of vehicles, the additional cost of extra stops, and the cost of delivery from the transshipment terminals. Objective function (3.28) minimizes the total distance. Constraint set (3.29) defines the fixed cost of each vehicle. The fixed cost includes the fixed operating cost and the cost per kilometer regarding the farthest destination from the initial location of the vehicle. Constraint set (3.30) ensures that if an order is delivered directly by a vehicle, the destination of that order is assigned to that vehicle. Similarly, constraint set (3.31) ensures that if an order is delivered from a transshipment terminal by a vehicle, the transshipment terminal location is assigned to that vehicle. Constraint set (3.32) guarantees that if an order is assigned to a vehicle, it is delivered either directly or from the transshipment terminal. Constraint set (3.33) assures assigning each order to the corresponding vehicle according to the current solution . Constraint sets (3.34) and (3.35) are related the decision variables with each other. Constraint sets (3.36) and (3.37) are sub-tour elimination constraints. Constraint sets (3.38), (3.39) and (3.40) restrict sign and identify types of decision variables.

This formulation is modified as a single objective mathematical model by normalizing the objective functions Z_1 and Z_2. Instead of using objective function (3.27) and objective function (3.28), we define a normalized objective function as follows:

$$Z_{norm} = \quad w_1 \frac{Z_1}{C_{min}} + w_2 \frac{Z_2}{D_{min}} \tag{3.41}$$

Where $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$, equation (3.41) defines Z_{norm} as the overall cost of the solution by dividing C_{min}, the lower bound on the overall cost and the total distance of the solution by dividing D_{min}, the lower bound on the overall distance. For small-sized instances, C_{min} and D_{min} is founded respectively the minimum value of cost and distance of the mathematical model that is presented in Section (3.2). For medium and large-sized instances, since optimal values cannot be found in an acceptable time, the minimum value of cost and distance values in the initial solution found by VNS are used as C_{min} and D_{min}.

The general framework of the proposed local search is presented in Algorithm 5. First, we prepare the necessary data for the algorithm. For the initial non-dominated solution set, we apply the local search algorithm for all orders. Afterwards, the mathematical model is solved only for the orders and vehicles that have been changed with the shaking procedure. We ensure this by updating the value of the decision variable X_k^t either 1 or 0. After these adjustments, we solve the mathematical model by using the callable library of Cplex solver. Then, according to the best solution that the mathematical model found, we update the solution according to the solution structure of the heuristic algorithm.

Algorithm 5 Local Search

Input: $\mathscr{T}'$: Set of vehicles that has been updated in the latest iteration

S_t : Set of orders assigned to vehicle t, $t \in \mathscr{T}'$

C_{min} : The lower bound on the total cost

D_{min} : The lower bound on the total distance

for $k \in \mathscr{K}$ **do**

  for $t \in \mathscr{T}$ **do**

    if $k \in S_t$ and $t \in \mathscr{T}'$ **then**

      $X_k^t \leftarrow 1$

    else

      $X_k^t \leftarrow 0$

    end if

  end for

end for

Solve the following model:

$$\text{Min} \quad Z_{norm} = \quad w_1 \frac{Z_1}{C_{min}} + w_2 \frac{Z_2}{D_{min}}$$

Subject to:

$$Z_1 = \sum_{t \in \mathscr{T}} (\alpha_t + p\beta_t + \sum_{i \in L_q} \sum_{k \in \mathscr{K}} cost_{ki} z_{ki}^t)$$

$$Z_2 = \sum_{i \in \mathscr{L}} \sum_{j \in \mathscr{L} \setminus i} \sum_{t \in \mathscr{T}} d_{ij} \gamma_{ij}^t$$

$$Equation (3.29) - (3.40)$$

Adapt the best solution found to the solution structure of the heuristic algorithm.

CHAPTER 4 : COMPUTATIONAL EXPERIMENTS

In this chapter, first, we clarify the data used in the experiments. Afterwards, we examine the results of the augmented ε-Constraint Method we applied for the exact approach. Eventually, we present and analyze our heuristic approach results.

During the testing of both the exact and heuristic approaches that we proposed, we used the data sets provided by Tokcaer (2018). We do not use the set of fixed routes data they use. As we mentioned before, the presented solution methods aim to choose the most efficient route from the set of feasible solutions. The data about locations, transshipment terminals, and distances between locations is fixed for every instance set. Instances are created with different combinations of 22 previously known locations. The sizes of the orders are examined in three different dimensions. These are weight, volume, and length. Total weight, volume, and height values are calculated for each order according to the pallets or boxes they contain. The sample set is diversified over three main parameters. These are the number of orders, the elasticity of orders for time windows, and the diversity of orders' destinations.

Number of orders: The number of orders is represented with the parameter I. It can have values of 10, 20,and 30. Each order has a release date and according to the distance between its destination and the depot, a deadline is determined for each order. As the distance gets closer, the transit time gets shorter with the same rate. This narrows the time interval that the orders can depart.

Elasticity of orders for time windows: Parameter B is defined to control the elasticity of orders for time windows. Time windows are determined with respect to the release date and delivery date of orders. The existing transit time is replaced by multiplying the values 0.1 and 0.7. In the samples examined, those multiplied by 0.1 are considered as $B = 10$, and those multiplied by 0.7 are considered as $B = 70$. When B is 10, the time between the due date and the release date is short. The same applies in the opposite case.

Diversity of order's destinations: D is used as a parameter to define the density of destinations. Different levels of D (5, 11, and 22), the diversity of the orders' destinations are controlled. D does not represent the exact number of destinations

Table 2. The using levels of parameters for instance generation

Parameter	Level		
I	10	20	30
B	10	70	
D	5	11	22

in the instance. It is defined as the upper bound of the number of destinations. For the smaller level of D provides to assign more orders to the same destination, while the larger level of D assigns orders mostly to different destinations.

Table 2 shows the levels of data used in the instances used in the study. For each combination of parameters, there are 10 different instances. Therefore, we have a total of 180 instances. Using this data, two different solution approaches are proposed to solve the problem. We use dedicated computers for each of these two approaches.

4.1 *Exact Approach*

As indicated in section 3.3.2, the suggested bi-objective MILP is solved using the augmented ε-constraint approach. Only the instances that have 10 orders are solved utilizing the augmented ε-constraint approach, because the method cannot solve larger-sized instances in a reasonable time. The proposed MOMILP is developed in C++ programming language on Microsoft Visual Studio 2019 and solved in IBM ILOG CPLEX 12.8. We performed all the experiments for MOMILP on Windows 10 Pro 64-bit Intel Xeon Gold 6138 with 2.00 GHz processor computer with 20 GB RAM.

Minimizing the total cost is considered an objective function where the total cost is a constraint. We prefer to obtain Pareto-optimal solutions with this structure. Because our distance is an integer while the cost is a decimal number. Thus, we use the slack variable $s = 1$ because we use the integer constraint instead of determining the best value of the slack variable when the total cost is a constraint. We also used value of ε level as 10^{-3}. The time limit is set to 3 hours for each iteration because the exact solution approach takes a long time, even for small-sized instances. Therefore, the solutions found do not guarantee optimality. Total time varies according to the number of non-dominated points on the efficient frontier. This time limit given for the problem with daily operation planning is excessive for real-life scenarios. However, we applied this method to ensure that the exact solution is insufficient to solve this problem and that we can compare the proposed heuristic method to approximate solutions to the optimal solutions.

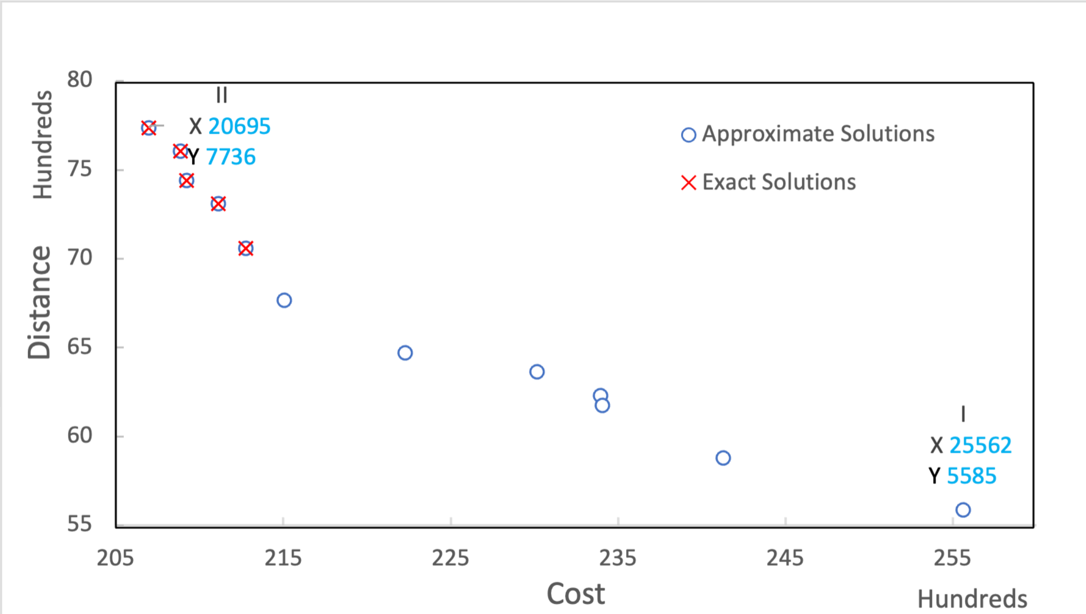

Figure 4. Non-dominated solutions obtained by augmented ε-constraint approach

Figure 4 gives an illustrative example of the non-dominated solutions obtained by the augmented ε-constraint approach for a small-sized instance with parameters of $I = 10$, $B = 10$ and $D = 5$. The model has difficulty in finding optimal results with the more restriction on distance. In the iterations where the right-hand side value of the distance constraint is greater, the exact solution can be obtained in a short time. As the distance is restricted with a smaller lower bound, routing becomes more important, and this increases the solution time exponentially. Table 3 shows the total cost, total distance, and GAP values of each point for the same instance. In the results found, it is seen that the GAP grows as the total distance value decreases within the three-hour time limit.

Table 3. Non-dominated solutions generated by augmented ε-constraint method

Total cost	**Total distance**	**GAP**
20695	7736	-
20884	7605	-
20921	7441	-
21111	7310	-
21276	7060	-
21504	6765	1.60%
22226	6470	4.41%
23016	6365	7.72%
23396	6230	9.30%
23407	6175	9.10%
24129	5880	11.12%
25562	5585	15.92%

To be able to represent numerically the impacts of objective functions on each other, two solutions from the solution set that are the furthest apart, as shown in Figure 4, are chosen and compared. The comparison of these selected solutions in the non-dominated solution set is shown in Table 4. Based on the results, the decreasing the total cost in 19.03% lead to an increase the total distance in 27.80%. Due to the results, the overall cost and overall distance do not have a strict inverse relationship with each other. Although the cost increases for a smaller distance value, the percentages of change are not very high.

Table 4. Comparison of two extreme points in a non-dominated solution set

	Solution I	**Solution II**	**Percentage of change**
Total cost	25562	20695	19.03%
Total distance	5585	7736	27.80%

Table 5 shows the results of the augmented ε-constraint method, which is implemented only for small-sized instances. Ten different instances are used for each combination of parameters B and D where $I = 10$. As a result, the method was applied to 600 different samples. The found values are the average numbers of these 10 instances within the time limit of 10800 seconds for each iteration. The average number of exact and approximate efficient points and average GAP values are indicated for each instance of B and D combination. In addition to the efficient points that could not be found exactly, a feasible solution could not be found within the given time limit for some examples. In other words, the results found contain missing efficient points.

As a result, even over a long time period, results are produced with significant GAP values. For instances with more than 10 orders, feasible results cannot be produced within the given time limit. This situation has revealed that the exact solution approach for the problem is quite inadequate. So, we propose a heuristic that can find efficient results in an acceptable time. The next section shows comparisons with the results found by the augmented method for small-sized instances. Also, for larger-sized instances, we report the performance of the approach using different performance metrics.

Table 5. Experimental results of augmented ε-constraint method for instances with $I = 10$

B	D	Avg. # of efficient points		Avg. GAP
		Avg. # of exact efficient points	Avg. # of approximate efficient points	
10	5	4.3	7.2	4.21%
	11	2.6	7.0	5.31%
	22	2.6	9.6	6.82%
70	5	5.7	7.3	7.92%
	11	4.6	10.0	6.95%
	22	3.2	16.8	13.21%

4.2 *Heuristic Approach*

As indicated in section 3.4.2, the proposed multi-objective VNS is used to solve SCDP for all existing instances that we have. The proposed MOVNS is developed in C++ programming language on Microsoft Visual Studio 2019. The mathematical model developed for local search is solved by the CPLEX solver with the callable library. We carried out all tests for MOVNS on Windows 7 Ultimate 64-bit Intel Core i5-4210U with 1.70 GHz processor computer with 4 GB RAM.

We apply the VNS heuristic for 240 instances that we introduced the parameters at the beginning of the chapter. Also, we use some other parameters which is specific with the VNS heuristic. For the stopping condition, we used the maximum number of iterations. First, we identified the number of iterations as 80 for each instance. Then we tested each of the instances according to their parameter I. A random sample is selected from each combination of B and D parameters for the experiments. In other words, the experimental results were calculated according to the average of the results obtained from six different samples for each level of I. In the experiments, it is observed that after a certain number of iterations, there is no significant change on the

Table 6. Average number of iterations with the latest change in the efficient frontier

	Avg # of iterations
***I*=10**	36
***I*=20**	67
***I*=30**	80

efficient frontier. According to the results of these experiments, the maximum number of iterations for instances with $I = 10$, 20, and 30 is determined as 40, 70, and 80, respectively. Table 6 shows the average for instances with different values of each I parameter, after which iteration no change is observed in the efficient frontier.

Furthermore, we used the C_{min} and D_{min} values given in equation (3.41) as parameters for MILP used in the local search procedure. Note that, w_1 and w_2 values may affect the results of local search phase and the solution quality of VNS algorithm. After conducting preliminary experiments, we assigned equal weights to both objectives. For the C_{min} and D_{min} values for the instances where the I parameter is 10, the results found from solving the mathematical model suggested in Section 3.2 for each objective function separately are used. For instances with $I = 20$, and 30, the minimum total cost value among the solutions obtained by VNS in the initial solution is calculated as C_{min} and the minimum total distance value as D_{min}.

To analyze the VNS heuristic's performance, we compare its non-dominated set with the augmented ε-constraint approach utilizing three performance measures proposed by Czyzżak and Jaszkiewicz (1998): *percentage, dist1,* and *dist2*.

We do not know the exact Pareto frontier of the problem for instances with more than 10 orders. Even for instances with 10 orders, the exact approach could not find all exact non-dominated points. To measure the performance of the suggested algorithm, we defined a new reference set that was proposed by Soylu (2007). A Non-Dominated Union Set (NDUS) is composed, which comprises the non-dominated solutions of all algorithms. Hence, the proximity indicator can be calculated according to NDUS.

Assume R is our reference set of non-dominated points that is generated with the augmented ε-constraint approach or with the union non-dominated set that is generated by VNS with 10 different random seeds respectively, and V is the set of solutions generated by the VNS algorithm. The following are the definitions and formulations of these performance measures:

The first performance measurement is *percentage*, and it represents as the percentage of non-dominated points that are found in both the reference set and the heuristic algorithm. To compare the small-sized instances with the exact solution method, the reference set of R is used the non-dominated solution set of the augmented ε-constraint method. For the rest of the comparisons, the reference set of R is used the union non-dominated solution set of VNS with different random seeds.

$$percentage = \frac{card\{V \cap R\}}{card\{R\}} \times 100\%$$

Second performance metric is *dist1* and it is defined as the average distance between a solution point of $x \in R$ and the closest solution point of $y \in V$.

$$dist1 = \frac{1}{card\{R\}} \sum_{y \in R} \{\min_{x \in V} \{c(x,y)\}\}$$

Last performance metric is *dist2* and it is defined as the average maximum between a solution point of $x \in R$ and the closest solution point of $y \in V$.

$$dist2 = \max_{y \in R} \{\min_{x \in V} \{c(x,y)\}\}$$

Czyzżak and Jaszkiewicz (1998) propose using the following achievement-scalarizing function to determine the distance between solutions x and y, $c(x,y)$:

$$c(x,y) = \max_{j} \{0, \omega_j (f_j(y) - f_j(x))\}$$

Hereby, if x approaches the value of solution y on all objectives, the measure becomes zero. Otherwise, the maximum weighted difference from y for each objective is used. The weighting factor ω_j is computed as follows where Δ_j represents f_j's range in R:

$$\omega_j = \frac{1}{\Delta_j}$$

While the *dist1* and *dist2* values decrease, the more the heuristic method's estimated efficient frontier approaches the set R. On the other hand, a higher *percentage* value means that the performance of the solution approach is superior.

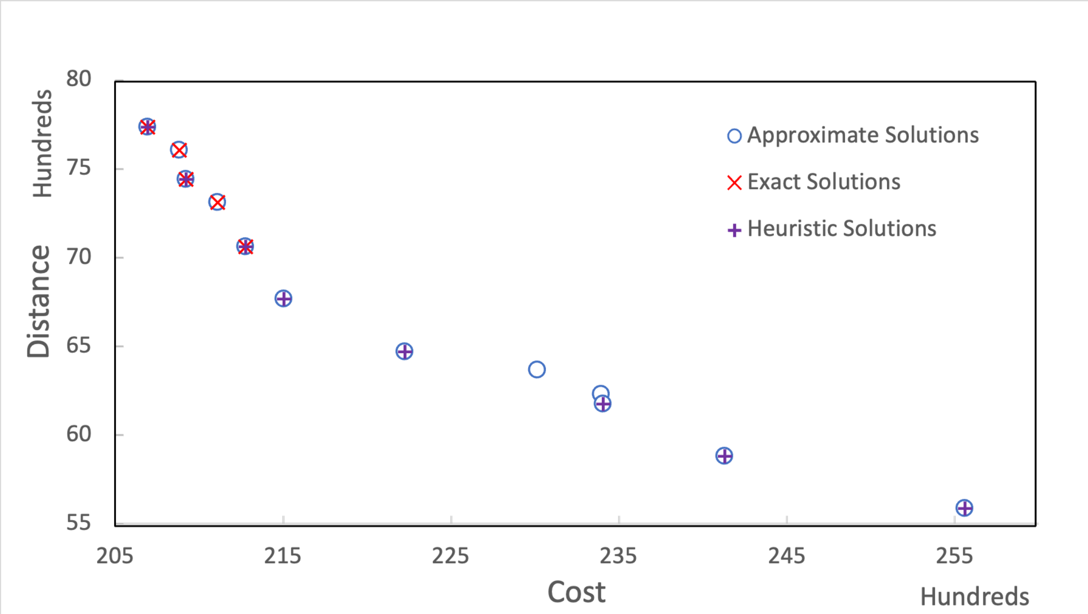

Figure 5. Non-dominated solutions obtained by augmented ε-constraint method and multi-objective VNS algorithm

Figure 5 is the illustrative example of the non-dominated solutions obtain via the augmented ε-constraint approach and multi-objective VNS algorithm for a small-sized instance with parameters of $I = 10$, $B = 10$ and $D = 5$. The used performance metrics values were computed as *percentage* = 66.67, *dist1* = 0.020, *dist2* = 0.080. The performance of VNS heuristics seems not so high according to the performance metric *percentage*. Because efficient frontier does not have so many points. It decreases the *percentage* whereas, VNS algorithm detects 8 non-dominated points out of 12 in 379 s. Figure 6 illustrates the same instance with Figure 5. In Figure 6, green symbols show the union efficient set of the results obtained by VNS with 10 different random seeds. The used performance metrics values were computed as *percentage* = 75.00, *dist1* = 0.013, *dist2* = 0.061. In solutions with different random seeds, performance metrics get better.

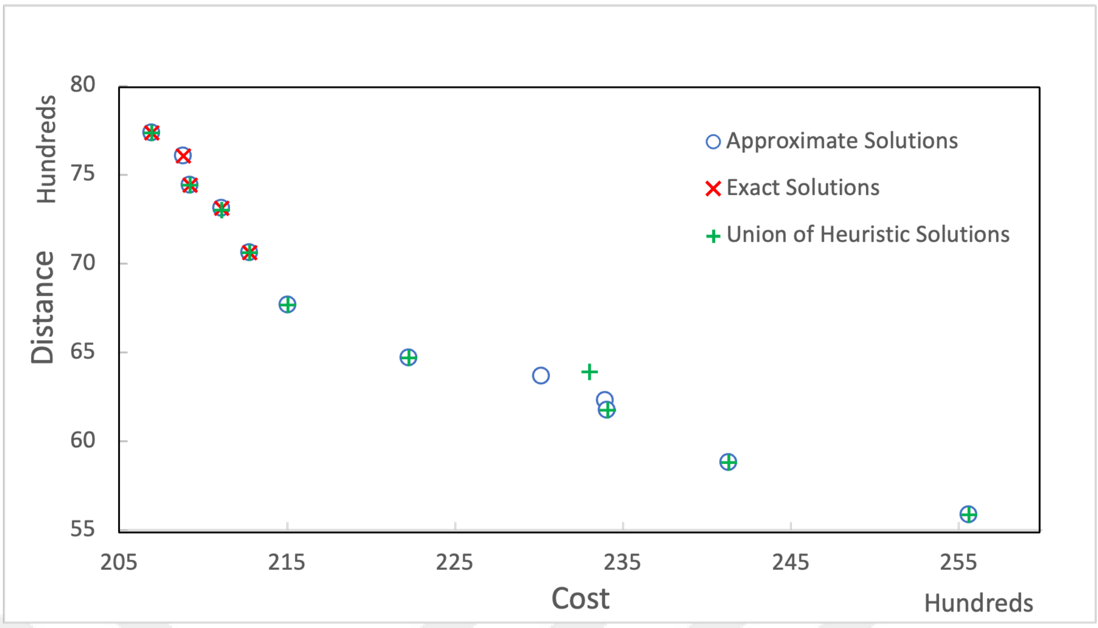

Figure 6. Non-dominated solutions obtained via augmented ε-constraint approach and the union non-dominated solution set of multi-objective VNS algorithm with different random seeds

When we compare Figure 5 and Figure 6, we observe that there is another non-dominated solution in the union of heuristic solutions in Figure 6, in addition to the non-dominated solutions obtained by the heuristic result in Figure 5. In Figure 6, it is seen that the union of heuristic solutions is an efficient frontier that is very close to the efficient frontier obtained by the mathematical model. As expected, the union heuristic set from replications with different random seeds achieved better results in all performance metrics than the efficient points of the heuristic algorithm run with a single random seed.

Table 7. Results of Variable Neighborhood Search

I	B	D	CPU (s)	|V|	|NDUS|	P	dist1	dist2
10	10	5	81.6	5.8	7.8	82.47	0.032	0.139
		11	206.4	7.8	10.1	69.42	0.020	0.052
		22	1529.4	13.3	16.8	67.35	0.007	0.021
	70	5	208.4	4.8	6.6	65.80	0.190	0.454
		11	133.4	3.7	4.7	57.63	0.213	0.509
		22	1215.7	10.5	15.0	51.75	0.036	0.078
20	10	5	985.6	13.8	16.4	63.77	0.076	0.117
		11	2500.1	7.4	13.0	55.54	0.021	0.043
		22	6094.8	10.1	13.4	53.12	0.036	0.138
	70	5	448.3	9.3	10.7	52.35	0.141	0.372
		11	1900.2	7.4	10.8	32.30	0.189	0.691
		22	-	-	-	-	-	-
30	10	5	1207.7	18.0	20.5	52.41	0.108	0.599
		11	7367.6	16.6	19.2	45.53	0.067	0.152
		22	-	-	-	-	-	-
	70	5	5835.1	9.0	10.5	42.50	0.358	1.341
		11	9815.3	11.1	14.3	30.8	0.819	2.373
		22	-	-	-	-	-	-

The performance evaluation of the VNS is shown according to the NDUS in Table 7. To determine the NDUS, we merge the latest populations of all runs, then detect the non-dominated solutions of the total population. Table 7 shows the average CPU for 10 samples with 10 replication, average number of efficient solutions in set V, and *percentage* (P), *dist1* and *dist2* values for the instances with NDUS for each parameter combination. As can be seen in Table 7, as the I and D parameters increase in the problem, the CPU time increases, while parameter B has no significant effect on the CPU time. A significant portion of CPU time is due to local search phase. The values in the percentage performance criterion are not very high because of the low number of non-dominated points in the efficient frontier. The inability to find only a few points precisely reduces the percentage. On the other hand, although the non-dominated one in NDUS cannot be found exactly, the small values of $dist1$ and $dist2$ indicate that there is a point very close to it. This shows that the algorithm works with good performance by obtaining efficient results.

CHAPTER 5 : CONCLUSION

In this thesis, we consider shipment consolidation and dispatching problem with two objectives. Our aim is to minimize the overall cost while minimizing the overall distance. We define the problem with its assumptions. Then we propose a multi-objective mathematical model. We use the augmented ε-constraint method to generate the non-dominated points. The experimental results show that the exact approach needs long computational times to solve even small-sized problems. Since our problem is based on daily operation planning in real life, it is very important to solve the problem in a short time. Therefore, we suggest a multi-objective VNS heuristic to obtain the approximate Pareto frontier in an acceptable time.

To analyse the performance of our proposed solution methods, we use the data set provided by Tokcaer (2018). The data set consists of a total of 180 instances. This data set is diversified over three different parameters that are the number of orders, elasticity, and diversity of destinations. Using this data, we find the non-dominated solutions of the problem with using the augmented ε-constraint approach and VNS algorithm. Comparative results are illustrated on an small-sized instance. Since the exact approach fails to find the non-dominated solutions in a reasonable time, a non-dominated union set of VNS is used as a reference set. To measure the performance of the VNS heuristic, three different measurements are used which are the percentage of non-dominated points that are found in both the reference set and the heuristic algorithm, the average and maximum distances between non-dominated solutions in reference set and estimated solutions. Results of the experiments show that the proposed multi-objective VNS heuristic performs well and obtains the solutions in a reasonable time.

Several future research directions may be addressed for this problem. Firstly, the problem can be handled as multi-modal by adding different transportation modes such as road, rail, air, or waterway. Also, with this innovation, the model can be adapted to real world problems by adding the constraint that some orders cannot be carried by some modes. Moreover, the number of handling operations, which is an important process for multi-modal problems, can be added to the model as a new objective as Tokcaer and Özpeynirci (2018) considered in their study. We may also consider the impact of shipment plans on the environment. The mode and transshipment terminal related decisions and the distance travelled affect the total carbon emission of the operation, which can be considered as another objective.

Currently, we are solving the routing problem in local search phase optimally, which requires a high computational effort. Instead, a heuristic may be applied in local search or different versions of VNS may be considered with the aim of saving computational time.

Finally, in this study, we created efficient solutions that the decision-maker to select the most preferred among them. However, which one of these efficient solutions should be selected is not in the scope of our study. The making the decision among the efficient solutions can be difficult and requires analytical solution methods. Thus, as a future work, multi-criteria decision support methods can be implemented to non-dominated solutions found. In this manner, decision-makers can choose their final and most desired solution from a set of various efficient solution points.

REFERENCES

Alfieri, A., Castiglione, C. and Pastore, E. (2020), *A multi-objective tabu search algorithm for product portfolio selection: A case study in the automotive industry*, Computers & Industrial Engineering [Online]. Available at: https://www.sciencedirect.com/science/article/pii/S0360835220301169 (Accessed: 28 June 2022) .

Attanasio, A., Fuduli, A., Ghiani, G. and Triki, C. (2007), *Integrated shipment dispatching and packing problems: a case study*, Journal of Mathematical Modelling and Algorithms 6(1), pp. 77–85.

Azadnia, A. H., Saman, M. Z. M. and Wong, K. Y. (2015), *Sustainable supplier selection and order lot-sizing: an integrated multi-objective decision-making process*, International Journal of Production Research 53(2), pp. 383–408.

Bouziaren, S. A. and Aghezzaf, B. (2018), *An improved augmented ε-constraint and branch-and-cut method to solve the TSP with profits*, IEEE Transactions on Intelligent Transportation Systems 20(1), pp. 195–204.

Brennan, J. J. (1982), *Models and Analysis Of Temporal Consolidation*. Unpublished Doctoral Thesis. University of California.

Campbell, J. F. (1990), *Freight consolidation and routing with transportation economies of scale*, Transportation Research Part B: Methodological 24(5), pp. 345–361.

Cetinkaya, S. and Lee, C.-Y. (2000), *Stock replenishment and shipment scheduling for vendor-managed inventory systems*, Management Science 46(2), pp. 217–232.

Crainic, T. G. (2000), *Service network design in freight transportation*, European Journal of Operational Research 122(2), pp. 272–288.

Crainic, T. G. and Laporte, G. (1997), *Planning models for freight transportation*, European Journal of Operational Research 97(3), pp. 409–438.

Czyzżak, P. and Jaszkiewicz, A. (1998), *Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization*, Journal of Multi-Criteria Decision Analysis 7(1), pp. 34–47.

Daganzo, C. F. (1988), *Shipment composition enhancement at a consolidation center*, Transportation Research Part B: Methodological 22(2), pp. 103–124.

Duarte, A., Pantrigo, J. J., Pardo, E. G. and Mladenovic, N. (2015), *Multi-objective variable neighborhood search: an application to combinatorial optimization problems*, Journal of Global Optimization 63(3), pp. 515–536.

Fleszar, K. and Hindi, K. S. (2008), *An effective VNS for the capacitated p-median problem*, European Journal of Operational Research 191(3), pp. 612–622.

Geiger, M. J. (2006), *Randomised variable neighbourhood search for multi objective optimisation*, 4th EU/ME: Design and Evaluation of Advanced Hybrid Meta-Heuristics pp. 34–42.

Guastaroba, G., Speranza, M. G. and Vigo, D. (2016), *Intermediate facilities in freight transportation planning: a survey*, Transportation Science 50(3), pp. 763–789.

Gupta, Y. P. and Bagchi, P. K. (1987), *Inbound freight consolidation under just-in-time procuremen*, Journal of Business Logistics 8(2), pp. 74–94.

Hansen, P. and Mladenović, N. (1997), *Variable neighborhood search for the p-median*, Location Science 5(4), pp. 207–226.

Hansen, P., Mladenović, N. and Moreno Perez, J. A. (2010), *Variable neighbourhood search: methods and applications*, Annals of Operations Research 175(1), pp. 367–407.

Hansen, P., Mladenović, N., Todosijević, R. and Hanafi, S. (2017), *Variable neighborhood search: basics and variants*, EURO Journal on Computational Optimization 5(3), pp. 423–454.

Hemmelmayr, V. C., Doerner, K. F. and Hartl, R. F. (2009), *A variable neighborhood search heuristic for periodic routing problems*, European Journal of Operational Research 195(3), pp. 791–802.

Hemmelmayr, V., Schmid, V. and Blum, C. (2012), *Variable neighbourhood search for the variable sized bin packing problem*, Computers & Operations Research 39(5), pp. 1097–1108.

Higginson, J. and Bookbinder, J. H. (1994), *Policy recommendations for a shipment-consolidation program*, Journal of Business Logistics 15(1), pp. 87–112.

Hu, B. and Raidl, G. R. (2008), *Effective neighborhood structures for the generalized traveling salesman problem*, *in* 'European Conference on Evolutionary Computation in Combinatorial Optimization', Springer, pp. 36–47.

Hwang, C.-L. and Masud, A. S. M. (2012), *Multiple objective decision making—methods and applications: a state-of-the-art survey*, Vol. 164, Springer Science & Business Media.

Koca, E. and Yıldırım, E. A. (2012), *A hierarchical solution approach for a multicommodity distribution problem under a special cost structure*, Computers & Operations Research 39(11), pp. 2612–2624.

Kocatürk, F. and Özpeynirci, Ö. (2014), *Variable neighborhood search for the pharmacy duty scheduling problem*, Computers & Operations Research 51, pp. 218–226.

Köksalan, M. and Wallenius, J. (2012), *Multiple criteria decision making: foundations and some approaches*, *in* 'New Directions in Informatics, Optimization, Logistics, and Production', INFORMS, pp. 171–183.

Masters, J. M. (1980), *The effects of freight consolidation on customer service*, Journal of Business Logistics 2(1), pp. 55–74.

Mavrotas, G. (2009), *Effective implementation of the ε-constraint method in multi-objective mathematical programming problems*, Applied Mathematics and Computation 213(2), pp. 455–465.

Miller, C. E., Tucker, A. W. and Zemlin, R. A. (1960), *Integer programming formulation of traveling salesman problems*, Journal of the ACM (JACM) 7(4), pp. 326–329.

Min, H. (1996), *Consolidation terminal location-allocation and consolidated routing problems*, Journal of Business Logistics 17(2), pp. 235–263.

Min, H. and Cooper, M. (1990), *A comparative review of analytical studies on freight consolidation and backhauling*, Logistics and Transportation Review 26(2), pp. 149–169.

Minkoff, A. S. (1993), *A Markov decision model and decomposition heuristic for dynamic vehicle dispatching*, Operations Research 41(1), pp. 77–90.

Mrkela, L. and Stanimirović, Z. (2022), *A Multi-objective variable neighborhood search for the maximal covering location problem with customer preferences*, Cluster Computing pp. 1–17.

Özpeynirci, S., Özpeynirci, Ö. and Mousseau, V. (2022), *Portfolio decision analysis with a generalized balance approach*, Computers & Operations Research 142 [Online]. Available at: https://www.sciencedirect.com/science/article/pii/S0305054822000090 (Accessed: 28 June 2022).

Powell, W. B. (1985), *Analysis of vehicle holding and cancellation strategies in bulk arrival, bulk service queues*, Transportation Science 19(4), pp. 352–377.

Puchinger, J. and Raidl, G. R. (2008), *Bringing order into the neighborhoods: relaxation guided variable neighborhood search*, Journal of Heuristics 14(5), pp. 457–472.

Resat, H. G. and Unsal, B. (2019), *A novel multi-objective optimization approach for sustainable supply chain: A case study in packaging industry*, Sustainable Production and Consumption 20, pp. 29–39.

Ripon, K. S. N., Glette, K., Khan, K. N., Hovin, M. and Torresen, J. (2013), *Adaptive variable neighborhood search for solving multi-objective facility layout problems with unequal area facilities*, Swarm and Evolutionary Computation 8, pp. 1–12.

Rivera, G., Coello, C. A. C., Cruz-Reyes, L., Fernandez, E. R., Gomez-Santillan, C. and Rangel-Valdez, N. (2022), *Preference incorporation into many-objective optimization: An Ant colony algorithm based on interval outranking*, Swarm and Evolutionary Computation 69 [Online]. Available at: https://www.sciencedirect.com/science/article/pii/S2210650221001863 (Accessed: 28 June 2022).

Sadati, M. E. H. and Çatay, B. (2021), *A hybrid variable neighborhood search approach for the multi-depot green vehicle routing problem*, Transportation Research Part E: Logistics and Transportation Review 149 [Online]. Available at: https://www.sciencedirect.com/science/article/pii/S1366554521000673 (Accessed: 28 June 2022).

Sadjadi, S. J., Heidari, M. and Alinezhad Esboei, A. (2014), *Augmented ε-constraint method in multiobjective staff scheduling problem: a case study*, The International Journal of Advanced Manufacturing Technology 70(5), pp. 1505–1514.

Schmidt, G. and Wilhelm, W. E. (2000), *Strategic, tactical and operational decisions in multi-national logistics networks: a review and discussion of modelling issues*, International Journal of Production Research 38(7), pp. 1501–1523.

Soylu, B. (2007), *An evolutionary algorithm for multiple criteria problems*. Unpublished Doctoral Thesis. Middle East Technical University.

Soylu, B. and Katip, H. (2019), *A multiobjective hub-airport location problem for an airline network design*, European Journal of Operational Research 277(2), pp. 412–425.

Steuer, R. E. (1986), *Multiple criteria optimization*, Wiley.

Tamby, S. and Vanderpooten, D. (2021), *Enumeration of the nondominated set of multiobjective discrete optimization problems*, INFORMS Journal on Computing 33(1), pp. 72–85.

Tokcaer, S. (2018), *Shipment consolidation and dispatching problem with transshipment terminals*. Unpublished Doctoral Thesis. Izmir University of Economics.

Tokcaer, S. and Özpeynirci, Ö. (2018), *A bi-objective multimodal transportation planning problem with an application to a petrochemical ethylene manufacturer*, Maritime Economics & Logistics 20(1), pp. 72–88.

Ülkü, M. A. (2012), *Dare to care: Shipment consolidation reduces not only costs, but also environmental damage*, International Journal of Production Economics 139(2), pp. 438–446.

Xu, X., Li, J. and Zhou, M. (2021), *Bi-objective colored traveling salesman problems*, IEEE Transactions on Intelligent Transportation Systems 142 [Online]. Available at: https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9457025 (Accessed: 28 June 2022).

Yu, H. and Solvang, W. D. (2016), *An improved multi-objective programming with augmented ε-constraint method for hazardous waste location-routing problems*, International Journal of Environmental Research and Public Health 13(6), pp. 548–569.

Yücel, E., Salman, F. S. and Erdoğan, G. (2022), *Optimizing two-dimensional vehicle loading and dispatching decisions in freight logistics*, European Journal of Operational Research 302(3), pp. 954–969.

Zhu, S. (2022), *Multi-objective route planning problem for cycle-tourists*, Transportation Letters 14(3), pp. 298–306.

Zhu, S. and Zhu, F. (2020), *Multi-objective bike-way network design problem with space–time accessibility constraint*, Transportation 47(5), pp. 2479–2503.

Zobolas, G., Tarantilis, C. D. and Ioannou, G. (2009), *A hybrid evolutionary algorithm for the job shop scheduling problem*, Journal of the Operational Research Society 60(2), pp. 221–235.