DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

AN INTEGRATED APPROACH TO VEHICLE ROUTING AND MULTI-DIMENSIONAL PACKING PROBLEMS

by
Mustafa KÜÇÜK

December, 2023
İZMİR

AN INTEGRATED APPROACH TO VEHICLE ROUTING AND MULTI-DIMENSIONAL PACKING PROBLEMS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy in Department of Industrial Engineering, Industrial Engineering Program

by
Mustafa KÜÇÜK

December, 2023
İZMİR

Ph.D. THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"AN INTEGRATED APPROACH TO VEHICLE ROUTING AND MULTI-DIMENSIONAL PACKING PROBLEMS"** completed by **MUSTAFA KÜÇÜK** under supervision of **PROF.DR. ŞEYDA A. YILDIZ** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Doctor of Philosophy.

....

Prof.Dr. Şeyda A. YILDIZ

Supervisor

....

Prof. Dr. Efendi NASİBOĞLU

Thesis Committee Member

....

Assoc. Prof. Dr. Şener AKPINAR

Thesis Committee Member

....

Prof. Dr. Pınar Mızrak ÖZFIRAT

Examining Committee Member

....

Assoc. Prof. Dr. Damla KIZILAY

Examining Committee Member

Prof. Dr. Okan FISTIKOĞLU

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGMENT

Writing this thesis was only possible with the assistance of several individuals and institutions, to whom I would like to express my profound appreciation. Firstly, I would like to extend my gratitude to my supervisor, Prof. Dr. Şeyda A. YILDIZ, for her unwavering support, guidance, and understanding. Not only did she serve as my supervisor, but she also provided encouragement throughout my graduate program. She allowed me the freedom to conduct my research while assisting me in overcoming any obstacles and providing advice. I am grateful for the guidance and support of my supervisor throughout the years, and for every meeting and conversation we shared. I see this PhD as the start of numerous future collaborations. My sincere thanks to the other members of my thesis committee, Prof. Dr. Efendi NASİBOĞLU and Doç. Dr. Şener AKPINAR, for their constructive feedback that improved the quality of this thesis.

Finally, I would like to express my gratitude to my family, Esma, Görkem, and Gökçen KÜÇÜK, for their unwavering love and support. They have consistently encouraged and assisted me in my academic pursuits, bolstered my decisions, and placed faith in me even beyond my own.

Mustafa KÜÇÜK

AN INTEGRATED APPROACH TO VEHICLE ROUTING AND MULTI-DIMENSIONAL PACKING PROBLEMS

ABSTRACT

In this thesis, in the context of the transportation of goods, we propose a set of algorithms regarding real combinatorial optimization problems called The Three-Dimensional Loading Capacitated Vehicle Routing Problems (3L-CVRP). The 3L-CVRP is an integrated routing problem combining two well-known optimization problems, Vehicle Routing Problem (VRP) and Container Loading Problem (CLP). This problem is solved by two main approaches: Constraint Programming (CP) and a new elitism-based Evolutionary Algorithm (EA).

The former approach employs constraints for reducing the set of values each variable can assume, and we present a first CP strategy for this problem via varied constructive techniques aiming for optimal solutions. The latter method involves an Evolution Strategy heuristic, which is a subfield of EA. Elitism serves as an operational feature of Evolutionary Algorithms (EAs) that effectively reduces genetic drift. It works by allowing the most competent individuals in the population to pass on their traits to the following generation. Additionally, we enhance the traditional 3L-CVRP version by incorporating additional loading constraints to tackle pickup and delivery problems.

All proposed methods underwent implementation and experimental testing. To ensure precision and clarity, a comprehensive suite of computational tests was executed using numerous benchmark instances. When required, both large and small benchmark instances were adapted to evaluate model quality. All the results obtained have subsequently been presented and discussed.

Keywords: Routing, Three-dimensional Loading, Constraint Programming, Elitism, Evolution Strategy

ARAÇ ROTALAMA VE ÇOK BOYUTLU YÜKLEME PROBLEMLERİNE ENTEGRE BİR YAKLAŞIM

ÖZ

Bu tezde, Üç Boyutlu Yükleme Kapasiteli Araç Rotalama Problemleri (3L-CVRP) olarak adlandırılan gerçek hayatta sıklıkla karşılaşılan kombinatoryal optimizasyon problemlerine ilişkin bir dizi algoritma öneriyoruz. 3L-CVRP, iyi bilinen iki optimizasyon problemini, Araç Rotalama Problemi (VRP) ve Konteyner Yükleme Problemini (CLP) birleştiren entegre bir rotalama problemidir. Bu problem iki ana yaklaşımla çözülür: Kısıtlama Programlaması (CP) ve yeni bir elitizm tabanlı Evrimsel Algoritma (EA).

CP'ye dayanan çözüm yaklaşımında, her değişkenin alabileceği değer kümesini azaltmak için kısıtlamalar kullanılarak en iyi çözüme ulaşılmaya çalışılır. Bildiğimiz kadarıyla, bu problemin çözümünde optimal çözümlere ulaşmak için farklı arama stratejileri aracılığıyla ilk kez CP yaklaşımı uygulanıyor. İkinci yaklaşım, EA'nın bir alt dalı olan Evrim Stratejileri olarak bilinen bir sezgisel yaklaşımdır. Ayrıca, EA'ların operasyonel bir özelliği olarak elitizm, popülasyondaki en iyi bireylerin özelliklerini bir sonraki nesle aktarmalarını sağlayarak genetik sürüklenmeyi azaltmanın bir yolunu sunar. Böylece, daha iyi sonuçlara kısa sürede ulaşılması sağlanır. Çözüm yaklaşımlarına ilave olarak, 3L-CVRP'nin klasik versiyonunu, teslim alma ve dağıtım problem olarak genelleştirilmiş ve daha fazla yükleme kısıtlamasını göz önünde bulundurarak çözülmüştür.

Önerilen tüm yöntemler, bilgisayar ortamında kodlanarak deneysel olarak test edildi. Bu testler, literatürde bu amaçla sunulan kapsamlı bir dizi kıyaslama örneği kullanılarak gerçekleştirildi. Ayrıca, önerilen modellerin kalitesini daha detaylı değerlendirebilmek için büyük boyutlu ve küçük boyutlu kıyaslama örnekleri oluşturulmuştur. Elde edilen tüm sonuçlar önerilen çözüm yöntemlerinin etkinliğini ortaya koymaktadır.

Anahtar kelimeler: Yönlendirme, Üç Boyutlu Yükleme, Kısıt Programlama, Elitizm, Evrimsel Strateji

CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

CHAPTER ONE

INTRODUCTION

1.1 The Problem and Its Significance/Importance

Today, the transportation sector is one of the most important economic and social life elements. Every step of the classical supply chain process is interconnected through shipping. Raw materials are transported from suppliers to production sites, and manufactured products are transported to end customers. On the other hand, owing to e-commerce, which has grown in recent years, the flow of materials from sellers to consumers has become the center of social life. Moreover, E-commerce grew exponentially with the COVID-19 pandemic that started in 2020, forcing all sectors to adapt to this new trend.

This emerging condition has increased the critical role of the vehicle routing problem in transportation and operations research. The proposed approaches to this problem are relevant not only for the scientific community but also for real-world situations because they significantly impact the final cost of goods and commodities. Moreover, the applications of the vehicle routing problem, first proposed by Dantzig and Ramsey (1959) as a generalization of the Traveling Salesman Problem (TSP), are not limited to logistics. The conceptual model expressed in this problem has applications in various fields, including computer networks and communication systems.

In general, the Vehicle Routing Problem (VRP) consists of finding the best set of routes for a fleet of vehicles that need to visit several customers, considering mandatory operational constraints such as the maximum uptime of drivers and limitations from customers and vehicles.

Additional constraints can be added to this problem, such as the capacity of the vehicles, which is one of the most common constraints. However, this constraint is insufficient for reflecting the complexity of real-world systems. The maximum weight the vehicles can carry is generally known as the capacity constraint, and each load that needs to be transported affects the solution in proportion to its weight. However, its weight may not be sufficient for describing the load. In particular, the physical properties of the load are one of the issues that should be prioritized when placing it in the vehicle compartment; for example, the weight of a load may be below the vehicle's capacity but cannot be placed in a compartment due to its dimensions. Taking into account both the volume of the compartment and the load is necessary, and even this may not be sufficient for a group of loads. The fact that the total volume of the loads is smaller than the compartment volume does not indicate that the loads can be placed. Therefore, we explicitly assume the loads to be three-dimensional cuboid items to provide a more realistic model of practically relevant optimization problems. As a result, the VRP is modeled and solved by considering the three-dimensional loading constraints in this thesis.

1.2 Research Objectives

This thesis has two complementary objectives: problem- and solution-oriented. The first objective of this thesis is to define a realistic optimization problem applied to the field of transportation and supply chain management in general. Therefore, the focus is on the actual constraints applied to these problems. VRPs typically neglect the shapes of the loads that relate to the load components of these problems. When the loads are small relative to the dimensions of the vehicles, they can be characterized by a single measure, such as weight or volume. When a single value limits the capacities of the vehicles or when the loads are substantial compared to the dimensions of the vehicles, deciding how to place them in the vehicle is an essential part of the problem. The optimization problem considered to address these emerging issues is referred to as the loading-constrained vehicle routing problem in the literature. Combining two difficult combinatorial optimization problems creates a very difficult optimization problem involving vehicle routing and three-dimensional loading problems.

Another objective of this thesis is to develop methods for solving this integrated problem. The proposed solution methods are Mixed-Integer programming (MIP) and Constraint Programming (CP) models for small-sized problems, CP-based decomposed and hybrid models, and EA-based metaheuristic solutions for large-sized problems.

The routing and loading problems that comprise the thesis subject are examined in detail first. Various variants of these problems are discussed, and their constraints, frequently discussed in the literature, are emphasized. We then describe the Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP). MIP and CP models were formulated for this problem. Because these models can only solve small-sized problems, a decomposed model of MIP with CP and a hybrid model of CP with an Evolutionary Algorithm (EA) were proposed to solve the problem. The results indicate that the developed approaches can produce high-quality solutions in reasonable computation times. Finally, we study the Pickup and Delivery Problem with Three-Dimensional Loading Constraints (3L-PDP), a generalized variant of the 3L-CVRP. A novel elitism-based search method was developed to solve the 3L-PDP.

1.3 Contributions

The performance tests of the new method showed good results within a reasonable time. In this thesis, the following contributions have been made:

- CP has been used for the first time to formulate the non-overlapping constraints on the loading part of the 3L-CVRP. The concept of "interval variable," one of the essential elements in CP, was adapted to packing problems for the first time, and container and box edge dimensions were defined with interval variables.

- For the first time, a CP model has been developed for the 3L-CVRP by integrating the routing and loading parts of the problem with global constraints such as $Cumulative, Alternative$ and $presenceOf$.

- Large-scale problems cannot be solved using the proposed CP model in a reasonable computational time due to the complexity of the 3L-CVRP. To take advantage of CP's usefulness, decomposed and hybrid models were developed for this reason.

- A new meta-heuristic solution approach, the sine function-controlled elitism-based EA, named Sine Wave Optimization (SWO), has been developed for the 3L-CVRP, which offers a new perspective on population-based solution methods due to the influence of elites on the population. It has a flexible and simple structure that can be employed to solve integrated vehicle routing and loading problems. This metaheuristic solution, SWO, integrates the evolutionary strategy, a branch of evolutionary algorithms, and the variable elitism approach. The sine waveform was used to create a change in elitism with ascending and descending values on the timeline to balance exploration and exploitation.

- In order to address more realistic and practically relevant issues, more challenging loading constraints such as LIFO, weight distribution, and pickup/delivery were also considered simultaneously, referred to as the 3L-PDP. This problem is the most complex and realistic form of integrated routing and loading problems.

- Due to the complexity of the 3L-PDP, the SWO method, which was developed for the 3L-CVRP, has been modified to suit its solution. A new set of benchmark examples has been created for the 3L-PDP problem, in which "Weight Distribution" and "Time Window" constraints are taken into account in addition to those in the literature for the 3L-PDP.

1.4 Outline of the Thesis

This thesis involves seven chapters, including the present one, and is organized as follows.

Chapter 2 describes the problems addressed in this thesis and the context in which they arise. More precisely, the classical VRP and its variants are explained in detail, and the official problem definition is provided along with the MIP formulation of the problem. Subsequently, a general definition of the Pickup and Delivery Problem (PDP), the generalized version of the VRP, is presented. After that, the constraints of loading problems are categorized and explained. Finally, the end of the chapter contains a description of the VRP with loading constraints in two and three dimensions.

The 3L-CVRP literature is presented in Chapter 3, and the solution methods, problem characteristics, and constraints considered for each study are outlined.

In Chapter 4, to solve the 3L-CVRP, all constraints and objective functions are formulated in the context of CP, and five models are developed: two integrated, two decomposed CP-based, and one decomposed hybrid model. The hybrid model (CP&EA) uses a CP model for routing and an EA for loading. First, these models have been compared with a MIP model adapted from the literature by solving small-sized problems. Then, benchmark examples common in the literature have been used to test the performance of the hybrid model (CP&EA) on large-sized problems.

In Chapter 5, a new elitism-based EA is proposed for the 3L-CVRP. A detailed computational study of the method, which includes the elitism rate controlled by the sine function, is provided.

In Chapter 6, we extend the 3L-CVRP presented in the previous chapters, considering the pickup and delivery restrictions. After the problem formulation is given, a new EA approach with sine function-controlled elitism is proposed for its solution. Next, the computational results obtained for the 3L-PDP are presented.

In Chapter 7, we present the main contributions and conclusions of the work presented in this thesis and provide a perspective for further research.

CHAPTER TWO
VEHICLE ROUTING PROBLEMS WITH LOADING CONSTRAINTS

2.1 Introduction

This chapter explains the vehicle routing problems with loading constraints. The chapter is organized as follows; Section 2.2 briefly describes the vehicle routing problem. In section 2.3, variants of the vehicle routing problem are presented. Section 2.4 explains the pickup and delivery problems, while Section 2.5 describes the container loading problems and the constraints of the loading problems. Section 2.6 gives a brief explanation of the routing problems with loading constraints. Finally, the context of this chapter is summarized in Section 2.7.

2.2 The Vehicle Routing Problem

The Vehicle Routing Problem (VRP) was first introduced as the "truck dispatching problem" more than 60 years ago by Dantzig & Ramser (1959). It was about how a homogeneous fleet of trucks could meet the demand for oil with a minimum travel distance from a central hub to a series of gas stations and was a generalization of the well-known traveling salesman problem.

Five years later, Clarke & Wright (1964) generalized this problem to a linear optimization problem commonly encountered in logistics and transportation, of how to serve a range of customers geographically dispersed around the central warehouse using a fleet of trucks of varying capacities. This problem has come to be known as the 'Vehicle Routing Problem' (VRP), one of the most studied topics in the field of Operations Research (Braekers, Ramaekers, & Van Nieuwenhuyse, 2016). Since then, the VRP has been studied extensively, and many problem variations have been identified. Many books or book chapters are devoted to the VRP, its derivatives, and the exact and heuristic algorithms for its solution. Irnich, Toth & Vigo (2014) categorize them according to network characteristics, transport demands, intra-route and inter-route constraints, fleet characteristics, and optimization goals. Practical requirements and new challenges require comprehensive definitions and formulations of the VRP. For example, restrictions on driver working hours, visiting time intervals, traffic conditions, etc., can occur in real-world VRPs and enrich the definition and applications of VRPs.

Over the last six decades, numerous articles have been published in the literature of Operations Research and Transportation showcasing mathematical models and suggesting heuristic and exact algorithms for the optimal and approximate resolution of multiple iterations of the VRP. The international research community's keen interest in various VRP variations is attributable not solely to their combinatorial optimization complexity but also their practical applicability. As a consequence, scholars from both academic and industrial domains persevere in their efforts to address the issue with equal intensity.

2.2.1 The Mathematical Formulation of the VRP

The objective of the VRP is often to minimize the total cost of travel when delivering goods located in a central depot to customers who have placed orders. This objective may differ depending on the particular real-world application. For example, we may have to minimize the total cost of travel and the fixed costs associated with used vehicles and drivers simultaneously. In another case, not all customer demands can be met, and in such cases, some customers' demands may be reduced, or some customers may be left without service. To deal with this situation, a priority variable can be introduced for each customer, or a corresponding penalty for partial or incomplete service can be introduced for each customer. In this case, the objective function is defined as minimizing the penalties associated with inadequate service to the customers.

This section presents the classical compact mixed integer formulation of the Capacitated Vehicle Routing Problem (CVRP), the most studied variant of VRP. The CVRP is an extension of the well-known Traveling Salesman Problem (TSP), which requires visiting a given set of points exactly once to determine a Hamiltonian circuit at minimal cost (Toth & Vigo, 2002). The customer demands are deterministic, known in advance, and indivisible. The vehicles are located in one central warehouse, and only capacity restrictions apply to vehicles.

The presented model is well-suited for solving simple VRP variants using mathematical programming-based techniques. The model is presented by Toth & Vigo (2014) on a directed graph in traditional notation as follows:

Let $G(N, A)$ be a graph having a node set N and an arc set A. For i and $j \in N$, we indicate the arc from i to j as $(i, j) \in A$. We use the term "network" to mean a graph with additional data on its nodes and arcs. $N = \{0,1,2, \ldots, n+1\}$ is the set of nodes, in which 0 and $n+1$ represent the depot. Additionally, each node $i \in N$ has a load q_i. The loads of the depots are equal to 0 ($q_0 = q_{n+1} = 0$). An unlimited fleet of different or identical vehicles with usage cost cv^k and capacity C_k is ready to meet customer demands. Each arc $(i, j) \in A$ has a travel cost/distance c_{ij} related to travel distance. The notation required for formulating the Mixed Integer Programming (MIP) model of the CVRP is introduced below.

Indices:

i, j	indices for nodes
k	index for vehicles

Sets:

N	set of all nodes, $i, j \in \{0,1,2, \ldots, n+1\}$
N'	set of customers, $i, j \in \{1,2, \ldots, n\}$
V	set of all vehicles, $k \in \{1,2, \ldots, m\}$

Parameters:

C^k	the capacity of vehicle k
cv^k	cost of using vehicle k
c_{ij}	traveling cost/distance between nodes i and j

q_i the demand of customer i

Decision variables:

x_{ij}^k 1, if arc (i, j) is traversed by vehicle k, 0 otherwise

y^k 1, if vehicle k is used and 0 otherwise

Objective function:

$$Minimize \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij}\, x_{ij}^k + \sum_{k \in V} cv^k y^k \tag{2.1}$$

Subject to:

$$\sum_{k \in V} \sum_{j \in N} x_{ij}^k = 1, \qquad \forall i \in N' \tag{2.2}$$

$$\sum_{j \in N} x_{0j}^k = y^k, \qquad \forall k \in V \tag{2.3}$$

$$\sum_{i \in N} x_{i(n+1)}^k = y^k, \qquad \forall k \in V \tag{2.4}$$

$$\sum_{i \in N} x_{ij}^k - \sum_{i \in N} x_{ji}^k = 0, \qquad \forall j \in N', \forall k \in V \tag{2.5}$$

$$\sum_{i \in N} \sum_{j \in N} x_{ij}^k\, q_i \le C^k, \qquad \forall k \in V \tag{2.6}$$

$$x_{ij}^k \in \{0,1\}, \qquad \forall k \in V, \forall i, j \in N \tag{2.7}$$

$$y^k \in \{0,1\}, \qquad \forall k \in V \tag{2.8}$$

The objective function (2.1) minimizes the total cost/distance traveled and the number of vehicles used. Constraint (2.2) requires that each customer be visited once by one vehicle. Constraint (2.3) guarantees that a vehicle is active only when it departs the depot. Constraint (2.4) ensures that each active vehicle returns to the depot. Constraint (2.5) is a flow conservation constraint; when a vehicle visits a customer, it must go to another customer from that customer. Constraint (2.6) ensures that the loading capacity of vehicles is not exceeded. Finally, the constraints (2.7)-(2.8) define the decision variables.

2.3 The Variants of the Vehicle Routing Problem

Since first designed by Dantzig & Ramser (1959), taxonomies and surveys have been published to classify the VRPs, review the literature, and identify the recent trends. A review of algorithms for solving the VRP has been published by Laporte & Nobert (1987), Toth & Vigo (2002), and Poggi & Uchoa (2014). In recent years, prominent taxonomies in this context belong to Golden, Raghavan, & Wasil (2008), Laporte (2009), Toth & Vigo (2014), and Braekers, Ramaekers, & Van Nieuwenhuyse (2016).

Some characteristic features of the VRP are summarized in Braekers et al.'s (2016) paper, such as scenario, customers' physical locations, and homogeneity of vehicles. Figure 2.1 introduces the most important variants of VRP presented by Režnar, Martinovič, Slaninová, Grakova, & Vondrák (2017). It is mainly intended to provide a classification perspective that can help identify specific features of the VRP.

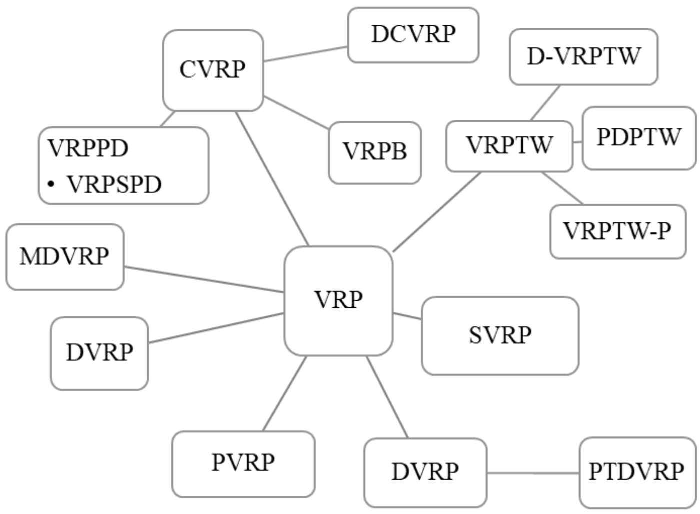


Figure 2.1 The variants of the VRP (Režnar et al. 2017)

The Distance-Constrained Vehicle Routing Problem (DVRP) minimizes the total distance traveled or the number of vehicles on all routes while limiting the route distances. Several algorithms were proposed for solving the DVRP (Laporte (2009); Desrochers & Nobert (1984); Nagarajan & Ravi (2012). The Distance-Constrained Capacitated Vehicle Routing Problem (DCVRP), an extension of the DVRP, considers vehicle capacity and maximum distance constraints (Kek, Cheu, & Meng, 2008).

The Vehicle Routing Problem with Backhauls (VRPB) is an extension of the CVRP, where the pickups should begin once all deliveries on the route have been completed. Pickup customers are also denoted as backhaul customers and delivery customers as linehaul customers. The problem was surveyed by Gribkovskaia & Laporte (2008).

The Vehicle Routing Problem with Time Windows (VRPTW) is an extension of the CVRP, where service for each customer must begin within an associated time window, and the vehicle must remain at the customer's location during the service. The objective is to serve customers within predefined time windows while minimizing vehicle travel distances. If the vehicle arrives before the customer's service start time, the vehicle will wait (e.g., Cordeau, 2000). The authors studied exact methods (e.g., Kallehauge, Larsen & Madsen, 2006; Kolen, Rinnooy Kan, & Trienekens, 1987) and metaheuristics (e.g., Pullen & Webb, 1967; Tan, Chew, & Lee, 2006) for solving the VRPTW. Kok, Meyer, Kopfer, & Schutten (2010) developed a solution method for the VRPTW with time-dependent travel times and driving hours regulations. Ando & Taniguchi (2006) developed a model of the Vehicle Routing Problem with Time Windows-Probabilistic (VRPTW-P) that takes into account the uncertainty in transit time.

The Dynamic Vehicle Routing Problem with Time Windows (D-VRPTW) is an extension of the VRP that considers the problem's dynamic nature. This feature requires vehicle routes to be constantly updated as new customer requests come into the system and to be included in a schedule that changes throughout the business day (Necula, Breaban & Raschip, 2017).

The Multi-Depot Vehicle Routing Problem (MDVRP) is another variant in which the vehicles start and end their routes in different depots. In the MDVRP, vehicle groups can be assigned to different depots due to limited depot capacity. For example, the MDVRP was studied in the literature by Renaud, Laporte, & Boctor (1996) using a tabu search heuristic. Recently, Contardo & Martinelli (2014) presented an exact algorithm for the MDVRP. In addition to presenting a literature review on the MDVRP, Vidal, Crainic, Gendreau, & Prins (2013) provided an efficient heuristic for its periodic variants.

The Periodic Vehicle Routing Problem (PVRP) is also a variant of the VRP. Customers can request repeated delivery of goods. There are two levels of planning in the PVRP: At the first level, an appropriate visit pattern is selected from a specific proper sequence for each customer. At the second level, a VRP is solved daily in line with the subset of customers to be visited, and their demands are determined in the first level. The problem is finding feasible routes for customers at the required times over a planning horizon with minimum cost. In practice, problems like bakery product distribution or waste collection can be solved as the PVRP. When the proposed solution methods for PVRP are examined, it is seen that heuristic (e.g., Mourgaya & Vanderbeck, 2006) and exact (e.g., Cacchiani, Hemmelmayr, & Tricoire, 2014) solution approaches are suggested.

The Vehicle Routing Problem with Pickup and Delivery (VRPPD) is one of the important VRP variants. Along the supply chain, some customers expect materials to be received, while others expect delivery, but no customer expects to be received and delivered simultaneously. Parragh, Doerner, & Hartl (2007) and Parragh, Doerner, & Hartl (2008) introduced excellent surveys for this variant. The Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD) is an extension of the VRPPD. This version allows for solving problems where customers can expect both pickup and delivery at the same time.

The Stochastic Vehicle Routing Problem (SVRP) with soft time windows under travel and service time uncertainties was studied by Zhang, Lam, & Chen (2013). They offered a stochastic time-independent scheduling model to minimize travel costs while ensuring the probability of arriving at each customer location on time.

2.4 The Pickup and Delivery Problems

The pickup and delivery problem (PDP) can be defined as routing problems where goods or people have to be collected from the point of origin and delivered to a destination. These problems can be defined on a graph illustrated in Figure 2.2, where the nodes represent the points of origin (pickup) or destination (delivery) of the different goods to be transported and the warehouse from which the vehicles leave.

In PDPs, a single vehicle or a group of vehicles can be deployed to meet the demand of a group of customers. However, every customer request requires a single vehicle to be assigned to load some goods from one location and deliver them to another location. Therefore, the objective of the classical PDP may be to minimize the number of vehicles used, the sum of the travel distances (costs) of vehicles, or both with specified weights.

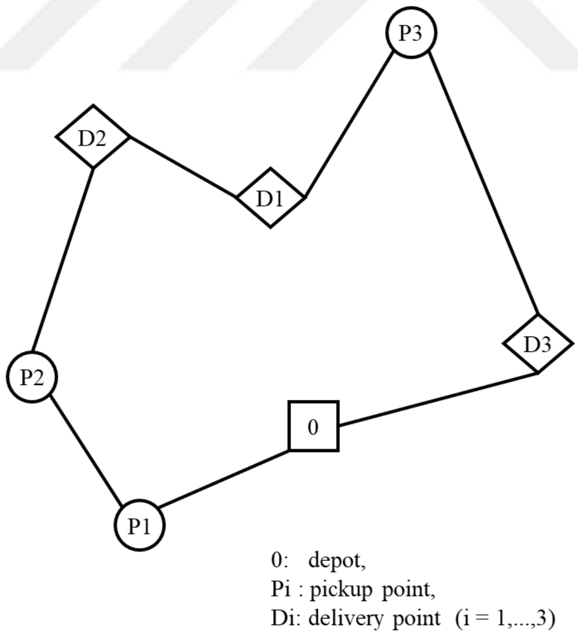


Figure 2.2 Pickup and delivery problems

The PDP enables the optimization of various logistics applications in a real-world environment, such as forward or reverse logistics, movement of freight carts, collection and distribution of empty cans and bottles, bicycle repositioning activities, transportation of student shuttles, and airline cargo/passenger transportation. PDPs are classified in the literature according to their configurations, receiving and delivery activities at nodes, and types of goods (Ho & Szeto, 2016).

Berbeglia, Cordeau, Gribkovskaia, & Laporte (2007) and Battarra, Cordeau, & Iori (2014) presented a classification of the PDP consisting of three main categories, Many-to-many (M-M) problems are the first category, where each good can have multiple pickup points and delivery points, and any point can be the starting and destination point of a set of goods. In the literature, these problems mostly refer to single-vehicle scenarios and are limited. This is probably because M-M problems are not often encountered in practice.

In the one-to-many-to-one (1-M-1) PDP category, some goods are transported from one depot to many nodes, while others are collected from many nodes and delivered to the depot. 1-M-1 problems arise, for example, in reverse logistics, where full bottles must be brought to customers, and empty bottles must be returned from customers to the warehouse. One-to-one (1-1) PDPs are the final category, where each commodity has a single starting node and a single ending node. A common example of applying 1-1 PDPs is found in local area courier services' routing operations. Like other VRPs, PDPs may also be classified according to the decision framework being considered and information availability.

Since the PDP was first introduced by Desrochers, Lenstra, Savelsbergh, & Soumis (1988), several studies have been published focusing on solving practical examples of the PDP through a variety of techniques, such as branching methods, decomposition methods (column building, row generation), dynamic programming, heuristics, and meta-heuristics (e.g., Ruland & Rodin, 1997, Dumas, Desrosiers, & Soumis, 1991, Cordeau, 2006, Ropke & Pisinger, 2006, Bent & van Hentenryck, 2006). This thesis examines the one-to-one PDP with visiting time windows and loading constraints in detail and presents an Evolutionary Algorithm (EA)-based solution approach in

Chapter 6. The next section gives the mathematical formulation of the one-to-one PDP with Time Windows to understand the problem's constraints.

2.4.1 The Mathematical Formulation of the One-to-One PDP with Time Windows

The one-to-one PDP with Time Windows can be formulated as the following MIP model (MIP-PDPTW); Let *r* denote the number of requests to satisfy. The PDP can be defined on a directed graph $G = (N, A)$ with node set $N = \{0, \ldots, n+1\}$ and arc set A. Nodes 0 and $n+1$ represent the origin and destination depots (which may have the same location), while subsets $P = \{\ldots\}$ and $D = \{\ldots\}$ represent pickup and delivery nodes, respectively. With each request, $r \in R$, a pickup node $P(r)$, and a delivery node $D(r)$ are associated. Each node $i \in N$ is associated with a load q_i and a non-negative service duration d_i, satisfying $d_0 = d_{n+1} = 0$, $q_0 = q_{n+1} = 0$. An unlimited fleet of identical vehicles with capacity C^k is available to serve the requests. Each arc $(i, j) \in A$ is associated with a routing cost c_{ij} and a travel time/distance t_{ij}. A time window $[e_i, l_i]$ is also associated with each node $i \in P \cup D$, where e_i and l_i represent the earliest and latest time, respectively, at which service may start at node i. The depot nodes may also have time windows $[e_0, l_0]$ and $[e_{n+1}, l_{n+1}]$, representing the earliest and latest times, respectively, at which the vehicles may leave and return to the depot. We assume that the triangle inequality holds for routing costs and travel times. The notation for the MIP-PDPTW is given as follows:

Indices:

i, j	indices for nodes
k	index for vehicles
r	index for requests

Sets:

N	set of all nodes (customers and depots), $\{0, \dots, n+1\}$
V	set of all vehicles, $\{1,2, \dots, m\}$
P	set of all pickup nodes
D	set of all delivery nodes

Parameters:

C^k	capacity of vehicle $k \in V$
c_{ij}	distance/cost between nodes i and $j, i \in N, j \in N$
t_{ij}	travel time/distance between nodes i and $j, i \in N, j \in N$
d_i	service duration of customer $i \in N$
q_i	load amount (demand) of customer $i \in N$
M_{ijk}	big number for the beginning time of arc $i, j \in N$ and vehicle $k \in V$
W_{ijk}	big number for the load of arc $i, j \in N$ and vehicle $k \in V$

Decision variables:

x_{ijk}	if $arc\ (i,j)$ is traversed by vehicle $k \in V$,0 otherwise
B_i^k	beginning time of service of vehicle $k \in V$ at node $i \in N$
Q_i^k	load of vehicle $k \in V$ when leaving node $i \in N$

With the introduced notation, the MIP-PDPTW model can be given as follows:

$$Min \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij}\, x_{ijk} \qquad (2.9)$$

Subject to:

$$\sum_{k\in V}\sum_{j\in N} x_{ijk} = 1, \qquad \forall i \in N \ (i \neq n+1, j \neq 0) \tag{2.10}$$

$$\sum_{j\in N} x_{0jk} = 1, \qquad \forall k \in V \tag{2.11}$$

$$\sum_{i\in N} x_{i(n+1)k} = 1, \qquad \forall k \in V \tag{2.12}$$

$$\sum_{i\in N} x_{ijk} - \sum_{i\in N} x_{jik} = 0, \qquad \forall j \in N, \forall k \in V \tag{2.13}$$

$$\sum_{j\in N} x_{P(r)jk} - \sum_{j\in N} x_{D(r)jk} = 0, \qquad \forall r \in R, \forall k \in V \tag{2.14}$$

$$x_{ijk} => 1 \quad B_j^k \geq B_i^k + d_i + t_{ij}, \qquad \forall i,j \in N, \forall k \in V \tag{2.15}$$

$$B_{P(r)}^k \leq B_{D(r)}^k, \qquad \forall r \in R, \ \forall k \in V \tag{2.16}$$

$$e_i \leq B_i^k \leq l_i, \qquad \forall i \in N, \forall k \in V \tag{2.17}$$

$$x_{ijk} => 1 \quad Q_j^k = Q_i^k + q_i, \qquad \forall i,j \in N, \forall k \in V \tag{2.18}$$

$$max(0, q_i) \leq Q_i^k \leq min(C^k + q_i), \qquad \forall i \in N, \forall k \in V \tag{2.19}$$

$$x_{ijk} \in \{0,1\}, \qquad \forall i,j \in N, \forall k \in V \tag{2.20}$$

The objective function (2.9) minimizes the total routing cost/distance. Constraints (2.10) state that every node has to be served precisely once. Constraints (2.11) and (2.12) guarantee that every vehicle starts at the depot and returns to the depot at the end of its route. Note that this does not mean that every vehicle has to be used. A vehicle may use the arc $(0, n+1)$, i.e. it may not leave the depot. The flow conservation principle is ensured by constraints (2.13). Constraints (2.14) guarantee that the same vehicle must serve both the origin and destination of a request. Time variables are used to eliminate sub-tours in constraints (2.15), given that $(t_{ij} + d_i) > 0$ for all $(i, j) \in A$ and delivery can only occur after pickup as enforced by constraints (2.16). Constraints (2.17) ensure that the time window restrictions are not violated. Constraints (2.18) and (2.19) guarantee that the capacity of a vehicle is not exceeded throughout its tour. It should be noted that this formulation requires the introduction of additional decision variables, Q_i^k, corresponding to the total load of vehicle k at node i.

Non-linear constraints, given in constraints (2.21) and (2.22), can be linearized using a big M formulation with introducing constants M_{ijk} and W_{ijk} as follows (Cordeau, 2006):

$$B_j^k \geq B_i^k + d_i + t_{ij} - M_{ijk}(1 - x_{ijk}), \qquad \forall i, j \in N \quad \forall k \in V \tag{2.21}$$

$$Q_j^k \geq Q_i^k + q_i - W_{ijk}(1 - x_{ijk}), \qquad \forall i, j \in N \quad \forall k \in V \tag{2.22}$$

The validity of these constraints is ensured by setting $M_{ijk} \geq max\{0,\ l_i + d_i + t_{ij} - e_j\}$ and $W_{ijk} \geq \min\{Q_{ik}, Q_{ik} + q_i\}$.

2.5 The Container Loading Problems

Cutting and packing (C&P) problems, which can be found in a multitude of situations such as cutting large rolls into small rolls, packing boxes into a vehicle, or, less intuitively, assigning personal shifts to an organization schedule, are combinatorial optimization problems that arise in many real-world contexts. C&P problems, which have gained increasing research interest in recent years, pose challenging academic problems, although they are strongly NP-hard (Pisinger, 2002). Very generally, the types of problems where subsets of small items are assigned to one or more large objects (such as containers) so that they are entirely placed inside the large objects and prevented from overlapping can be defined as C&P problems (Wäscher, Haußner, & Schumann, 2007). C&P problems can be classified according to dimensionality, optimization purpose, assortment of items and containers, and heterogeneity of items. The three-dimensional C&P problems are called Container Loading Problems (CLPs).

CLPs are geometric assignment problems that can be interpreted as packing small rectangular three-dimensional objects called boxes into large three-dimensional rectangular objects called containers (bins). The standard procedure for solving such assignment problems is determining the loading pattern (Bortfeldt & Wäscher, 2013). It should be known that a large object may be an actual container, but it can also be a truck or pallet loading area to a certain height.

In the CLP, all the boxes must be packed into a single container with an infinite length. Thus, the problem is finding a feasible solution to minimize the container's length. The three-dimensional bin packing problem calls for a solution where all the boxes are packed into the containers, but in contrast to the CLP, all the containers have finite dimensions, and the objective is to find a solution using the smallest possible number of bins (Martello, Pisinger, & Vigo, 2000).

On the other hand, two types of objective functions are usually given to optimize C&P problems. The first is the minimization of the input. This type includes, for example, minimizing the length or height of the container used or the number of containers used. In this case, a set of items should be assigned to a container group, and all items should be placed. The second type of objective function consists of maximizing the output, that is, maximizing the packed value or volume. For example, as with the knapsack problems (KPs), not all items can be accommodated by the existing containers, and the assigned items must be selected.

In this thesis, we consider a different version of the CLP which does not maximize output or minimize input but tries to determine whether a feasible packing plan exists that accommodates all given items in a given container.

In general, small objects can have any regular or irregular shape. However, with few exceptions, publications in the container loading area deal with small rectangular items. The objective is to provide geometric feasibility conditions such that all items are completely contained in the container and do not overlap.

In the process of loading the items, in accordance with the available literature, it will be assumed that only orthogonal layouts are allowed; that is, the surfaces of the items must be aligned parallel to the floor and walls of the container. As this thesis focuses on loading three-dimensional items, only three-dimensional packing problems are covered in the remainder of this subsection.

Optimizing the CLP has become crucial to running the supply chain efficiently. Failure to do so will result in poor performance, unnecessary costs, and unsatisfactory customer service. Therefore, it is unsurprising that CLPs are frequently addressed in the operations research literature. Unfortunately, few exact solution methods exist, and only a few models have been proposed because of their difficulty. Chen, Lee, & Shen (1995) introduced a mixed-integer linear programming (MILP) formulation, which was later extended by Fasano (1999) and Padberg (2000). Fekete, Schepers, & Van der Veen (2007) presented an exact two-level tree search algorithm.

Due to the difficulty of solving the CLP, numerous heuristic approaches have been proposed in the literature, including construction and improvement heuristics, meta-heuristics, and heuristic tree search approaches (e.g., Gehring, 1997; George & Robinson, 1980; Ren, Tian, & Sawaragi, 2011). These approaches are often based on the geometrical structures of loading plans, such as wall construction, stack construction, layer construction, and block construction (Pisinger, 2002).

Vertical layers of items are filled along the length axis within the container when employing wall-building approaches. Such approaches are presented, for example, by George & Robinson (1980), Gehring, Menschner, & Meyer (1990), Bortfeldt & Gehring (2001), and Pisinger (2002).

Stack-building algorithms organize stacks of objects on the floor of shipping containers. When using horizontal layer building methods, the container is filled from the bottom up with horizontal layers. The approach is used by Bischoff (1995) and Terno, Scheithauer, Sommerweiß, & Riehme (2000).

Block-building approaches entail creating blocks containing mostly identical items with the same spatial orientation. These blocks are then placed within a container. In recent times, block-building approaches have been introduced, wherein the items forming a block need not be of the same type or possess the same spatial orientation (Fanslau & Bortfeldt, 2010; Araya & Riff, 2014). Examples of the construction heuristics are illustrated in Figure 2.3.

To achieve better results, construction heuristics are frequently combined with more intricate search techniques. For instance, Gehring & Bortfeldt (1997) elaborate on a solution method comprising a stack-building algorithm for creating stacks of items and a GA, assigned to resolve the ensuing issue of covering the container base with the stacks of items.

In other GAs, applied by Bortfeldt & Gehring (2001), a wall-building procedure forms vertical layers, and GA operators transfer layers from parent to offspring solutions. Heuristic approaches for solving the CLP Heuristic tree search approaches are considered by Terno et al. (2000), Eley (2002), Hifi (2002), Pisinger (2002), and Fanslau & Bortfeldt (2010). Zhao, Bennell, Bektaş, and Dowsland (2016) provide a recent review and comparison of solution approaches for the CLP.

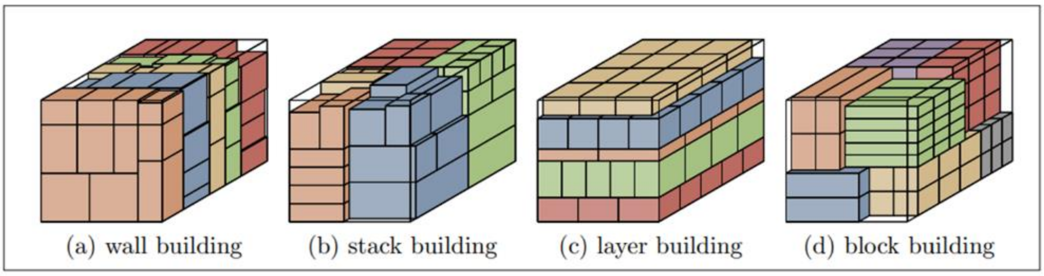


Figure 2.3 The construction heuristics for the CLP (Koch, 2018)

2.5.1 *Container Loading Constraints*

As stated initially, this thesis focuses on routing and loading problems. This section, where the loading problem is discussed, is aimed to create a perspective for the loading-induced variations of the problems solved in the thesis. While the most important factor in the differentiation of loading problems is the objective function, the other is the constraints, many of which are discussed in the literature. According to Bortfeldt & Wäscher (2013), these constraints refer to containers (container-related constraints), items (item-related constraints), sets of goods (cargo-related constraints, the relationship between containers and items), or final load (load-related constraints).

Constraints may arise during container loading which can be categorized as either hard or soft constraints. Hard constraints are non-negotiable and must always be adhered to; a loading pattern that breaches a hard constraint is not feasible. It is crucial to acknowledge and adhere to these constraints to ensure safe and efficient container loading. Soft constraints, on the other hand, are guidelines that should ideally be followed, although violations within certain limits are acceptable.

2.5.1.1 Classical Loading Constraints (CLC)

The primary constraints of the CLP are these constraints, which consist of two rules; All loaded small items (boxes) are completely contained within the large object (container), and small items (boxes) do not overlap (Wäscher et al., 2007).

2.5.1.2 Weight Limits (WL)

When loading a container or vehicle compartment, the overall volume occupied by small items to be loaded is more restrictive. However, when heavy items need to be loaded, weight limits can become more restrictive than the three-dimensional constraints set by the dimensions of the containers (Figure 2.4).

Figure 2.4 Weight limits

Within container loading algorithms, weight limits can be modeled similarly to the knapsack problem constraints, where the sum of the weights of the items loaded must be less than or equal to the weight limit imposed by the container. Thus, the suitability of solutions can be checked simply and quickly.

2.5.1.3 Weight Distribution Constraints (WD)

Weight distribution constraints, also called load balance constraints, require the load's center of gravity to be close to the container floor and spread evenly over the floor surface. The most practical way of achieving this is that heavier items must be stacked near the bottom of the container, while lighter items should be packed at a higher level. Balanced loads reduce the risk of cargo slipping during the movement of the container while saving additional energy for stability, as in air cargo. Davies and Bischoff (1999), Eley (2002), and Baldi, Perboli, and Tadei (2012) tried to balance the container by positioning the total center of gravity of the loaded objects as close as possible to the geometric midpoint of the container base. In Baldi et al. (2012), where the three-dimensional knapsack problem is addressed, the authors determined a two-dimensional convex area bottom of the container to move the center of gravity to the desired position. They then positioned the selected items such that the center of gravity was inside the area.

In this thesis, the container is filled from the bottom to the top, with the items to be almost balanced along the vertical y-axis. It is challenging to obtain an ideal balance along the z-axis in which the center of gravity is as close as possible to the bottom of the container while achieving high-volume utilization. To balance the container along the x-axis, we try to obtain a feasible loading solution in which the relative difference in the moment of force along the x-axis does not exceed the given allowable level. The y-axis is handled similarly. According to actual shipping companies, the given allowable levels that can be accepted are 20 and 10 % for the relative differences in the moment of force along the x-axis and y-axis. However, the ideal level is zero. Figure 2.5 shows an example of WD at the level whose sizes are 20% of the container dimensions.

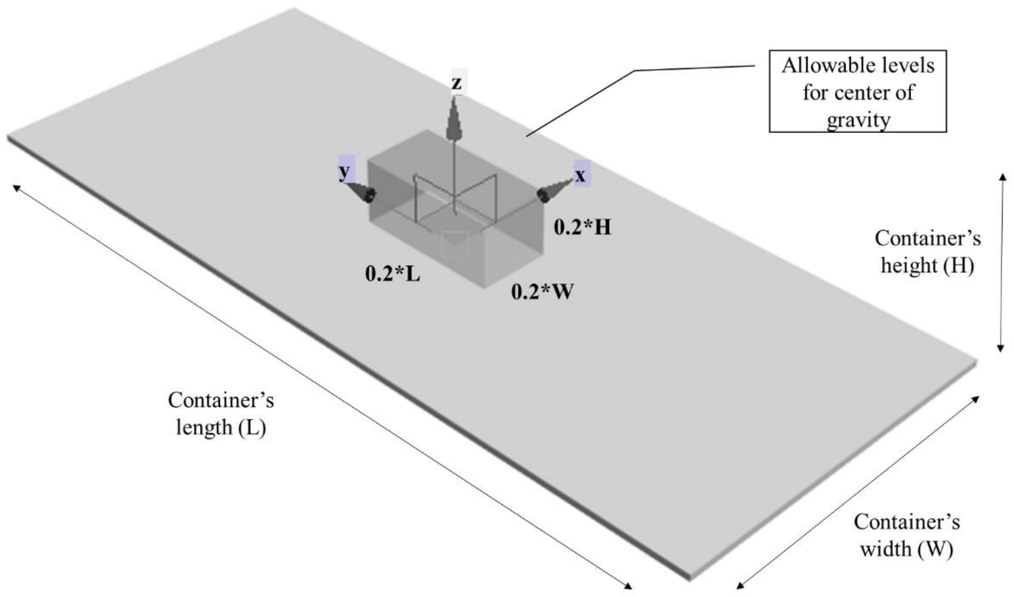


Figure 2.5 Weight distribution constraint

2.5.1.4 Loading Priorities (LP)

For reasons such as delivery dates or shelf life, it may be more desirable to load some items than others, i.e., there are loading priorities for items. When the available container capacity is insufficient to accommodate all these small items, the problem of deciding which should be loaded and which should be left behind arises.

In such problems with output maximization, loading some items is mandatory, while loading others is optional in line with soft constraints. In addition, other items can be further differentiated according to different priority classes. That way, if a higher-priority item requires dropping, it can reflect a situation where no lower-priority items will be shipped.

2.5.1.5 Orientation Constraints (OR)

Each dimension of a box can act as the height, and the dimension chosen as the height is defined as the vertical direction of the box. In this way, each box can be placed inside the container in six different directions (Figure 2.6). However, in some cases, the specified dimensions of a box may not be allowed to be height, thus limiting the allowable number of orientations of the box in both vertical and horizontal directions.

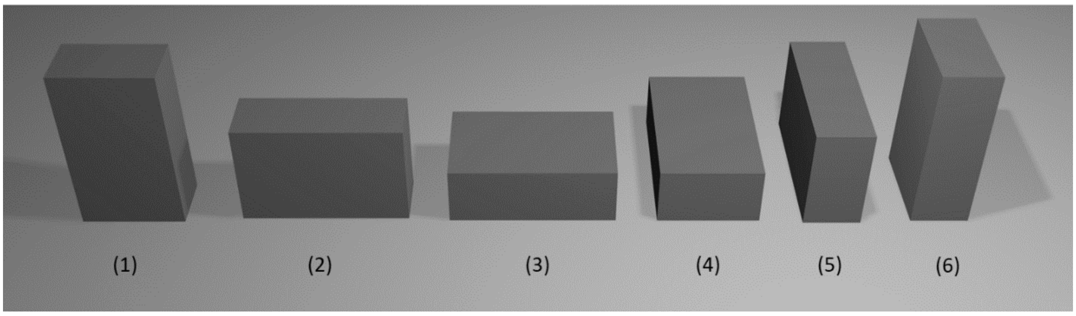


Figure 2.6 Six orientations of a box

As a result, orientation constraints frequently restrict a box's vertical orientation to one or two dimensions, also known as the "This way up!" constraint. These limitations aim to prevent damage to items and maintain load stability, resulting in the constraints being referred to as hard constraints in literature.

2.5.1.6 Stacking (Fragility) Constraints (FR)

Stacking constraints, also called load-bearing constraints, restrict how boxes can be stacked depending on their load-bearing strength. The load-bearing strength of the support box cannot be solely determined by the maximum weight applied per unit area. The strength of the side walls also plays a crucial role, making it a more appropriate measure of load-bearing strength to consider the weight that can cause the edges to be crushed. For convenience, in some cases, the boxes are assumed to be fragile and non-fragile. Fragile boxes can only be stacked on top of other fragile boxes, while both fragile and non-fragile boxes can be stacked on top of a sturdy box.

2.5.1.7 Complete-Shipment Constraints (CS)

When trying to maximize the output, it's important to arrange the load in the best possible way. However, if the container space is limited, some items might have to be left behind. In such cases, if certain subsets of items are functionally or administratively linked, then all items in that subset must be loaded if at least one item is loaded. Similarly, if one item in a subset fails to load, then none of the other items in that subset will be loaded. These limitations can arise when all the shipping options are taken, such as when a piece of furniture is packed in separate parts and must be assembled at the customer's home.

2.5.1.8 Allocation Constraints (AL)

Allocation constraints occur only in the case of loading multiple containers or vehicles. In such problems, it may be necessary that the elements of a given subset all fall into the same container; for example, all products shipped to the same destination go to a customer who wishes to receive it as one consignment, not piecemeal.

2.5.1.9 Positioning Constraints (LIFO)

Positioning constraints restrict the location of items within the container or their location relative to each other. Due to the limited maneuverability of the forklift or various safety measures, the entrance door represents the most challenging area to load and should not be used for large, heavy items.

Multiple drop conditions, which combine absolute and relative positioning constraints, are defined by subsets of products that are delivered to different customers. The literature on integrated container loading and vehicle routing problems often includes a specific multiple drop condition to facilitate unloading operations. This requirement necessitates the availability of requested items at each stop without any rearrangements while meeting it through the loading of items into the truck compartment following the Last In First Out (LIFO) policy.

2.5.1.10 Stability Constraints (ST)

Load stability is a crucial matter discussed in literature besides efficient container utilization. Unstable loads may harm and injure employees during transportation, loading, and unloading tasks, as well as damage the cargo.

Load stability comprises vertical and horizontal stability, where vertical stability prevents items from dropping onto the container floor or other items. The document pertains to situations in which the container remains stationary and specifies the load's ability to resist gravity's force.

To address vertical stability concerns, it is customary to require that the underside of a container be fully or partially supported by a level area on the container floor or the top surfaces of adjacent containers. Furthermore, support for the entire base area may be necessary. Horizontal stability guarantees items do not undergo significant movement due to inertia while being transported in the container.

2.6 The Vehicle Routing Problem with Loading Constraints

In numerous industrial areas, competition during the procurement process has compelled firms to create computer systems that integrate logistics operations. Correspondingly, researchers have directed their attention towards optimisation problems originating from real-world logistics operations, specifically finding the most appropriate vehicle routes and loading patterns. This incentive can aid in introducing more effective solution techniques for distribution logistics globally (Hokama, Miyazawa, & Xavier, 2016).

In the context of the CVRP, most approaches in the literature assume that the vehicle's capacity is a simple one-dimensional measure, which is the vehicle's maximum weight capacity. However, it is evident that this approach is not sufficient in real-world conditions. For example, in some cases, it may be more realistic to distribute loads with loading constraints that take into account shape, not just weight.

In this way, while it is possible to meet the weight capacity, creating a suitable layout for all loads may be impossible. In this sense, and to include loading specificities, some studies, including this thesis, have been presented in the last decade that integrate the multi-dimensional loading problem with the CVRP.

This emerging problem is defined as the load-constrained CVRP in the literature, and it is common to handle loads in two or three dimensions (2L-CVRP or 3L-CVRP) (Pinto, 2016). This problem is NP-hard because it integrates two NP-hard combinatorial problems Methods applied to this problem must take into account both the constraints of the routing problem and the constraints of loading two- or three-dimensional elements in the vehicle. For the 3L-CVRP, the loading component can be treated as a CLP. For this reason, the compartments of the vehicles related to freight transportation should be considered by identifying them with the container. The packing scheme of the vehicle changes each time a load is received or delivered; This means that loading restrictions must be monitored throughout the journey, not just at departures.

2.6.1 The 2L-CVRP

In the 2L-CVRP, the customers' demands and the vehicles' dimensions are expressed in two dimensions (e.g., Iori, Salazar-González, & Vigo, 2007; Fuellerer, Doerner, Hartl, & Iori, 2009). The width and length are usually taken into account, while the height is not. In real-life distribution logistics, this problem arises when items cannot be stacked because of their weight, fragility, or large size. Examples are distributing large kitchen appliances such as refrigerators, large mechanical components, or fragile glass-based items.

Each customer's demand consists of a limited number of two-dimensional rectangular pieces and their weight information. The 2L-CVRP consists of finding a set of routes that start and end at the warehouse, minimizing the total cost of travel, and meeting problem-specific constraints (Pinto, 2016):

- The number of routes in the solution cannot exceed the total number of vehicles,
- All customers should be served in a single visit without interrupting their demands,
- The loaded items must be fully within the vehicle's bearing surface, not overlapping, and the edges of the parts must be parallel to the surface edges.

Some additional constraints can be considered in the problem, such as weight capacity constraints and pickup and delivery scenarios.

For example, sequential constraints impose that the unloading of a part at the customer must be accomplished through a free passage towards the door (Pinto, Alves, & Valério de Carvalho, 2018). This means that on parts to be delivered to other customers, no lateral movement is allowed to block the passage of that customer's shipments. During the unloading process, the part must protect its edges parallel to the surface edges.

Figure 2.7 shows the case where $I_{i,j}$ represents item j of customer i. For customers 4, 1, and 3, the items can be unloaded in one straight motion for each item. However, according to the loading plan on the left of the figure, item $I_{3,1}$ prevents the passage of item $I_{2,2}$ while the items belonging to customer 2 are being unloaded. Since the items belonging to the same customer can be emptied in a straight motion without rearranging the items to be delivered, the loading plan on the right side of the figure can be considered feasible.

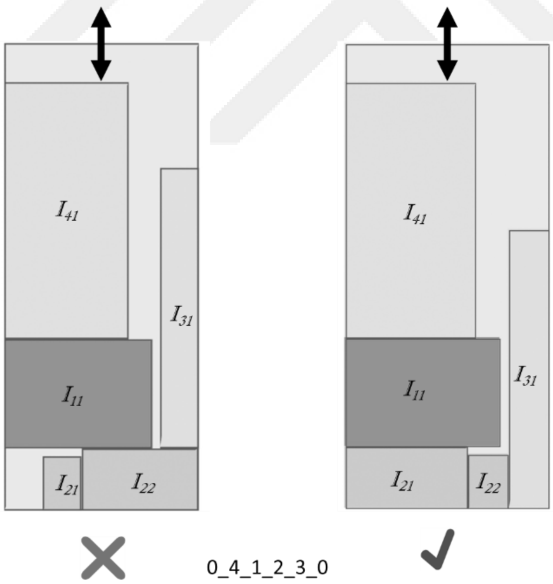


Figure 2.7 Loading plans for the 2L-CVRP

2.6.2 The 3L-CVRP

The 3L-CVRP is the generalization of the 2L-CVRP with the three-dimensional axis introduced by Gendreau et al. (2006). The mathematical formulation of both problems for routing is the same (Pollaris, Braekers, Caris, Janssens, & Limbourg, 2015). However, the vehicles should be viewed as a three-dimensional rectangular loading container, and each customer's request consists of three-dimensional rectangular items (boxes). Therefore, the 3L-CVRP requires determining a set of optimal routes that start and end at the depot while meeting routing constraints and minimizing the total cost of travel. Loading constraints include those that must be met in the classic CLP. Gendreau et al. (2006) introduced several loading constraints considering vertical stability, fragility, item orientation, and loading order. This constraint set was adopted by most researchers who later addressed this problem.

The most prominent constraint among these constraints is the request sequence constraint. This constraint ensures that unloading an item from a particular customer does not require moving other items for delivery to other customers. In the 3L-CVRP, the free passage between the vehicle's parts and rear is insufficient. Also, no items from other customers should be on top of the unloaded.

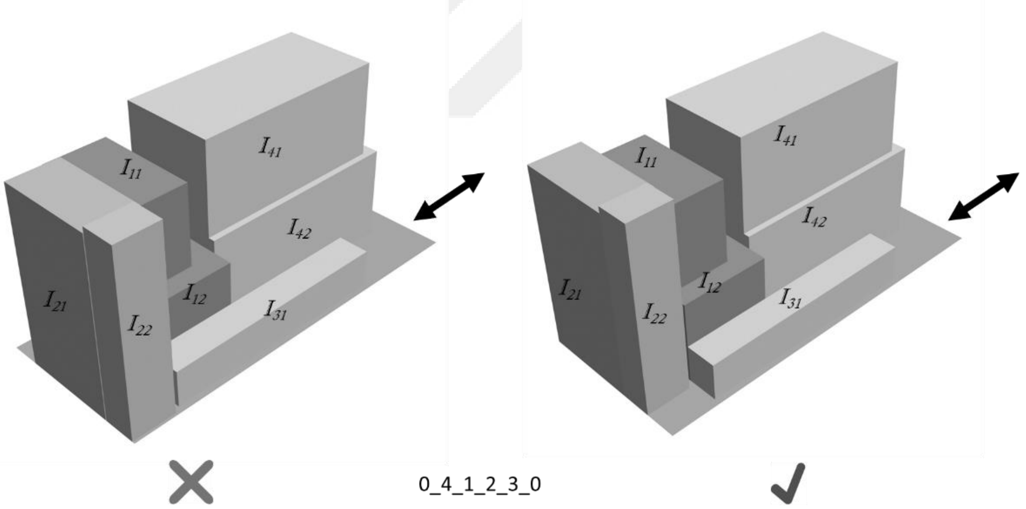


Figure 2.8 Loading plans for the 3L-CVRP

Figure 2.8 shows two examples assuming route $0-4-1-2-3-0$. The loading plan on the left of the figure is not feasible as $I_{3,1}$ blocks $I_{2,1}$. The loading plan on the right side of the figure can be considered a viable solution.

2.7 Chapter Summary

This chapter explains the VRP, PDP, and CLP, which are the main components of the 3L-CVRP. In addition, the loading constraints considered for the CLP in the literature have been categorized. It is emphasized that loading constraints are essential in getting close to VRPs in real-world applications.

Then, brief information is given about the VRP with loading constraints. Moreover, the importance of the thesis work is better understood by giving the importance of routing and loading integration.

The following chapter gives a detailed literature review of the problem. The mathematical formulation of the 3L-CVRP and the exact and heuristic solution methods to solve the problem are explained.

CHAPTER THREE
LITERATURE REVIEW ON THE 3L-CVRP

3.1 Introduction

This thesis focuses on the 3L-CVRP, and a review of the relevant literature is provided in this chapter.

3.2 Literature Review of the 3L-CVRP

An overview of the literature on the 3L-CVRP and the solution methods for each study, the problem characteristics, and the constraints considered are summarized in Table 3.1. The loading-related constraints are already discussed in Chapter 2.

Recently, there has been a growing interest in the 3L-CVRP owing to the advantages of integrating routing and loading problems. Gendreau et al. (2006) is the first study that addressed the 3L-CVRP. The authors proposed a two-step tabu search algorithm that minimizes the total travel cost by considering all loading constraints, such as batch restriction, LIFO restriction, weight restriction, orientation restriction, and support restriction. After a while, Moura & Oliveira (2009) discussed the 3L-CVRP with service time windows. Their article developed a MIP model that minimizes the total distance traveled and the number of vehicles required for loading. They also proposed two solution approaches based on the greedy randomized adaptive search procedure (GRASP) algorithm. In the first sequential approach, the planning of the routes and loading of vehicles are done simultaneously. In the second hierarchical approach, the CVRP with time windows is solved, and routes are obtained first, and only after that, for each route are the items packed in the vehicles. Also, their paper introduces benchmark instances for the 3L-CVRP.

Table 3.1 Detailed information about the literature on the 3L-CVRP

Authors	Solution methods		ToV		Problem characteristics and constraints								
	Heuristic	Exact	Hmg	Het	TW	CLC	WL	WD	LIFO	FR	PD	BH	RB
Gendreau et al. (2006)	TSA		√			√	√		√	√			
Moura & Oliveira (2009)	GRASP		√		√	√							
Fuellerer et al. (2010)	ACO		√			√	√		√	√			
Bartók & Imreh (2011)	LS		√			√	√				√		
Bortfeldt (2012)	TSA+TRSA		√			√	√		√	√			
Zhu et al. (2012)	TSA		√			√	√		√	√			
Miao et al. (2012)	GA+TSA					√	√		√	√			
Ruan et al. (2013)	HBMO		√			√	√		√	√			
Bortfeldt & Homberger (2013)	P1R2		√		√	√	√		√	√			
Junqueira et al. (2013)		MIP	√			√	√		√	√			
Lacomme et al. (2013)	GRASP/ELS		√			√	√		√	√			
Zachariadis et al. (2013)	LS		√		√	√	√		√	√	√		
Ceschia et al. (2013)	SA+LNS		√	√		√	√		√	√			
Wei et al. (2014)	AVNS			√		√	√		√	√			
Zhanga et al. (2015)	ELS		√			√	√		√	√			
Junqueira & Morabito (2015)	SA+RTA		√			√	√			√			
Mahvash et al. (2015)		CG	√			√	√		√	√			
Tao & Wang (2015)	TSA		√			√	√		√	√			
Pace et al. (2015)	ILS+ SA			√	√	√	√		√	√			
Männel & Bortfeldt (2016)	ALNS		√			√	√		√	√	√		
Escobar-Falcón et al. (2016)	MIP+GRASP		√			√	√		√	√			
Männel & Bortfeldt (2018)	LNS+TRSA		√			√	√		√	√	√		√
Reil et al. (2018)	ES+TS		√			√	√		√	√		√	
Mak-Hau et al. (2018)		MIP		√	√	√	√		√				
Ngoenriang et al. (2019)	SPA		√			√	√		√	√			
Moura (2019)	MBH	MIP	√		√	√	√						
Vega-Mej´ıa et al. (2019)		NLMIP	√			√	√	√	√	√			
Vega-Mej´ıa et al. (2020)	**GRASP**		√		√	√	√	√	√	√			
Bortfeldt & Yi (2020)	GA+LS		√			√	√		√	√			
Ayough et al. (2020)	SA		√		√	√							
Krebs & Ehmke (2021)	ALNS+DBFL		√			√	√	√	√	√			
Rajaei et al. (2022)		CG		√		√	√		√	√			
Chapter 4; Küçük & Yildiz (2022)	CP+EA	CP+MIP	√		√	√	√						
Rodríguez et al. (2022)	CG+GRASP		√			√	√		√	√			
Krebs et al. (2023)	ALNS		√			√	√		√	√			
Chapter 5	ES		√			√	√		√	√			
Chapter 6	ES		√		√	√	√	√	√	√	√		

ToV: Type of Vehicle fleet Het: Heterogeneous, Hmg: Homogeneous TW: Time Windows, CPC: Classical Loading Constraints, WL: Weight Limit, WD: Weight Distribution, LIFO: LIFO Loading, FR: Stacking (Fragility), PD: Pickup and Delivery, BH: Backhauls, RB: Reloading Ban

Wang, Tao & Shi (2009) presented a survey paper on the 2L-CVRP and 3L-CVRP. Later, Iori & Martello (2010) and Pollaris, Braekers, Caris, Janssens & Limbourg (2015) provided detailed literature reviews on the 2L/3L-CVRPs. Finally, Bortfeldt & Wäscher (2013) presented a survey in which practical constraints of the container loading problem are identified and categorized.

Fuellerer, Doerner, Hartl & Iori (2010) reconsidered the 3L-CVRP using a modified savings-based Ant Colony Optimization (ACO) algorithm with fast and straightforward loading heuristics. Bortfeldt (2012) developed an efficient hybrid algorithm to solve the 3L-CVRP, which includes a tabu search algorithm (TSA) for routing and a tree search algorithm for loading (TRSA). Miao, Ruan, Woghiren & Ruo (2012) introduced a hybrid Genetic Algorithm (GA) for the 3L-CVRP in which a GA solves the routing sub-problem, and a TSA solves the loading sub-problem. Zhu, Qin, Limand & Wang (2012) divided the problem into two parts: finding a good solution to the loading sub-problem and developing an effective general algorithm to solve the routing problem, using the loading sub-problem as a subroutine. The tabu search was adapted to the general algorithm, which has proven effective for VRPs.

When solving the 3L-CVRP, Ruan, Zhang, Miao & Shen (2013) discussed two objective functions that minimize the number of vehicles and the total travel cost and maximize the number of items loaded on vehicles. Their approach consists of Honey Bee Mating Optimization (HBMO) for routing and six loading heuristics for loading problems. Bortfeld & Homberger (2013) proposed a heuristic method called "Packing First, Routing Second" (P1R2), which minimizes the number of vehicles used with higher priority and minimizes the total travel distance with lower priority. The authors also discussed under what conditions the routing and loading problems should be treated as a combined task. Junqueira, Oliveira, Carravilla, & Morabito (2013) and Mak-Hau, Moser & Aleti (2018) proposed a new MIP model for the 3L-CVRP to minimize the overall cost. Ceschia, Schaerf & Stützle (2013) considered a different LIFO constraint approach, such as manual reloading instead of a forklift. They proposed a local search approach based on Simulated Annealing (SA) and Large-Neighbourhood Search (LNS) to solve the integrated problem in one stage.

Lacomme, Toussaint & Duhamel (2013) proposed a hybrid algorithm using the GRASP and Evolutionary Local Search (ELS) metaheuristics, combining the positive characteristics of both methods. Wei, Zhang & Lim (2014) developed an adaptive variable neighborhood search (AVNS) for solving this challenging problem with the heterogeneous fleet of vehicles. Mahvash, Awasthiand & Chauhan (2015) presented a column generation (CG) technique-based heuristic for the 3L-CVRP using the shortest path problem and heuristic pricing method. Later, Tao & Wang (2015) designed a straightforward TSA for the routing problem and recursively called the loading sub-problem. Pace, Turky, Moser & Aleti (2015) applied SA and Iterative Local Search (ILS) methods for the routing problem of fiberboard distribution with 3D loading constraints. Two hybrid algorithms for solving the VRP with clustered backhaul and packing of three-dimensional items have been proposed by Bortfeldt et al. (2015); in the first algorithm, an Adaptive Large Neighborhood Search (ALNS) procedure is applied to routing while a Variable Neighborhood Search (VNS) procedure is responsible for loading in the second algorithm.

Zhang, Wei & Lim (2015) developed an ELS method, an improved version of ILS for the 3L-CVRP, minimizing the fuel consumption of active vehicles. In the study of Junqueira & Morabito (2015), they employed two metaheuristic procedures, SA and record-to-record travel algorithms (RTA), to obtain better results for the 3L-CVRP.

Zachariadis, Tarantilis & Kiranoudis (2013) suggested a solution approach built with a local search-based framework to solve the pickup and delivery routing problem with time windows and pallet loading. Escobar-Falcón, Álvarez-Martínez, Granada-Echeverri, Escobar & Romero-Lázaro (2016) developed an exact approach for solving the routing subproblem and a GRASP algorithm for solving the loading problem for the 3L-CVRP. The PDP with 3D loading constraints was first discussed by Bartók & Imreh (2011). They specified a local search method to solve a PDP variant that neglects the LIFO constraint and does not account for reload effort. Later, Männel & Bortfeldt (2016) described a new hybrid algorithm for the VRP with pickup and delivery and three-dimensional loading constraints (3L-PDP), consisting of two algorithms. The routing algorithm is adapted from the ALNS heuristic for solving the pickup and delivery problems with time windows and the tree search for the packing algorithm. The ALNS heuristic is also used for the 3L-CVRP in Krebs & Ehmke (2021). Männel & Bortfeldt (2018) proposed a hybrid algorithm based on LNS for routing and TRSA for loading to solve the 3L-PDP with a reloading ban constraint that ensures identical placement of the same boxes in different loading plans.

Reil, Bortfeldt, & Mönch (2018) developed a new algorithm for the VRP with backhaul, time windows, and three-dimensional loading constraints, which integrates the TSA for loading and a multi-start evolutionary strategy for routing. Ngoenriang, Sawadsitang, Leangsuksun, Niyato, & Tan (2019) proposed a stochastic programming approach (SPA) that finds an optimal delivery route, loading customers' orders and trading off between using in-house delivery facilities and outsourcing items to a carrier under uncertainty of customer demand. Moura (2019) proposed a MIP model and a model-based heuristic (MBH) to solve the 3L-CVRP with time windows. Vega-Mejía, Montoya-Torres & Islam (2019) presented a non-linear mixed-integer programming model (NLMIP) for the 3L-CVRP. The model is solved by applying a simple procedure that ignores the nonlinearity of the problem. They considered the weight-bearing strength of boxes, customer time windows, weight limits of vehicles, weight distribution, loading, and fragility of products. Bortfeldt & Yi (2020) explored the CVRP with split delivery and three-dimensional loading constraints by ignoring the only one-visit assumption. It was formulated similarly to that of Gendreau et al. (2006). They proposed a hybrid algorithm consisting of an LS algorithm for routing and a GA for loading. Ayough, Khorshidvand, Massomnedjad, & Motameni (2020) proposed a new integrated approach to the 3L-CVRP with time windows using an SA-based heuristic algorithm. Vega-Mejía, González-Neira, Montoya-Torresand, & Islam (2020) proposed a hybrid heuristic method based on the GRASP method and the Clarke and Wright Savings algorithm. Eventually, Rajaei, Moslehi, & Reisi-Nafchi (2022) proposed a CG-based heuristic that uses three heuristics to solve the subproblem and a hybrid algorithm for loading. Rodrígueza, Martíneza, & Escobarb (2022) presented a metaheuristic algorithm based on a CG structure for the 3L–CVRP. In the CG approach, the master problem manages the selection of the best-set routes, and the slave problem solves the Constrained Shortest Path (CSP) problem for feasible routes. Krebs, Ehmke, & Koch (2023) proposed a hybrid algorithm consisting of an outer ALNS tackling the routing problem in combination with an inner (Deepest-Bottom-Left-Fill) DBFL algorithm solving the loading problem.

Regarding the relevant literature, no study has used constraint programming (CP) for solving the 3L-CVRP. This thesis proposes integrated and decomposed CP-based solution methods for the first time, which have already been published by Küçük and Yildiz (2022). These solution approaches are presented in detail in Chapter 4. In Chapter 5, an evolutionary heuristic approach is proposed to solve the 3L-CVRP involving a large number of customers. This heuristic is a new version of the evolutionary algorithm (EA) based on the shifting elitism rate.

The 3L-CVRP is studied with pickup and delivery constraints in Chapter 6. The problem with these constraints, referred to as the 3L-PDP, has become much more complex with handling new constraints such as request sequencing and LIFO loading. In this thesis, in addition to the constraints discussed in previous studies in the literature (Bartók & Imreh, 2011; Zachariadis et al., 2013; Männel & Bortfeldt, 2016; Männel & Bortfeldt, 2018), Time Window (TW) and Weight Distribution (WD) constraints are also considered. As a solution method, the EA presented in Chapter 5 is adapted to the 3L-PDP, and detailed computational analysis is performed using benchmark and newly generated instances.

3.3 Chapter Summary

In this chapter, a detailed literature review of the 3L-CVRP is presented. When the relevant literature is examined, it is seen that there is no study with a CP approach for the 3L-CVRP. Furthermore, in the literature, it has been seen that in the solution of the 3L-CVRP problem, complex loading constraints and the pickup and delivery processes are not handled simultaneously.

In this thesis, the 3L-PDP, the most generalized version of the 3L-CVRP, is solved using the EA heuristic with the TW and WD constraints discussed for the first time.

Considering this field's still uncompleted research potential, this thesis aims to create solution methods to complete these two important shortcomings. Regarding the point of the literature on the problem, the proposed solution methods and the additional constraints considered in this thesis offer a wide projection for future studies. Based on the fact that there are advanced solution methods required for the complexity of real-world problems, the development of heuristic and exact methods should be the primary task of scientists.

CHAPTER FOUR
SOLUTION APPROACHES FOR THE 3L-CVRP

4.1 Introduction

Optimizing vehicle use and goods loading has created an integrated vehicle routing and loading problem in distribution logistics. In 2006, Gendreau et al. first formulated and solved this problem, namely, The Capacitated Vehicle Routing Problem (CVRP) with Three-dimensional Loading (3L) constraints (3L-CVRP). The 3L-CVRP can be defined very concisely as finding a set of routes that meet all customer demands while minimizing the total routing cost with good packing patterns of items (Bortfeldt et al., 2015).

The problem addressed in this chapter is a combination of vehicle routing and three-dimensional loading problems with various packing constraints that often arise in real-world environments (Gendreau et al., 2006). Unlike the classical CVRP, which is the most investigated problem in combinatorial optimization, rectangular box sets represent customer demand in this problem. A three-dimensional rectangular loading space replaces the scalar capacity of a vehicle. A logistics expert must set out a series of vehicle fleet routes to ship goods consisting of rectangular boxes of specific sizes and weights requested by each customer while minimizing travel costs. All goods a customer requires are placed in the same vehicle, and each vehicle performs only one tour. Thus, the number of routes is the same as the number of vehicles. Each route starts and ends at the depot, containing at least one customer, and each customer is visited just once. The visit to each customer is planned within a specified time window. According to heterogeneous/homogenous vehicle fleets, vehicles can have the same or distinct weight capacity and three-dimensional loading space. For each vehicle, the problem's solution must contain a set of items whose total weight does not exceed the vehicle capacity and a feasible packing of these items in the loading space. Each vehicle must have a non-overlapping packing of a set of rectangular boxes, i.e., no two boxes occupy the same portion of its rectangular container (Vega-Mejía et al., 2020).

Moreover, this packing procedure allows all boxes to rotate in three axes with permitted orientation. Also, stable packing is necessary because the boxes do not suffer any damage during transportation (Junqueira et al., 2013). Boxes can be packaged in multiple layers of boxes of different sizes and shapes to minimize vehicle volume use (Talbi, 2013). In addition to the above weight and packing constraints, various operational constraints may be often encountered in real-world applications. The loading-related constraints have already been discussed in Section 2.5. As a result, the 3L-CVRP involves the three-dimensional allocation of mixed-size boxes to the vehicles' compartments while minimizing the total travel cost. Because of its potential for practical applications, the 3L-CVRP is a concentrated research stream in logistics, and several solution approaches have been proposed to solve different variants of this problem (Moura & Oliveira, 2009). The vehicle routing and container loading problems (CLP) are individually NP-hard. Thus, the integrated 3L-CVRP is also an NP-hard problem.

The contributions of this chapter are considered in two stages. First, we developed a constraint programming (CP) model of the 3L-CVRP to create an exact solution method to find optimal solutions for small-size problems. Optimal solutions derived from the CP model can be used for performance comparison with metaheuristic approaches (Talbi, 2013). This is the first time the CP model has searched for an optimal solution to the 3L-CVRP. Unlike the MIP formulation, the CP model does not need many decision variables and constraints, and difficult constraints such as planning customer visits, removing sub-tours, and adhering to time window restrictions can be expressed easily. Also, several search strategies provided by the CP optimizer can efficiently solve the problem. Second, we developed CP-based decomposition models inspired by the existing decomposed algorithms in the literature. This way, we shortened the computation time and improved the solution quality. The most competitive decomposition model (CP&EA) that stands apart is the one that uses CP for the vehicle routing part of the 3L-CVRP and an Evolutionary Algorithm (EA) for the loading part.

4.2 The Mixed-Integer Programming Model for the 3L-CVRP

This mathematical model given in this section is based on Mauro & Oliveira's (2009) and Mauro's (2019) models. The CVRP formulation can be used as a starting point to model the 3L-CVRP (Junqueira et al., 2013). The CVRP can be described using graph theory terms as follows: Let $G(N, A)$ be a graph having a node set N and an arc set A. For i and $j \in N$, we indicate the arc from i to j as $(i, j) \in A$. We use the term "network" to mean a graph having additional data on its nodes and arcs. $N = \{0,1,2, \ldots, n+1\}$ is the set of nodes, in which 0 represents the initial depot, and $n+1$ represents the final depot. Additionally, each node $i \in N$ has a load q_i and a positive service duration d_i. The loads and durations of the depots are equal to 0 ($q_0 = q_{n+1} = 0$, $d_0 = d_{n+1} = 0$). A limitless fleet of different or identical vehicles with usage cost cv^k and capacity Q_k is ready to meet customer demands. Each arc $(i, j) \in A$ has a travel cost c_{ij} connected to travel distance or time t_{ij}. Moreover, each node also has a time window $[a_i, b_i]$, where a_i and b_i indicate the possible earliest and latest service start times, respectively. The service may start between these times at node i. The depot usually has time windows by representing $[a_0, b_0]$ and $[a_{n+1}, b_{n+1}]$. There is also a set of boxes that must be delivered to the customers $O = \{1, \ldots, o\}$. Each box t is characterized by length (l_t), width (w_t), and height (h_t) dimensions. β_{tps}^k binary variables determine whether the dimensions $p \in \{l_t, w_t, h_t\}$ of box t are aligned with the dimensions $s \in \{L^k, W^k, H^k\}$ of vehicle k. The notation required for formulating the mathematical model of the 3L-CVRP is introduced below;

Indices:

i, j	indices for nodes
k	index for vehicles
t, r	indices for boxes
g	index for overlapping, $g \in \{1,2, \ldots, 6\}$

Sets:

N	set of all nodes, $i, j \in \{0,1,2, \dots, n+1\}$
N'	set of customers, $i, j \in \{1,2, \dots, n\}$
V	set of all vehicles, $k \in \{1,2, \dots, m\}$
O	set of all boxes, $b \in \{1,2, \dots, o\}$

Parameters:

Q^k	the capacity of vehicle k
cv^k	cost of using vehicle k
c_{ij}	traveling cost between nodes i and j
t_{ij}	traveling time between nodes i and j
d_i	service duration of customer i
q_t	load amount of single box t
$L^k,\ W^k, H^k$	length, width, and height of vehicle k
l'_t, w'_t, h'_t	length, width, and height of unrotated box t
l_t, w_t, h_t	length, width, and height of box t
e_{t_i}	1, if box t belongs to customer i and 0, otherwise
a_i, b_i	service start and end times for customer i
M	a large number

Decision variables:

x_{ij}^k 1, if arc (i, j) is traversed by vehicle k, 0 otherwise

y^k 1, if vehicle k is used and 0 otherwise

Y_t^k 1, if box t is transported by vehicle k and 0 otherwise

s_i^k visit start time of vehicle k at node i

Q_i^k load of vehicle k when leaving node i

$zx_{tk}, zy_{tk}, zz_{tk}$ placement coordinates of box t inside vehicle k

T_{gtrk} a binary variable for non-overlapping of boxes t and r for vehicle k

β_{tps}^k a binary variable for orientation of box t for vehicle k with $p \in \{l, w, h\}$ and $s \in \{L, W, H\}$

Objective function:

$$Minimize \sum_{k\in V}\sum_{i\in N}\sum_{j\in N} c_{ij}\ x_{ij}^k + \sum_{k\in V} cv^k y^k \tag{4.1}$$

Subject to:

$$\sum_{k\in V}\sum_{j\in N} x_{ij}^k = 1, \qquad \forall i \in N' \tag{4.2}$$

$$\sum_{j\in N} x_{0j}^k \quad = y^k, \qquad \forall k \in V \tag{4.3}$$

$$\sum_{i\in N} x_{i(n+1)}^k \ = y^k, \qquad \forall k \in V \tag{4.4}$$

$$\sum_{i\in N} x_{ij}^k - \sum_{i\in N} x_{ji}^k \ = 0, \qquad \forall j \in N', \forall k \in V \tag{4.5}$$

$$s_j^k \geq s_i^k + d_i + c_{ij} - M\left(1 - x_{ijk}\right), \qquad \forall i, j \in N, \forall k \in V \tag{4.6}$$

$$\sum_{t \in O} Y_t^k \; q_t \leq Q^k, \qquad \forall k \in V \tag{4.7}$$

$$\sum_{k \in V} Y_t^k = 1, \tag{4.8}$$

$$\sum_{j \in N} x_{ij}^k = Y_t^k, \qquad \forall k \in V, \forall i \in N, \forall t \in O, e_{t_i} = 1 \tag{4.9}$$

$$a_i \leq s_i^k \leq b_i, \qquad \forall i \in N, \forall k \in V \tag{4.10}$$

$$zx_{tk} - L^k + l_t \leq \left(1 - Y_t^k\right)M, \qquad \forall t \in O, \forall k \in V \tag{4.11}$$

$$zy_{tk} - W^k + w_t \leq \left(1 - Y_t^k\right)M, \qquad \forall t \in O, \forall k \in V \tag{4.12}$$

$$zz_{tk} - H^k + h_t \leq \left(1 - Y_t^k\right)M, \qquad \forall t \in O, \forall k \in V \tag{4.13}$$

$$zx_{rk} + l_r \leq zx_{tk} + T_{1trk}M, \qquad \forall t, r \in O, r \neq t, \forall k \in V \tag{4.14}$$

$$zx_{tk} + l_t \leq zx_{rk} + T_{2trk}M, \qquad \forall t, r \in O, r \neq t, \forall k \in V \tag{4.15}$$

$$zy_{rk} + w_r \leq zy_{tk} + T_{3trk}M, \qquad \forall t, r \in O, r \neq t, \forall k \in V \tag{4.16}$$

$$zy_{tk} + w_t \leq zy_{rk} + T_{4trk}M, \qquad \forall t, r \in O, r \neq t, \forall k \in V \tag{4.17}$$

$$zz_{rk} + h_r \leq zz_{tk} + T_{5trk}M, \qquad \forall t, r \in O, r \neq t, \forall k \in V \tag{4.18}$$

$$zz_{tk} + h_t \leq zz_{rk} + T_{6trk}M, \qquad \forall t, r \in O, r \neq t, \forall k \in V \tag{4.19}$$

$$\sum_{g=1}^{6} T_{gtrk} \leq 5 + (1 - Y_r^k) + \left(1 - Y_t^k\right), \quad \forall t, r \in O, r > t, \forall k \in V \tag{4.20}$$

$$\sum_{k \in V} (\beta_{t11}^k \, l_t' + \beta_{t12}^k \, w_t' + \beta_{t13}^k \, h_t') = l_t, \qquad \forall t \in O \tag{4.21}$$

$$\sum_{k \in V} (\beta_{t21}^{k}\, l_t' + \beta_{t22}^{k}\, w_t' + \beta_{t23}^{k}\, h_t') = w_t, \qquad \forall t \in O \tag{4.22}$$

$$\sum_{k \in V} (\beta_{t31}^{k}\, l_t' + \beta_{t32}^{k}\, w_t' + \beta_{t33}^{k}\, h_t') = h_t, \qquad \forall t \in O \tag{4.23}$$

$$\sum_{k \in V} \sum_{s \in \{1..3\}} \beta_{tps}^{k} = 1\,, \qquad \forall t \in O, \forall p \in \{1..3\} \tag{4.24}$$

$$\sum_{k \in V} \sum_{p \in \{1..3\}} \beta_{tps}^{k} = 1\,, \qquad \forall t \in O, \forall s \in \{1..3\} \tag{4.25}$$

$$x_{ij}^{k} \in \{0,1\}, \qquad \forall k \in V, \forall i, j \in N \tag{4.26}$$

$$y^{k} \in \{0,1\}, \qquad \forall k \in V \tag{4.27}$$

$$zx_{tk}, zy_{tk}, zz_{tk} \geq 0, \qquad \forall t \in O, \forall k \in V \tag{4.28}$$

$$T_{gtrk} \in \{0,1\}, \qquad g \in \{1..6\}, \forall t, r \in O, \forall k \in V \tag{4.29}$$

$$Y_t^{k} \in \{0,1\}, \qquad \forall k \in V, \;\; \forall t \in O \tag{4.30}$$

The objective function (4.1) minimizes the cost of the total distance traveled and vehicles used. Constraint (4.2) requires that each customer be visited once by one vehicle. Constraint (4.3) guarantees that a vehicle departs the depot only if it is active. Constraint (4.4) guarantees that each active vehicle returns to the depot. Constraint (4.5) is a flow conservation constraint; when a vehicle visits a customer, it must go to another customer from that customer. Constraint (4.6) ensures that customer service does not start before the vehicle arrives at the customer. Constraint (4.7) ensures that the weight capacity of vehicles is not exceeded. Constraint (4.8) ensures that each box is carried by only one vehicle. Constraint (4.9) links loading variables to routing variables; if a box is loaded inside a vehicle, that vehicle must visit the customer to which this box belongs. Constraint (4.10) guarantees that the visits occur within the time windows. Constraints (4.11)–(4.13) ensure that each box is entirely inside the vehicle. Constraints (4.14)–(4.20) prevent overlapping the two boxes packed inside the same vehicle. It is sufficient to guarantee no overlap with at least one of the axes to prevent the two boxes from occupying the same part of the space. T_{gtrk}'s are extra binary variables used to check the intersection of two boxes on the *x*, *y*, and *z* axes. For example, if there is an intersection between the t and r boxes on the x-axis; T_{1trk} and T_{2trk} are equal to 1; otherwise, one of them equals 0. Constraints (4.21)–(4.25) ensure the orientation limitations of each box. For instance, if box t in vehicle k has a "this way up" limitation, the variable β_{t33}^{k} must be equal to one. Thus h_{t}'determined as the initial and unrotated height value becomes the precise height value of the box ($h_t' = h_t$). Finally, the constraints (4.26)–(4.30) define the decision variables.

4.3 Constraint Programming

CP is a robust technique for solving common combinatorial search problems based on various techniques such as artificial intelligence, operations research, and graph theory. CP formulates the 3L-CVRP as a constraint satisfaction problem that assigns a suitable value to every variable to satisfy the constraints. Thus, CP is an exact search technique based on modeling and constraint satisfaction techniques (Apt, 2003). As a usual way of solving a CP model, the user specifies a real-life problem represented using decision variables and constraints, and a CP solver is used to solve them. CP is successfully applied to combinatorial problems with tightly constrained features, such as the traveling salesman problem (Pesant, Gendreau, Potvin & Rousseau, 1998), the vehicle routing problem (Shaw, 1998; Rousseau, Gendreauand, & Pesant, 2002; Rousseau, Gendreau, Pesant & Focacci. 2004; Quoc & Anh, 2010), the team orienting problem (Gedik, Rainwater, Nachtmann & Pohl, 2016) and the pickup and delivery problem (Küçük & Yıldız, 2019).

Hybrid approaches combining meta-heuristics and CP have recently grown with effective optimization software, mathematical programming structures, and advanced modeling techniques. Increasingly, hybrid approaches combining meta-heuristic methods with CP are used to solve optimization problems. While CP is successfully applied for tightly constrained combinational optimization problems, metaheuristics perform better for unconstrained optimization problems. Naturally, metaheuristics and

CP effectively combine search and modeling approaches because of their complementary properties (Focacci, Lodi & Milano 2002). For example, CP models are based on a declarative programming paradigm, so adding or deleting new constraints is simple. This flexibility is the main advantage of hybrid approaches.

Despite the upward trend, few hybrid strategies have been studied, combining metaheuristics and CP in the literature. Hybrid strategies create an effective architecture for optimization algorithms in which data are exchanged during the search by applying different solution strategies to solve the target problem. For example, the exchanged data might include additional constraints, optimal solutions for subproblems, and partial solutions (Talbi, 2016). Wallace (2006) introduced a survey of hybrid algorithms from a CP perspective. Qu & He (2008) investigated a two-stage hybrid approach consisting of CP and VNS. Finally, Ahmeti & Musliu (2021) proposed a hybrid approach combining CP and a meta-heuristic algorithm for solving multiple project scheduling problems with resource constraints.

4.3.1 Integrated Constraint Programming Models

Unlike MIP models, the formulation of CP models depends heavily on CP solvers. There is no standard for CP modeling representation. In this thesis, we used keywords and global constraints of IBM ILOG CP Optimizer adapted from the modeling language of Laborie (2009). When developing the integrated CP model of the 3L-CVRP, we used interval variables for two cases. The first case is related to the routing part of the problem. The interval variable determines whether an activity occurs and its position on the timeline. In the second case, the interval variable is used to define the positions of the boxes in the compartment of vehicles for the loading part of the problem. When we define an activity using an interval variable with an optional attribute, no additional constraint is required to provide a binary correlation for the absence or presence of activity in the timeline. For example, suppose an optional interval variable is absent in the final solution; in that case, it can be easily understood that its domain is empty and does not exist in the final solution. In another situation, if it exists in the final solution, it can be said that its domain consists of a single value, which is set out by its "start" and "end" times. In the CP formulation, the final state of an interval variable can be queried using the $presenceOf(Interval\ Variable)$ function.

We developed two integrated CP models. In the first integrated CP model (CP-1A), we use interval variables for the loading part, as used in the routing part. In the second integrated CP model (CP-2A), we use integer variables instead of interval variables for the routing part.

4.3.1.1 The Routing Part of the CP model

The interval variables used in the routing part of the proposed CP models are possible time intervals representing the start and end times of a vehicle's customer visit. The duration of customer service time corresponds to the size of interval variables (Gedik et al., 2017). The start time of customer service does not necessarily correspond to the start time of the customer visit, which is the arrival time to the customer node, as the vehicle can wait until the service start time. Figure 4.1 shows an interval variable representing the customer visit start and end times, service start and end times, and service time.

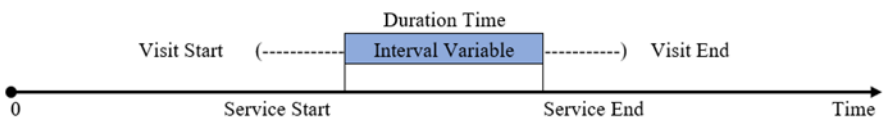


Figure 4.1 Representation of a customer visit by an interval variable

x_{ik} is an interval variable representing whether customer i is visited using vehicle k, and the duration time of this visit is d_i. Since the demand of each customer must be satisfied, an interval variable for each customer visit must exist in the solution. The $StartMin$ and $EndMax$ parameters, representing the earliest start time of the visit and the latest completion time, are determined according to the time window constraints of the customer. $\theta_k = \{x_{0k}, x_{1k}, x_{2k}, \dots, x_{ik}, \dots, x_{(n+1)k}\}$ is a collection of optional interval variables of vehicle k that represents visits to each customer $i \in N'$. This collection of interval variables is called a sequence variable for each vehicle and is used to investigate the feasibility of each vehicle's visit sequence.

In this thesis, we used several global constraints such as $Alternative, NoOverlap, Pulse$, which are generated by the CP Optimizer of IBM. The $Alternative$ $\{X_i, \{x_{i1}, . . ., x_{im}\}$ global constraint chooses a single alternative between interval variables $\{x_{i1}, . . ., x_{im}\}$. This constraint enforces that only one of the decision variables from the collection of $\{x_{i1}, . . ., x_{im}\}$ is chosen for the corresponding interval variable X_i that denotes visiting customer i to satisfy the demand. Furthermore, X_i starts and ends together with the chosen interval variable from the collection of $\{x_{i1}, . . ., x_{im}\}$. This constraint is used to assign a visit to each customer by only one vehicle. $TransitionTime(t_{ij})$ is a function to create a two-dimensional array that stores the distances between all pairs of nodes. This function is used with another global constraint called $NoOverlap\,\{x_{i1}, . . . x_{im}\}$. The $NoOverlap$ constraint featured with $TransitionTime\,(t_{ij})$ function obtains a sequence of non-overlapping interval decision variables with a minimum time between each consecutive variable in the collection of $\{x_{i1}, . . . x_{im}\}$. The $NoOverlap$ constraint featured with $TransitionTime(\,t_{ij})$ matrix ensures that a vehicle does not visit more than one customer at any particular time. It also allows a customer to be visited by a vehicle traveling at a certain speed. The time to visit the next customer is obtained by adding the time to leave the customer and the travel time between the next customer.

Furthermore, $Cumulative$ is another global constraint that tracks the cumulated utilization of resources related to customer visits as a function of time. Several elementary cumulative functions can be used to analyze the impact of each interval variable during the activity time. For example, $Step$ function is used to obtain the amount of resource consumption. Using the $Step$ function and the $Cumulative$ constraint together, the load of the vehicles during the tour is limited to be less than or equal to their capacity.

4.3.1.2 The Loading Part of the CP model

In the first version of the integrated CP model (CP-1A), we used interval variables to load the parts of CP-1A as decision variables. As far as we know, this is the first study that uses interval variables to express the position of an item. The bx_t, by_t, and bz_t are interval variables representing the position of box t on the x, y, and z axes with the starting and ending points associated with these variables. The $sizeOf$ expression is used to access the size of the interval variables bx_t, by_t, and bz_t and specifies the length, width, and height $(l_t, w_t,\ h_t)$ dimensions of box t, respectively. The walls of the vehicle containers must be defined to fit the boxes in the containers. This identification is made using the $StartMin$ and $EndMax$ parameters for the interval variables. They represent the boundaries of time intervals where we place the interval variables. $StartMin$ takes the value 0, and $EndMax$ gets the length, width, and height (L^k, W^k, H^k) of the container for vehicle k. In this way, there is no need to write an additional constraint in the CP model to prevent boxes from overlapping the container walls. Figure 4.2 illustrates the positioning of two boxes using interval variables. The constraints that prevent the boxes from overlapping are provided by the expressions $startOf$ and $endOf$ that specify the start and end times of interval variables. Also, the $presenceOf$ constraint provides a link between the routing and loading parts of the problem. If interval variable x_{ik} is present, the $presenceOf(x_{ik})$ constraint returns a true value, which is 1; otherwise, it returns false, which is 0.

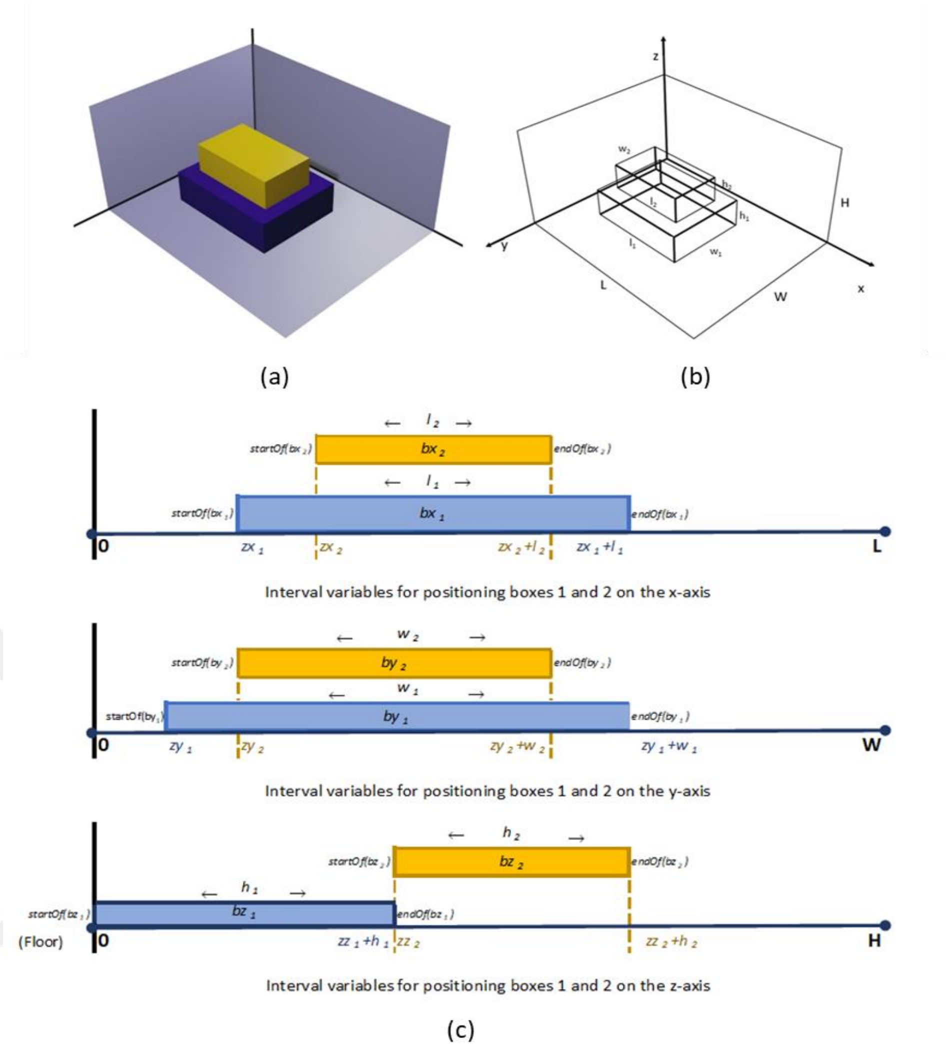


Figure 4.2 Packing representation of two boxes using interval variables

4.3.1.3 The Formulation of the CP-1A Model

Additional notation required for the representation of the CP-1A model is as follows:

Decision variables:

x_{ik} optional interval variable that represents visiting node i using vehicle k

X_i mandatory interval variable representing visiting node i and the service time and time window assigned for that visit; $X_i.size = d_i$, $X_i.startMin = a_i$, $X_i.startMax = b_i$

θ^k interval sequence variable representing a set of interval variables (x_{ik}) that express possible visits to nodes by the same vehicle k

bx_t interval variable representing the x-axis position of box t in the 3D Cartesian coordinate system, $bx_t.size = l_t$, $bx_t.startMax = L^k - l_t$

by_t interval variable representing the y-axis position of box t in the 3D Cartesian coordinate system, $by_t.size = w_t$, $by_t.startMax = W^k - w_t$

bz_t interval variable representing the z-axis position of box t in the 3D Cartesian coordinate system, $bz_t.size = h_t$, $bz_t.startMax = H^k - h_t$

ρ_t^k a binary variable for the presence of box t in vehicle k

Objective function:

$$Minimize \sum_{i \in N} \sum_{k \in V} \sum_{j=typeOfNext(\theta_k, x_{ik}, i, i)}^{j} c_{ij} + \sum_{k \epsilon V} cv^k y^k \tag{4.31}$$

Subject to:

$$Alternative\ (X_i, \{x_{i1}, x_{i2} \dots x_{im}\})\ , \forall i \in N' \tag{4.32}$$

$$Cumulative\ \left(Step(x_{ik}, q_t | i \in C, t \in O, e_{t_i} = 1)\right) \leq Q^k, \forall k \in V \tag{4.33}$$

$$NoOverlap\ \left(\theta_k, TransitionTime\left(t_{ij} \middle| i, j \in N\right)\right), \forall k \in V \tag{4.34}$$

$$\theta_k.First(X_0), \forall k \in V \tag{4.35}$$

$$\theta_k.Last(X_{n+1}), \forall k \in V \tag{4.36}$$

$$presenceOf(x_{ik}) = 1 => \rho_t^k = y^k = 1, \forall k \in V, \forall t \in O \mid e_{t_i} = 1 \tag{4.37}$$

$$\rho_t^k = \rho_r^k = 1 => \{endOf(bx_t) \leq startOf(bx_r) \, V \, endOf(bx_r) \leq startOf(bx_t) \quad V endOf(by_t) \leq startOf(by_r) \, V \quad endOf(by_r) \leq startOf(by_t) \, V \, endOf(bz_t) \leq startOf(bz_r) \quad V \quad endOf(bz_r) \leq startOf(bz_t)\}, \quad \forall t, r \in O | t \neq r, \forall k \in V \tag{4.38}$$

The objective function (4.31) minimizes the cost of total distance traveled and vehicles used. We used the $typeOfNext\ (\theta_k, x_{ik}, i, i)$ function to find the index j in the first aggregation term. This function returns the first index of the consecutive interval variable after x_{ik} in the θ_k. This function returns the value i if x_{ik} does not exist, or is the last interval variable of the θ_k. The $Alternative$ constraint (4.32) guarantees that each customer is visited by one vehicle. The $Cumulative$ (4.33) ensures that the load of the vehicle does not exceed its capacity while it is touring. The load of the vehicle during the activity period is calculated by sub-function $Step(x_{ik})$. Visit time windows determined for each customer are specified as the start and end times of the interval variables related to the customer visit. The $NoOverlap$ constraint (4.34) ensures that the interval variables in sequence θ_k, indicating the possible visits of vehicle k , do not overlap each other. Besides, with the help of $TransitionTime(t_{ij})$, the $NoOverlap$ global constraints obtain a minimum travel time (t_{ij}) between the end time of the interval variable x_{ik} and the start time of the interval variable x_{jk}, which represents the consecutive visits of vehicle k. Constraints (4.35)and (4.36) make the initial depot the first (X_o) and the final depot (X_{n+1}) the last node to be visited by each vehicle k. Constraints (4.37) connect with boxes and vehicles; a box must be loaded onto the vehicle visiting its owner. Constraint (4.38) ensures that the boxes do not overlap in loaded vehicles.

4.3.1.4 The Formulation of the CP-1B Model

Instead of using bx_t, by_t, and bz_t interval variables for the loading part of the CP-1A model, we use px_t, py_t, and pz_t integer decision variables representing the front-left-bottom corner coordinates of box t on the x, y, and z axes, respectively. For instance, px_t indicates the position of the front-left-bottom corner to the x-axis origin of the coordinate system. The rest of the model is the same as the previous model. In this version of the model, we updated the non-overlap constraint (4.39) as follows:

$$\rho_t^k = \rho_r^k = 1 => (px_t + l_t) \leq px_r \, V \, (px_r + l_r) \leq px_t \, V \, (py_t + w_t) \leq py_r \, \mathrm{V} \, (py_r + w_r) \leq py_t \mathrm{V} \, (pz_t + h_t) \leq pz_r \, \mathrm{V} \, (pz_r + h_r) \leq pz_t, \forall k \in V, \forall t, r \in O | t \neq r \tag{4.39}$$

4.3.1.5 Symmetry-Breaking Constraints

Computing time is wasted as new solutions are found that are symmetrical to those already visited. Symmetry-breaking constraints can reduce the size of the CP solver search tree by removing these symmetric solutions. The routing part of our problem assumes that the vehicles are homogeneous. Unfortunately, numerous permutations of vehicles provide symmetric solutions with the same route plan and objective function value, and we waste computation time when we encounter these solutions. Therefore, we add the following symmetry-breaking constraints (4.40) to the model to remove such solutions in advance.

$$y^k \geq y^{k+1}, \forall k \in \{1, \ldots, m-1\} \tag{4.40}$$

4.3.2 Decomposition Methods for the CP model

A decomposition method is a general approach for solving a complicated problem by breaking it up into easy smaller ones and solving them separately. It can be performed in parallel or sequential ways. Each variable of this new problem is related to the original variables. Benders (1962) and Dantzig & Wolfe's (1960) decomposition methods are the best-known decomposition approaches. In these approaches, after deriving the initial master problem and sub-problem, the algorithm switches between them until an optimal solution is found. The algorithm starts with the master problem computation and solves the sub-problem with the input from the main problem. If the sub-problem is not feasible, some cuts are produced to prevent the infeasibility of this solution. This cut is added to the master problem, and the process repeats (Talbi, 2016). Hooker (2005) proposed a decomposition model that combines MIP and CP for scheduling problems. El Hachemi, Gendreau, & Rousseau (2011) proposed a decomposition approach based on CP and MIP models and applied their approach to a log-truck scheduling problem. Lately, Martin, Hokama, Morabito, & Munari (2020) introduced a CP-based decomposed model for solving the constrained two-dimensional guillotine cutting problem with defects.

Due to their nature, complex decomposition structures may be needed to solve combinatorial optimization problems. However, the decomposition methods can be applied to our 3L-CVRP without complication, consisting of vehicle routing and three-dimensional loading problems. Therefore, the 3L-CVRP is decomposed into a master problem as the vehicle routing problem and loading sub-problems for each vehicle. The master problem driven by the specified objective function (4.28) involves the routing constraints, and the sub-problem tries to find a suitable loading plan for each vehicle according to the information of which customers are being served obtained from the master problem. The master model makes decisions such as how many vehicles are used, the customer subset served by each vehicle, the service start times of the customers, and the route plan of each vehicle. The sub-problem examines whether the boxes of each client subset can be loaded into the respective vehicle. If the answer is negative, the model generates a cut and ensures that the same customer subset does not reoccur in the following solution, and the cut is added to the master model as a constraint. We named this decomposition model CP-2A, and the flowchart of this model is shown in Figure 4.3.

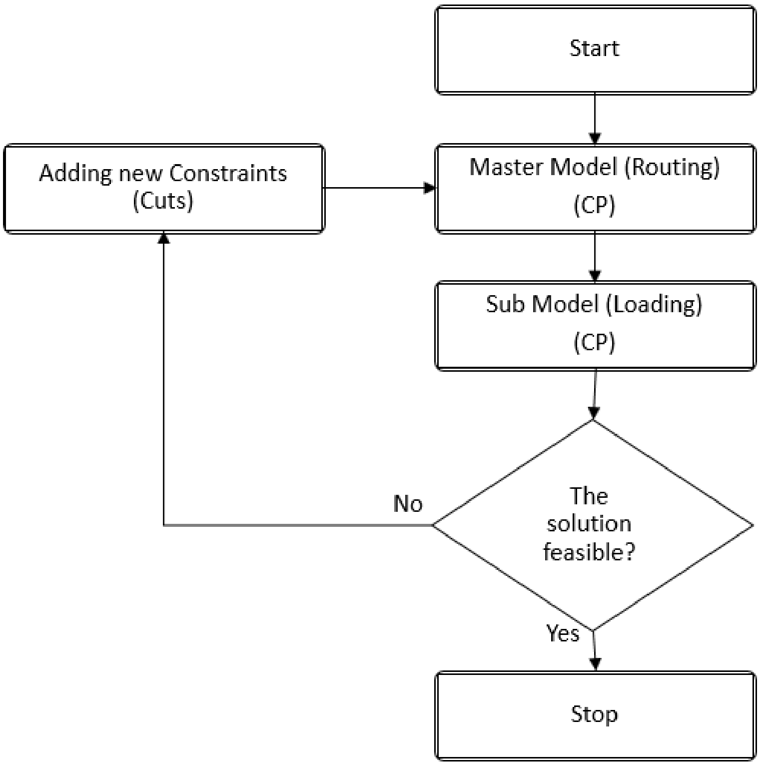


Figure 4.3 The flowchart of the CP-2A decomposition model

The objective function (4.31) and constraints (4.32)-(4.36) form the master model. Constraints (4.37)-(4.38) constitute the feasibility sub-model for the vehicles used. In the second and subsequent iterations, new constraints (cuts) must be added to the master model to avoid finding the same solution. To formulate these cuts; Let η_t^{gk} be a binary variable taking value one if box t is loaded into vehicle k in the $g^{th} \in G$ iteration, and let ∂^{gk} be a binary variable taking value one if no feasible loading plan can be found for vehicle k in the $g^{th} \in G$ iteration. The following constraint (4.41) is added to the master model formulation in the g^{th} iteration, using these variables.

$$\sum_{t \in O} (\rho_t^k \eta_t^{gk}) + \partial^{gk} \leq \sum_{t \in O} \eta_t^{gk}, \qquad \forall k \in V \tag{4.41}$$

This inequality indicates that at least one of the interval variables (boxes) found in the current solution must be changed in the next iteration solution for loading the vehicles to prevent infeasibility. The sub-problem only explores the feasibility of the loading constraints. Adding an objective function can sometimes shorten the computation time to get a feasible solution. The reason is that the CP solver focuses on finding feasible solutions when optimizing an objective function. For example, an objective function, minimizing the end edge of the boxes, can be added to the model as below.

$$Minimize\ max_{t \in O}\left(endOf(bx_t\ )\right) \tag{4.42}$$

The second version of our decomposition model, which we call CP-2B, is solved with a hierarchical approach. As mentioned earlier, the main model decides on the subsets of customers served by the vehicles. In the first step, improving solutions of the master model provided by the CP optimizer are stored in the solution pool ($\bar{S}$). In the second step, starting from the best of these solutions, it is checked with the sub-model whether it is possible to load the vehicles decided in the main model according to the sequence of their customer visits. After finding the first feasible solution, the search is stopped, and the solution is reported. The flowchart of this model is shown in Figure 4.4.

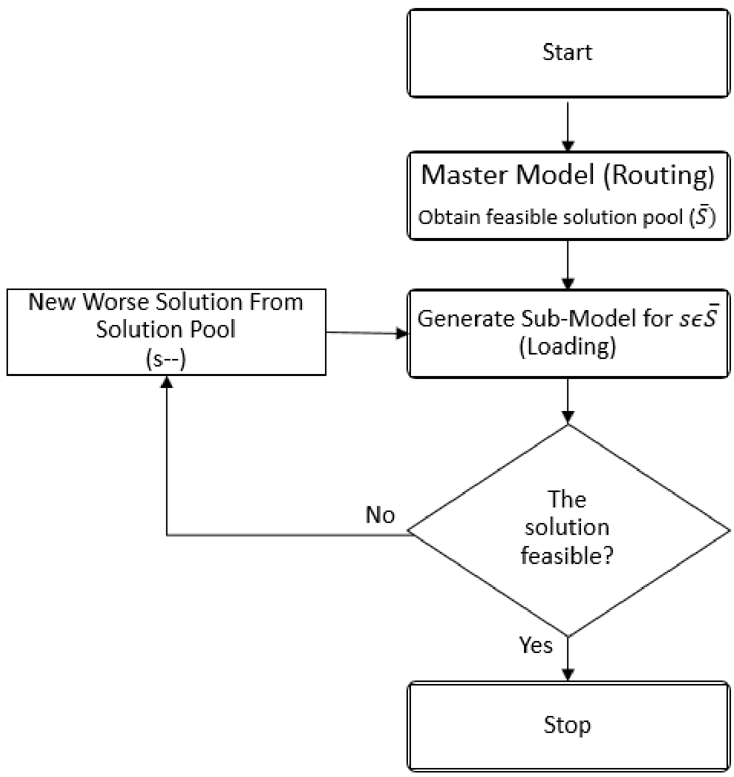


Figure 4.4 The flowchart of the CP-2B decomposition model

Let χ_{ik}^{s} be a boolean variable indicating whether the interval variable X_{ik} is present in the $s^{th}\epsilon\bar{S}$ solution. Constraint (4.37) is modified in the sub-model as follows for each solution obtained from the main model, going from the best solution to the worst. The solution is also optimal for the 3L-CVRP if the main model finds the optimal solution and the sub-model gives a suitable loading plan for each vehicle according to this solution; otherwise, the solution is not guaranteed to be optimal.

$$\chi_{ik}^{s} = \rho_{t}^{k}, \forall i \in \mathrm{N}, \forall k \in V, \forall t \in O, s \in \overline{S} \tag{4.43}$$

4.3.3 Hybrid Model (CP&EA)

This section proposes a hybrid model combining CP and a meta-heuristic evolutionary algorithm (CP&EA). This model facilitates a decomposition-based approach to solving the 3L-CVRP, as in the CP-2A model. The main difference from the CP-2A model is that it controls the feasibility of the solutions obtained from the routing part with the EA. Moreover, injecting an initial solution into the CP Optimizer engine would be useful to accelerate the search. Thus, an initial solution is created by applying a local search to a feasible solution in a number of randomly generated solution sets before running the CP model. This very simple local search procedure is similar to the one described in De Backer (2000), which minimizes the number of vehicles used. We used the $SetStartingPoint$ (sol) function to transfer some information to the CP model as a starting point that could be used as a search guide. The “sol” is a solution in the language of the CP optimizer created through the local search.

The design of population-based metaheuristic EAs is inspired by natural evolution in biological populations. Due to their structural design flexibility, EAs have recently become a diverse field of study. While numerous ways exist to determine an algorithm that operates according to the basic concepts of evolution, they all rely on the same base structure. Each has a population of search points, such as individuals, chromosomes, and candidate solutions. These search points are randomly generated and then iteratively evolved over several generations using operators and selection methods. Each EA uses unique operators inspired by mutation and recombination processes that create variety in biological evolution. The mutation operator moves around an existing search point to get new ones by sampling the neighborhood like other optimization algorithms. The recombination operator randomly searches the area between two or more existing search points to generate new ones (Corne & Lones, 2018).

Inspired by the work of Kucuk & Ermis (2010), each solution of the loading phase is a permutation of a set of boxes $(O_k = b_1, b_2, \dots b_o)$. O_k is a dynamic list of the boxes and indicates the boxes currently available in the container.

The permutation represents each box's packing order in a given solution in a one-dimensional array. Each gene value of the chromosome structure derived from the EA concept is transferred from this array. The encoding representation includes the packing order and possible orientations of the boxes (Jamrus and Chien, 2016). Figure 4.5 shows an example of encoding the packing order of three customers while each customer has a different number of boxes. According to the random permutations of the packing order, the total number of genes in the first chromosome is 15. In the second chromosome, the genes are encoded in random orientations of the boxes, as shown in Figure 4.5.

Costumers	1						2				3				
Boxes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Packing Orders	3	2	1	4	6	5	10	8	7	9	11	12	15	14	13
Orientations	1	1	3	4	1	2	6	5	1	4	1	2	3	1	2

Figure 4.5 Encoding representation of the packing and orientation chromosomes

As a construction heuristic, a corner placement variant of the deep-bottom-left-fill (DBLF) heuristic method of Karabulut & İnceoğlu (2004) is adapted to the algorithm. The DBLF heuristic ensures the stability of the cargo while each box is entirely supported by the container or other boxes with fragility control. Loading the vehicle from the rear is the standard approach in the literature. A common and practical method in transportation is that when a customer is visited, the goods belonging to the customers following each other along the route can be unloaded from the rear of the vehicle with forklifts without the need to move them (Fuellerer, Doerner, Hartl, & Iori, 2009).

EA generates a random initial population; then, the single-point crossover operator (Figure 4.6) is applied to the selected parents to generate new candidate solutions. Crossover operators are used to transfer the packing orders from parents to offspring. Then, a swapping mutation operator (Figure 4.7) is applied to the output of the crossover operator. The box types and other specifications, such as orientations, are transferred to offspring. This process continues until the specified termination condition is satisfied. The EA pseudo-code is introduced in Algorithm 4.1 below;

Costumers	1						2				3				
Boxes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Parent 1	3	2	1	4	6	5	8	7	9	10	11	15	13	14	12
Parent 2	2	3	4	6	1	5	10	9	7	8	15	14	11	13	12
Offspring 1	3	2	1	4	6	5	8	7	9	10	15	14	11	13	12
Offspring 2	2	3	4	6	1	5	10	9	8	7	11	15	13	14	12

Figure 4.6 Single-point crossover operator

Costumers	1						2				3				
Boxes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Offspring 2	2	3	**4**	6	1	**5**	10	9	8	7	11	15	13	14	12

$\Downarrow$

Offspring 2	2	3	**5**	6	1	**4**	10	9	8	7	11	15	13	14	12

Figure 4.7 Swapping mutation

Algorithm4.1 *Evolutionary algorithm generation scheme*

$P \leftarrow$ *population size*
$G \leftarrow$ *generation size*
Find the initial population $(G0)$
$it \leftarrow 1$ $\Rightarrow$ *iteration/generation number*
while *feasibility is not achieved and* $it \leq G$ **do**
  Apply the elitism procedure to the new generation
  $i \leftarrow 1$ $\Rightarrow$ *population number*
  while $i \leq P$ **do**
    Select two individuals to be parents
    Apply crossover mechanism
    Apply mutation mechanism
    Compute fitness value
  $i \leftarrow i + 1$
 end while
 $it = it + 1$
end while

4.4 Computational Study of CP-Based Solution Approaches for the 3L-CVRP

This section gives the experimental results and analysis of the solution approaches developed for the 3L-CVRP, which are explained in detail in previous sections. Experimental studies were mainly conducted in two groups; small-size and large-size benchmark instances. Small-size instances are used to evaluate the performance of MIP and CP models, which are exact methods developed for the related problem. The performance of the decomposed and hybrid models was measured with the benchmark instances commonly used in the literature.

The first part of the computational study compares the performances of MIP and integrated CP and CP-based decomposition models by solving small-size randomly generated problem instances. Also, parameter settings for the CP models are made using these instances. In the second part of the study, large-size benchmark instances from Moura and Oliveira (2009), Bortfeldt and Homberger (2013), and Gendreau et al. (2006) are used to compare the performance of the hybrid decomposition model with existing solution methods. All decomposition models were implemented with C# and executed through IBM ILOG CPLEX Optimization Studio V12.9.0, using the MIP optimizer for the MIP model and the CP Optimizer for the CP models. All problem instances were solved on an Intel Core i5-7400 processor at 3.0 GHz using 12 GB of RAM.

4.4.1 Computational Analysis for Small-Size Problem Instances

To examine the performance of MIP, CP, and CP-based decomposition models, which are exact solution approaches, and the hybrid method, we created small-size instances from Moura & Oliveira (2009). These instances consist of three classes and eight groups. Groups (G1, G2, G3) are defined from shorter planning horizons to longer, where "planning horizon" indicates an average service duration of customers. Classes (1-8) range from fewer customers to more customers. Vehicle capacities (dimensions) vary according to the number of customers. Twenty-four instances are created with 5 to 25 customers and 15 to 115 boxes. Each group has eight instances, as seen inTable 4.1.

Vehicles delivering boxes to customers have the same capacities, and the boxes are categorized into five types, as indicated in Table 4.2, and randomly distributed to customers. The dimensions given for the boxes in hod table are not width, length, and height but the maximum medium and minimum edge dimensions for each box type. The H.flag value is "1" for those allowed to be heights from these edges and "0" for those not allowed.

Table 4.1 Vehicles' dimensions for classes

Classes	Vehicles'		
	Length	Width	Height
1	180	245	300
2	240	245	300
3	240	245	300
4	240	245	300
5	360	245	300
6	360	245	300
7	480	245	300
8	540	245	300

Table 4.2 Boxes' dimensions with height flags

Box Types	Max. Dim.	H.Flag	Med. Dim.	H.Flag	Min. Dim.	H.Flag
1	108	0	76	0	30	1
2	110	0	43	1	25	1
3	92	1	81	1	55	1
4	81	0	33	1	28	1
5	120	1	99	1	73	1

CP Optimizer uses a constructive search to create a solution using various heuristics. Constructive search is a search method that tries to create solutions by fixing decision variables to values. Some parameter settings for the CP models can be useful for fast convergence to good solutions in a short computation time. We provide additional structural information to the search process by adjusting the parameters. Search phases can be defined as a simple way to select critical variables, the hierarchy between groups of variables, and values for which variables should be fixed (IBM, 2019). There are two main search phases for searching with the CP solver: the search phase on interval variables and the search phase on sequence variables. A search phase on interval variables fixes the start and end value unidirectionally and begins fixing the intervals to which a small start or end value will be assigned first. During the search phase on sequence variables, each sequence variable is assigned a fully ordered array of available interval variables, fixing the presence of interval variables in the array.

The other CP parameter that can be set is the search type parameter that can control the type of search applied to a problem. There are three types of tree search strategies in the CP Optimizer: "restart," "depth-first," and "multipoint." Choosing the right search can further improve the performance of the CP model. We also added the symmetry-breaking constraints in Equation (4.37) to increase the efficiency of the CP model. We created several scenarios to run the CP-based solution approach (CP1-A) to analyze the efficiency of symmetry-breaking constraints, different search phases, and tree search strategies.

In this study, our comparison criteria are the number of optimal solutions (nOpt.) achieved in a given time, the number of feasible solutions (nFeas.) found, and the average solution times (aTime), respectively. The average solution time is the secondary performance criterion among other performance values. After the model was run for 300 seconds, the efficiency of different parameter combinations was analyzed, considering these performance values in Table 4.3. The CP model finds an increasing number of optimal solutions when symmetry-breaking constraints are used. The best values obtained for performance criteria are shown in bold. Based on these results, we set $Searchtype = Restart$ and $searchphase = null$ and added symmetry-breaking constraints when running the CP models.

Table 4.3 Performance values obtained for different parameter settings

SearchType	**Search Phase**	**Sym.Cons.**	**nFeas.**	**nOpt.**	**aTime**
-	-	-	15	7	169
Depth-first	-	-	15	0	300
Multipoint	-	-	15	7	138
Restart	-	-	15	7	169
Restart	-	√	14	12	83
Restart	Sequence variable	-	12	7	155
Restart	Interval variable	-	12	7	142
Restart	Sequence variable	√	13	11	78
Restart	Interval variable	√	12	9	97
Multipoint	Sequence variable	√	15	8	217

Notes. aTime., average time; nOpt, number of optimum solutions; nFeas., number of feasible solutions

The MIP and CP models are set to run for 300 and 10800 sec (3h). In this way, we check these models' ability to quickly find good solutions and investigate whether the models can solve these instances optimally. The results of the first computational study are summarized in Table 4.4. The first part of Table 4.4 gives the computational results for a 300-sec run, whereas the second part gives the computational results for a 3h-run.

The results show that the decomposed CP-2A, CP-2B, and CP&EA models performed significantly better than the CP-1A and CP-1B models. The CP-2A and CP-2B models found optimal solutions for 16 out of 24 instances and at least one feasible solution for all instances. The hybrid model CP&EA found optimal solutions for 19 out of 24 instances and at least one feasible solution for all instances. These models are also not affected by the computation time extension. This shows that the CP model for the routing part rather quickly converges to the best solution. There is no relative difference between the average solution times between the CP-2A and CP-2B models. The CP&EA model has a slightly longer average computation time than the CP-2A and CP-2B models for the 300-sec time limit; however, it outperforms all other models for the 3-hr time limit. The CP&EA model has a slightly longer average computation time for the 300-sec time limit than the CP-2A and CP-2B models; however, for the 3-hr time limit, it is faster than any other model.

The CP-1B model outperforms the MIP and CP-1A models in finding feasible solutions for the 300-sec time limit. The performance of the MIP model is slightly better than CP-1A in this respect for the 3-hr time limit. On the other hand, the CP-2A model outperforms the MIP and CP-1A models in finding optimal solutions.

The results show that the CP models, particularly the CP-1A and CP-1B models, perform better in the G1 instances than in the other two groups. Therefore, it can be said that the CP models perform better in problem instances with shorter planning horizons. On the other hand, the CP&EA model shows the same performance for almost all group instances. Also, we present detailed solution results for each problem instance in both time limits inTable 4.5 and Table 4.6. In all instances, cost parameters are taken as one unit. Therefore, we give the solution results as the number of vehicles used (nv) and the total travel distance (tdist).

Table 4.4 The summarized computational results for small-size instances

	MIP			CP-1A			CP-1B			CP-2A			CP-2B			CP&EA		
Classes	**aTime**	**nFeas.**	**nOpt.**	**aTime**	**nFeas.**	**nOpt.**	**aTime**	**nFeas.**	**nOpt.**	**aTime**	**nFeas.**	**nOpt.**	**aTime**	**nFeas.**	**nOpt.**	**aTime**	**nFeas.**	**nOpt.**
G1	78.85	8	7	85.09	7	6	95.52	8	6	62.38	8	7	58.78	8	7	73.18	8	7
G2	265.49	8	2	91.22	6	5	264.36	8	1	83.19	8	6	82.49	8	6	99.02	8	6
G3	259.25	6	1	23.25	1	1	293.96	7	1	170.81	8	3	164.93	8	3	231.42	8	6
Ave./Tot.	**201.19**	**22**	**10**	**66.52**	**14**	**12**	**217.95**	**23**	**8**	**105.46**	**24**	**16**	**102.07**	**24**	**16**	**134.54**	**24**	**19**
G1	222.28	8	8	1980.91	8	7	2392.46	8	7	1470.43	8	7	1412.77	8	7	491.14	8	7
G2	8144.14	8	2	4957.12	7	4	9453.67	8	1	2729.22	8	6	2724.00	8	6	934.20	8	6
G3	6913.53	8	3	5381.03	5	3	9632.70	8	1	8739.39	8	3	8501.09	8	3	1224.12	8	6
Ave./Tot.	**5093.32**	**24**	**13**	**4106.35**	**20**	**14**	**7159.61**	**24**	**9**	**4313.01**	**24**	**16**	**4212.62**	**24**	**16**	**883.15**	**24**	**19**

Notes. **aTime.**, average time; **nOpt,** number of optimum solutions; **nFeas.,** number of feasible solutions;

Table 4.5 The detailed computational results for small-size instances (300-sec time limit)

			MIP					CP-1A					CP-1B					CP-2A					CP-2B					CP&EA				
#ins	**nodes**	**boxes**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**
G1_01	5	15	**2**	**157**	0.99	1	1	**2**	**157**	2.40	1	1	**2**	**157**	1.01	1	1	**2**	**157**	0.51	1	1	**2**	**157**	0.71	1	1	**2**	**157**	0.47	1	1
G2_01	5	20	**1**	**131**	4.50	1	1	**1**	**131**	5.30	1	1	**1**	**131**	12.70	1	1	**1**	**131**	0.66	1	1	**1**	**131**	0.71	1	1	**1**	**131**	0.91	1	1
G3_01	5	25	**1**	**151**	30.49	1	1	**1**	**151**	23.25	1	1	**1**	**151**	255.54	1	1	**1**	**151**	3.43	1	1	**1**	**151**	3.18	1	1	**1**	**151**	1.31	1	1
G1_02	6	20	**2**	**170**	1.45	1	1	**2**	**170**	3.89	1	1	**2**	**170**	1.46	1	1	**2**	**170**	0.38	1	1	**2**	**170**	0.28	1	1	**2**	**170**	0.45	1	1
G2_02	6	30	**1**	**140**	222.97	1	1	**1**	**140**	16.17	1	1	2	135	t.l.	t.l.	0	**1**	**140**	2.12	1	1	**1**	**140**	2.08	1	1	**1**	**140**	0.86	1	1
G3_02	6	35	2	157	t.l.	1	0	-	-	-	-	-	2	157	t.l.	t.l.	0	**1**	**158**	71.67	1	1	**1**	**158**	64.48	1	1	**1**	**158**	122.43	1	1
G1_03	7	25	**2**	**198**	4.37	1	1	**2**	**198**	6.88	1	1	**2**	**198**	2.60	1	1	**2**	**198**	0.41	1	1	**2**	**198**	0.51	1	1	**2**	**198**	0.98	1	1
G2_03	7	35	2	156	t.l.	1	0	**1**	**150**	48.64	1	1	2	150	t.l.	t.l.	0	**1**	**150**	6.45	1	1	**1**	**150**	6.73	1	1	**1**	**150**	2.28	1	1
G3_03	7	40	3	220	t.l.	1	0	-	-	-	-	0	3	189	t.l.	t.l.	0	2	189	100.59	1	0	3	189	100.59	1	0	**1**	**190**	233.01	1	1
G1_04	8	30	**3**	**218**	4.80	1	1	**3**	**218**	33.80	1	1	**3**	**218**	3.73	1	1	**3**	**218**	0.49	1	1	**3**	**218**	0.52	1	1	**3**	**218**	0.52	1	1
G2_04	9	35	3	210	t.l.	1	0	**3**	**201**	118.32	1	1	3	201	t.l.	1	0	**3**	**201**	32.14	1	1	**3**	**201**	28.83	1	1	**3**	**201**	130.29	1	1
G3_04	9	40	2	257	t.l.	1	0	-	-	-	-	-	-	-	-	0	0	2	224	128.30	1	0	2	243	102.12	1	0	**1**	**224**	233.82	1	1
G1_05	10	30	**4**	**269**	9.80	1	1	**4**	**269**	16.13	1	1	**4**	**269**	12.26	1	1	**4**	**269**	3.67	1	1	**4**	**269**	3.15	1	1	**4**	**269**	3.31	1	1
G2_05	10	35	3	256	t.l.	1	0	**3**	**229**	58.67	1	1	3	229	t.l.	1	0	**3**	**229**	13.86	1	1	**3**	**229**	12.01	1	1	**3**	**229**	30.22	1	1
G3_05	10	40	2	249	t.l.	1	0	-	-	-	-	-	2	249	t.l.	1	0	**1**	**254**	7.51	1	1	**1**	**254**	6.24	1	1	**1**	**254**	1.55	1	1
G1_06	15	55	**5**	**383**	41.68	1	1	**5**	**383**	232.08	1	1	**5**	**383**	142.53	1	1	**5**	**383**	14.03	1	1	**5**	**383**	11.27	1	1	**5**	**383**	45.36	1	1
G2_06	15	65	12	580	t.l.	1	0	14	558	t.l.	1	0	4	342	t.l.	1	0	**3**	**332**	7.91	1	1	**3**	**332**	7.01	1	1	**3**	**332**	20.20	1	1
G3_06	15	75	7	552	t.l.	1	0	-	-	-	-	-	8	544	t.l.	1	0	2	331	t.l.	1	0	2	363	t.l.	1	0	**2**	**331**	t.l.	1	1
G1_07	20	80	**6**	**512**	t.l.	1	1	15	857	t.l.	1	0	6	512	t.l.	1	0	**6**	**512**	178.46	1	1	**6**	**512**	152.62	1	1	**6**	**512**	230.34	1	1
G2_07	20	90	15	828	t.l.	1	0	-	-	-	-	-	6	433	t.l.	1	0	6	433	t.l.	1	0	6	433	t.l.	1	0	6	433	t.l.	1	0
G3_07	20	100	-	-	-	-	0	-	-	-	-	-	10	696	t.l.	1	0	2	414	t.l.	1	0	2	414	t.l.	1	0	2	408	t.l.	1	0
G1_08	25	105	13	1075	t.l.	1	0	-	-	-	-	-	13	716	t.l.	1	0	8	616	t.l.	1	0	8	616	t.l.	1	0	8	616	t.l.	1	0
G2_08	25	110	12	1078	t.l.	1	0	-	-	-	-	-	15	899	t.l.	1	0	7	548	t.l.	1	0	7	548	t.l.	1	0	7	547	t.l.	1	0
G3_08	25	115	-	-	-	-	0	-	-	-	-	-	14	961	t.l.	1	0	2	525	t.l.	1	0	2	525	t.l.	1	0	2	523	t.l.	1	0
Ave.	**12.08**	**52.08**	**-**	**-**	**-**			**-**	**-**	**-**			**4.91**	**351.7**	**-**			**2.92**	**288.88**	**-**			**2.96**	**291.00**	**-**			**2.83**	**288.54**	**-**		

Notes. **nv.**, number of vehicles**, t.l.**, time limit, **tdist**., total distance, **feas.**, if solution feasible 1, otherwise 0, **opt.**, if solution optimal 1, otherwise 0. The optimum solutions are given in bold.

Table 4.6 The detailed computational results for small-size instances (3-hr time limit)

					MIP					**CP-1A**					**CP-1B**					**CP-2A**					**CP-2B**					**CP&EA**		
#ins	**nodes**	**boxes**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**	**nv**	**tdist.**	**time**	**feas.**	**opt.**
G1_01	5	15	**2**	**157**	1.128	1	1	**2**	**157**	1.937	1	1	**2**	**157**	2.898	1	1	**2**	**157**	0.855	1	1	**2**	**157**	0.843	1	1	**2**	**157**	0.469	1	1
G2_01	5	20	**1**	**131**	5.087	1	1	**1**	**131**	4.301	1	1	**1**	**131**	19.26	1	1	**1**	**131**	1.77	1	1	**1**	**131**	1.767	1	1	**1**	**131**	0.906	1	1
G3_01	5	25	**1**	**151**	35.39	1	1	**1**	**151**	11.81	1	1	**1**	**151**	1453	1	1	**1**	**151**	9.696	1	1	**1**	**151**	9.534	1	1	**1**	**151**	1.312	1	1
G1_02	6	20	**2**	**170**	1.745	1	1	**2**	**170**	3.925	1	1	**2**	**170**	5.259	1	1	**2**	**170**	0.888	1	1	**2**	**170**	0.825	1	1	**2**	**170**	0.453	1	1
G2_02	6	30	**1**	**140**	254.7	1	1	**1**	**140**	13.81	1	1	1	140	t.l.	1	0	**1**	**140**	6.228	1	1	**1**	**140**	6.534	1	1	**1**	**140**	0.859	1	1
G3_02	6	35	**2**	**157**	10.93	1	1	**1**	**158**	5148	1	1	2	157	t.l.	1	0	**1**	**158**	189.6	1	1	**1**	**158**	211	1	1	**1**	**158**	122.4	1	1
G1_03	7	25	**2**	**198**	3.939	1	1	**2**	**198**	7.912	1	1	**2**	**198**	9.615	1	1	**2**	**198**	1.095	1	1	**2**	**198**	1.134	1	1	**2**	**198**	0.978	1	1
G2_03	7	35	2	156	t.l.	1	0	**1**	**150**	36.72	1	1	2	156	t.l.	1	0	**1**	**150**	18.34	1	1	**1**	**150**	20.25	1	1	**1**	**150**	2.275	1	1
G3_03	7	40	2	189	t.l.	1	0	2	230	t.l.	1	0	2	190	t.l.	1	0	2	189	t.l.	1	0	3	189	t.l.	1	0	**1**	**190**	233	1	1
G1_04	8	30	**3**	**218**	8.926	1	1	**3**	**218**	15.74	1	1	**3**	**218**	14.03	1	1	**3**	**218**	1.611	1	1	**3**	**218**	1.638	1	1	**3**	**218**	0.516	1	1
G2_04	9	35	3	201	t.l.	1	0	**3**	**201**	2244	1	1	3	201	t.l.	1	0	**3**	**201**	96.58	1	1	**3**	**201**	95.28	1	1	**3**	**201**	130.3	1	1
G3_04	9	40	2	224	t.l.	1	0	2	224	t.l.	1	0	9	322	t.l.	1	0	2	224	t.l.	1	0	2	243	t.l.	1	0	**1**	**224**	233.8	1	1
G1_05	10	30	**4**	**269**	18.14	1	1	**4**	**269**	28.31	1	1	**4**	**269**	49.02	1	1	**4**	**269**	9.966	1	1	**4**	**269**	9.402	1	1	**4**	**269**	3.313	1	1
G2_05	10	35	3	233	t.l.	1	0	7	299	t.l.	1	0	3	233	t.l.	1	0	**3**	**229**	38.58	1	1	**3**	**229**	37.96	1	1	**3**	**229**	30.22	1	1
G3_05	10	40	**1**	**254**	1151	1	1	**1**	**254**	145	1	1	1	254	t.l.	1	0	**1**	**254**	21.1	1	1	**1**	**254**	20.89	1	1	**1**	**254**	1.547	1	1
G1_06	15	55	**5**	**383**	52.76	1	1	**5**	**383**	263.3	1	1	**5**	**383**	592.9	1	1	**5**	**383**	36.8	1	1	**5**	**383**	35.85	1	1	**5**	**383**	45.36	1	1
G2_06	15	65	4	382	t.l.	1	0	6	466	t.l.	1	0	4	326	t.l.	1	0	**3**	**332**	22.67	1	1	**3**	**332**	22.08	1	1	**3**	**332**	20.2	1	1
G3_06	15	75	7	552	t.l.	1	0	-	-	-	-	-	3	352	t.l.	1	0	2	331	t.l.	1	0	2	336	t.l.	1	0	**2**	**331**	537.3	1	1
G1_07	20	80	**6**	**512**	298.4	1	1	**6**	**512**	4725	1	1	**6**	**512**	7665	1	1	**6**	**512**	907.2	1	1	**6**	**512**	448	1	1	**6**	**512**	230.3	1	1
G2_07	20	90	7	599	t.l.	1	0	10	729	t.l.	1	0	6	433	t.l.	1	0	6	433	t.l.	1	0	6	433	t.l.	1	0	6	433	t.l.	1	0
G3_07	20	100	5	508	t.l.	1	0	-	-	-	-	-	3	387	t.l.	1	0	2	414	t.l.	1	0	2	414	t.l.	1	0	2	408	t.l.	1	0
G1_08	25	105	**8**	**616**	1393	1	1	10	771	t.l.	1	0	8	616	t.l.	1	0	8	616	t.l.	1	0	8	616	t.l.	1	0	8	616	t.l.	1	0
G2_08	25	110	12	1078	t.l.	1	0	-	-	-	-	-	7	587	t.l.	1	0	7	548	t.l.	1	0	7	548	t.l.	1	0	7	547	t.l.	1	0
G3_08	25	115	6	630	t.l.	1	0	-	-	-	-	-	14	961	t.l.	1	0	2	525	t.l.	1	0	2	525	t.l.	1	0	2	523	t.l.	1	0
Ave.	**12.08**	**52.08**	**3.8**	**337.8**	-			-	-	-			**3**	**242**	-			**2.9**	**288.9**	-			**3**	**291**				**2.8**	**288.5**			

Notes. **nv.**, number of vehicles**, t.l.**, time limit, **tdist**., total distance, **feas.**, if solution feasible 1, otherwise 0, **opt.**, if solution optimal 1, otherwise 0. The optimum solutions are given in bold.

4.4.2 Computational Analysis for Large-Size Problem Instances

In the second part of the computational study, we examine the performance of the hybrid CP& EA model in all benchmark instances of Moura & Oliveira (2009), Bortfeldt & Homberger (2013), and Gendreau et al. (2006).

4.4.2.1 Moura & Oliveira's (2009) Benchmark Instances

In all Moura & Oliveira's (2009) instances, there are 25 customers with identical locations and weak heterogeneous demands. The instances are divided into two groups (GI and GII), each into two classes (I1 and I2). Forty-six problem instances consisting of four subgroups were created from combinations of the GI and G2 groups and the I1 and I2 classes (GI-I1, GI-I2, GII-I1, and GII-I2). There are 1050 boxes in the GI and 1550 boxes per instance in the GII. Class I1 has tight time windows with a short planning horizon, whereas Class I2 has large time windows with a long planning horizon. Class I2 means, unlike Class I1, many customers per route and few vehicles per problem (Moura, 2019).

If there are only a few types of boxes and many boxes per type, customer demand is poorly heterogeneous. A block consists of multiple boxes of the same box type in the same orientation (Bortfeldt & Homberger, 2013). In the CP&EA model, the concept of blocks is used to reduce the computation time. We set a time limit of 300 seconds for the CP part of the model, and other parameters are taken the same as the CP models mentioned earlier. The parameters for the EA part of the model are set as follows: The number of iterations is 1000, the population size is 100, the crossover rate is 0.80, and the mutation rate is 0.20, as suggested in Kucuk & Ermis (2010).

In Table 4.7, the solution results of the CP&EA model are compared with the results of Moura & Oliveira (2009), Bortfeldt & Homberger (2013), and Moura (2019). The second and third columns indicate the number of customers and the number of boxes for each group-class combination. In all instances, cost parameters are taken as one unit. Therefore, we can give the results in the next columns as the number of vehicles used (nv) and the average total travel distance (adist). The proposed CP&EA model and the work of Bortfeldt & Homberger (2013) and Moura (2019) have only been compared to Moura's (2019) work because this study always gives better or at least the same results as the other two solution methods. The % Gap column gives the relative reduction in total travel distance ($Dist\_Sol$) obtained from all solution methods compared to Moura’s (2019) solution method ($Dist\_Mo$) when the number of vehicles used is equal. The negative gap value denotes the percentage improvement compared to Moura (2019).

$$\%Gap = 100x\left(\frac{Dist\_Sol - Dist\_Mo}{Dist\_Mo}\right) \tag{4.44}$$

Table 4.7 The summarized computational results for Moura & Oliveira's (2009) instances

Sub-group	# nodes	# boxes	Moura (2019)*		Moura & Oliveira (2009)			Bortfeldt & Homberger (2013)			CP&EA		
			adist	nv	adist	nv	%Gap	adist	nv	%Gap	adist	nv	%Gap
GI-I1	25	1050	511.35	63	891.20	79	-	540.97	63	6.33	505.49	63	-0.52
GI-I2	25	1050	457.96	44	1715.44	54	-	528.44	44	15.00	463.71	44	1.38
GII-I1	25	1550	549.85	75	1251.46	91	-	599.00	75	9.13	534.23	75	-2.77
GII-I2	25	1550	521.89	66	2907.88	75	-	553.81	66	6.11	503.76	66	-3.15
		Total Gap					-			36.58			-5.00
Avg. CPU times(sec)			405.41		<60			136			<350		

When the results obtained according to the problem subgroups are examined, we see that the CP&EA model in all subgroups except the GI-I2 subgroup gives better results on average. On the other hand, it performs better than the other two methods. Problems in the GI-I2 subgroup have a larger service time window. Therefore, finding quality solutions to such problems may be possible by providing a longer computation time. Tight time windows evoke intense constraint propagation so that optimum or near-optimum solutions can be obtained within a reasonable computation time. Table 4.8 to give detailed solution results for each subgroup's problems. According to these results, the CP&EA model develops the best-known solutions in 24 of 46 examples, giving close results to the best-known solutions for the rest. There is no doubt that the proposed solution approach can obtain quality solutions for the 3L-CVRP in a reasonable computation time.

Table 4.8 The detailed computational results for the GI–I1 instances

#ins	#nodes	#boxes	Moura (2019) *		Moura & Oliveira (2009)			Bortfeldt & Homberger (2013)			CP&EA		
			tdist	nv	tdist	nv	%Gap	tdist	nv	%Gap	tdist	nv	%Gap
GI-I1-01	25	1050	623.57	8	762.59	9	-	625.03	8	0.23	625.04	8	0.24
GI-I1-02	25	1050	570.90	7	675.24	8	-	575.09	7	0.73	570.71	7	-0.03
GI-I1-03	25	1050	498.80	5	1250.86	6	-	501.73	5	0.59	457.15	5	-8.35
GI-I1-04	25	1050	462.64	4	605.72	6	-	557.50	4	20.50	462.17	4	-0.10
GI-I1-05	25	1050	543.17	6	1398.47	9	-	550.83	6	1.41	545.09	6	0.35
GI-I1-06	25	1050	503.65	5	757.08	7	-	549.13	5	9.03	468.44	5	-6.99
GI-I1-07	25	1050	446.64	5	1108.67	6	-	474.80	5	6.30	465.49	5	4.22
GI-I1-08	25	1050	575.55	4	397.19	5	-	576.52	4	0.17	467.57	4	-18.76
GI-I1-09	25	1050	469.04	5	1050.70	6	-	517.29	5	10.29	500.40	5	6.69
GI-I1-10	25	1050	474.48	5	578.36	6	-	505.46	5	6.53	476.95	5	0.52
GI-I1-11	25	1050	428.69	5	1128.55	6	-	500.33	5	16.71	477.52	5	11.39
GI-I1-12	25	1050	539.12	4	980.97	5	-	557.90	4	3.48	507.96	4	-5.78
			Average Gap				-			6.33			-0.52

Table 4.9 The detailed computational results for the GI-I2 instances

#ins	#nodes	#boxes	Moura (2019)*		Moura & Oliveira (2009)			Bortfeldt & Homberger (2013)			CP&EA		
			tdist	nv	tdist	nv	%Gap	tdist	nv	%Gap	tdist	nv	%Gap
GI-I2-01	25	1050	505.48	4	2668.55	5	-	765.83	4	51.51	541.88	4	7.20
GI-I2-02	25	1050	468.94	4	2555.26	5	-	561.11	4	19.65	503.03	4	7.27
GI-I2-03	25	1050	468.33	4	2526.11	5	-	508.65	4	8.61	459.81	4	-1.82
GI-I2-04	25	1050	423.18	4	1953.67	5	-	471.17	4	11.34	437.62	4	3.41
GI-I2-05	25	1050	448.13	4	635.96	5	-	529.95	4	18.26	465.53	4	3.88
GI-I2-06	25	1050	434.67	4	2394.25	5	-	542.60	4	24.83	451.26	4	3.82
GI-I2-07	25	1050	453.46	4	2187.27	5	-	470.82	4	3.83	436.72	4	-3.69
GI-I2-08	25	1050	433.30	4	472.35	4	9.01	465.13	4	7.35	429.28	4	-0.93
GI-I2-09	25	1050	427.27	4	674.01	5	-	442.30	4	3.52	469.31	4	9.84
GI-I2-10	25	1050	521.37	4	753.04	5	-	578.13	4	10.89	475.95	4	-8.71
GI-I2-11	25	1050	453.41	4	2049.39	5	-	477.16	4	5.24	430.46	4	-5.06
			Average Gap				-			15.00			1.38

Table 4.10 The detailed computational results for the GII–I1 instances

#ins	#nodes	#boxes	Moura (2019) *		Moura & Oliveira (2009)			Bortfeldt & Homberger (2013)			CP&EA		
			tdist	nv	tdist	nv	%Gap	tdist	nv	%Gap	tdist	nv	%Gap
GII-I1-01	25	1550	639.15	8	823.04	9	-	654.62	8	2.42	641.86	8	0.42
GII-I1-02	25	1550	581.43	7	1622.59	9	-	592.14	7	1.84	577.87	7	-0.61
GII-I1-03	25	1550	523.75	6	1451.39	7	-	548.49	6	4.72	523.64	6	0.00
GII-I1-04	25	1550	532.52	6	1221.44	7	-	540.43	6	1.49	507.40	6	-4.72
GII-I1-05	25	1550	543.12	6	1532.44	10	-	693.22	6	27.64	580.90	6	6.96
GII-I1-06	25	1550	612.54	6	1576.10	8	-	627.53	6	2.45	524.99	6	-14.29
GII-I1-07	25	1550	501.63	6	1378.36	7	-	622.50	6	24.10	507.02	6	1.07
GII-I1-08	25	1550	532.21	6	1187.52	7	-	549.18	6	3.19	501.78	6	-5.72
GII-I1-09	25	1550	572.74	6	625.91	6	9.28	709.61	6	23.90	528.23	6	-7.77
GII-I1-10	25	1550	507.34	6	1235.62	7	-	570.38	6	12.43	508.61	6	0.25
GII-I1-11	25	1550	534.75	6	1293.95	7	-	540.03	6	0.99	509.71	6	-4.68
GII-I1-12	25	1550	516.99	6	1069.11	7	-	539.84	6	4.42	495.61	6	-4.14
			Average Gap				-			9.13			-2.77

Table 4.11 The detailed computational results for the GII–I2 instances

#ins	#nodes	#boxes	Moura (2019) *		Moura & Oliveira (2009)			Bortfeldt and Homberger (2013)			CP&EA		
			tdist	nv	tdist	nv	%Gap	tdist	nv	%Gap	tdist	nv	%Gap
GII-I2-01	25	1550	529.25	6	3740.55	7	-	591.35	6	11.73	530.16	6	0.17
GII-I2-02	25	1550	552.33	6	3496.39	7	-	560.88	6	1.55	503.83	6	-8.78
GII-I2-03	25	1550	502.21	6	3134.62	7	-	525.99	6	4.74	503.99	6	0.35
GII-I2-04	25	1550	487.49	6	3814.29	6	-	516.35	6	5.92	497.60	6	2.07
GII-I2-05	25	1550	550.29	6	627.66	7	-	649.80	6	18.08	511.95	6	-6.97
GII-I2-06	25	1550	547.43	6	3115.18	7	-	560.64	6	2.41	499.61	6	-8.74
GII-I2-07	25	1550	476.92	6	2740.03	7	-	506.16	6	6.13	497.60	6	4.34
GII-I2-08	25	1550	512.87	6	2212.02	7	-	534.97	6	4.31	497.60	6	-2.98
GII-I2-09	25	1550	503.44	6	2962.35	7	-	544.63	6	8.18	499.61	6	-0.76
GII-I2-10	25	1550	587.45	6	3512.25	7	-	599.93	6	2.12	503.83	6	-14.23
GII-I2-11	25	1550	491.11	6	2631.39	6	-	501.26	6	2.07	495.61	6	0.92
			Average Gap				-			6.11			-3.15

4.4.2.2 Bortfeldt and Homberger's (2013) benchmark instances

Bortfeldt and Homberger's (2013) instances were tested to examine the proposed solution method's validity for larger problems. These 120 instances consist of six test cases, each with 20 instances derived from the CVRP instances by Solomon (1987) and the CLP instances by Bischoff and Ratcliff (1995). The first test case of instances has 100 customers and 5000 boxes. While some instances are generated with wide depot time windows, others have narrow depot time windows. Table 4.12 shows the results of BH1-20 instances yielded by the P1R2 algorithm of Bortfeldt & Homberger (2013), the model-based heuristic of Moura (2019), and the CP&EA model. The number of vehicles and the total distance are given per instance, with the best in bold font. CP&EA achieved the best results for 3 of the 20 instances; the other results are the same or slightly worse. In most cases, CP&EA outperforms the P1R2 algorithm in terms of the results, surpassing the model-based heuristic in computational time.

A box should not be placed over a box requested by another customer subsequently visited to avoid the reloading cost. Bortfeldt & Homberger (2013) and Moura (2019) neglected this constraint in their studies, known as the LIFO rule. We have also solved these instances with the LIFO constraint and presented the results in Table 4.12. The results show a 12% increase in the number of vehicles used and a 10% increase in total distance traveled with the LIFO constraint.

Table 4.12 The detailed computational results for the BH1-20 instances

#ins	#nodes	#boxes	Bortfeldt & Homberger (2013)			Moura (2019)			CP&EA			CP&EA (LIFO)		
			tdist	nv	time(s)	tdist	nv	time(s)	tdist	nv	time(s)	tdist	nv	time(s)
BH1	100	5000	1711.56	21	971	1660.82	20	2942	1773.56	20	690	1850.95	22	979
BH2	100	5000	1516.92	17	975	1487.23	17	4006	1530.60	17	740	1742.01	17	1172
BH3	100	5000	1229.31	14	969	1293.75	14	4976	1281.62	14	660	1437.50	14	1091
BH4	100	5000	994.44	11	972	1024.65	10	2945	1048.90	10	509	1096.53	11	700
BH5	100	5000	1895.41	16	860	1472.11	16	2606	1543.66	16	886	1655.40	18	1150
BH6	100	5000	1749.37	19	992	1899.31	20	3006	1806.08	19	841	1859.03	21	1332
BH7	100	5000	1523.94	17	1015	1487.35	17	5176	1566.84	17	678	1632.97	18	1201
BH8	100	5000	1309.23	14	765	1297.34	14	3913	1359.79	14	974	1394.26	16	1193
BH9	100	5000	1227.09	11	799	1117.31	10	1998	1085.38	11	943	1149.99	13	1592
BH10	100	5000	1584.00	15	762	1378.65	14	2905	1573.48	15	894	1629.95	17	1781
BH11	100	5000	1622.99	20	731	1745.96	20	1828	1750.29	19	969	1839.93	21	1306
BH12	100	5000	1269.55	15	738	1291.7	15	1945	1386.95	15	1118	1534.36	16	1751
BH13	100	5000	1060.68	12	730	1023.58	11	2925	1089.57	11	1043	1423.52	12	2422
BH14	100	5000	911.18	10	742	984.33	10	1455	939.99	8	1464	1011.44	10	2304
BH15	100	5000	1469.08	16	735	1298.1	15	1838	1421.99	16	1042	1506.02	18	1325
BH16	100	5000	1518.19	18	751	1644.71	18	2578	1652.54	17	1169	2241.12	19	1876
BH17	100	5000	1266.03	15	776	1147.01	14	1940	1239.34	14	1052	1447.57	17	2133
BH18	100	5000	1177.58	14	778	1131.47	12	1545	1239.16	13	1127	1365.80	16	2507
BH19	100	5000	1023.38	11	776	968.59	10	1940	981.07	11	1274	1138.24	13	2218
BH20	100	5000	1221.35	15	765	1097.33	14	1713	1344.99	14	1191	1616.54	16	2470
		Total	**27281.28**	**301**		**26451.3**	**291**		**27615.27**	**291**		**30573.13**	**325**	

4.4.2.3 Gendreau et al.'s (2006) Benchmark Instances

Finally, we used Gendreau et al.'s (2006) benchmark instances to examine the CP&EA model's performance for large-size problems without time windows. They consist of 27 problem instances, with the number of customers varying between 15 and 100. The nodes' coordinates, the weights demanded by the customers, and the vehicle weight capacities were taken from the CVRP instances of Toth and Vigo (2002). The loading space of the vehicles and widths, lengths, and heights of requested items were generated by Gendreau et al. (2006). Table 4.13 presents the comparison of the performance of the CP&EA for the least constrained variant of the 3L-CVRP problems with the algorithms of Gendreau et al. (2006), Bortfeldt (2012), Lacomme et al. (2013), and Zhang et al. (2015). CP techniques are usually used for tightly constrained problems such as timetabling and scheduling problems (Talbi, 2009). However, the efficiency of CP&EA shows that it is a very competitive approach for this type of routing problem as well. CP&EA achieved the best results for 10 of the 27 instances in a reasonable time, with the remaining results slightly worse.

Table 4.13 The detailed computational results for the Gendreau et al.'s (2006) instances

#ins	#nodes	#boxes	#vehicles	Gendreau et al. (2006)		Bortfeldt (2012)		Lacomme et al. (2013)		Zhang et al. (2015)		CP&EA	
				distance	time(s)	distance	time(s)	distance	time(s)	distance	time(s)	distance	time(s)
3l_cvrp01	15	32	5	297.65	3	297.65	11	297.65	4	297.65	0.30	**282.95**	379
3l_cvrp02	15	26	5	334.96	1	334.96	0	335.67	0	**334.96**	0.10	**334.96**	308
3l_cvrp03	20	37	5	362.27	448	381.36	80	362.27	14	**362.27**	1.50	**362.27**	376
3l_cvrp04	20	36	6	430.89	11	430.89	1	430.89	0	430.89	0.10	**430.88**	311
3l_cvrp05	21	45	6	395.64	1	397.16	182	**379.43**	8	395.96	2.70	389.87	427
3l_cvrp06	21	40	6	495.85	15	498.07	3	495.85	0	**495.85**	0.20	**495.85**	311
3l_cvrp07	22	46	6	742.23	2	741.80	96	**725.43**	237	731.81	1.70	738.45	405
3l_cvrp08	22	43	8	735.14	105	735.14	98	735.14	37	730.66	10.10	**708.32**	437
3l_cvrp09	25	50	8	630.13	978	631.82	3	630.13	2	630.13	1.60	**625.10**	402
3l_cvrp10	29	62	10	717.90	411	739.94	161	**687.57**	589	707.10	218.30	689.97	433
3l_cvrp11	29	58	9	718.24	208	723.44	105	718.24	1453	718.25	3.30	**695.71**	423
3l_cvrp12	30	63	9	614.60	1303	623.10	5	610.05	20	**610.00**	4.20	610.23	334
3l_cvrp13	32	61	9	2316.56	2317	2348.48	315	**2306.04**	1242	2309.75	121.90	2433.45	436
3l_cvrp14	32	72	11	1276.60	2121	1234.54	313	**1186.96**	2424	**1186.05**	427.50	1231.32	412
3l_cvrp15	32	68	10	1196.55	2916	1202.34	315	**1161.20**	2145	**1155.60**	656.90	1177.54	425
3l_cvrp16	35	63	11	698.61	863	704.47	1	698.61	3	**698.61**	3.80	**698.61**	312
3l_cvrp17	40	79	14	906.42	753	928.93	1	861.80	9	**861.79**	11.20	**861.79**	312
3l_cvrp18	44	94	14	1124.33	2199	1108.37	295	**1084.26**	1894	1096.54	1180.90	1093.21	434
3l_cvrp19	50	99	13	680.29	1390	678.59	414	670.44	3323	**660.28**	1216.00	700.21	447
3l_cvrp20	71	147	20	529.00	7008	520.55	418	510.95	2893	**505.33**	2574.50	566.70	580
3l_cvrp21	75	155	18	1004.40	6263	964.66	437	943.05	4174	**928.93**	2402.90	1040.67	698
3l_cvrp22	75	146	19	1068.96	2079	1041.92	416	1029.87	3562	**1004.60**	2184.50	1090.96	721
3l_cvrp23	75	150	18	1012.51	4314	995.22	418	987.06	3121	**960.44**	1353.70	1035.70	758
3l_cvrp24	75	143	18	1063.61	1053	1053.41	296	1056.33	2610	**1035.80**	1228.90	1083.37	781
3l_cvrp25	100	193	24	1371.32	501	1238.83	433	1232.73	4489	**1175.95**	3256.10	1404.60	908
3l_cvrp26	100	199	28	1557.12	1075	1444.58	416	1415.15	3485	**1363.80**	2573.50	1430.18	1247
3l_cvrp27	100	198	25	1378.52	3983	1342.23	423	1317.38	3373	**1287.72**	2610.10	1484.88	1085
			Average	**876.31**	**1567.36**	**864.54**	**209.36**	**847.04**	**1522.56**	**839.88**	**816.54**	**877.69**	**522.34**

4.5 Chapter Summary

This chapter discussed the 3L-CVRP, which integrates the three-dimensional loading problem with the capacitated vehicle routing problem. To solve this NP-hard problem, we formulated all constraints and objective functions in the context of constraint programming and developed five models: two integrated, two decomposed CP-based, and one decomposed hybrid model. The hybrid model (CP&EA) uses a CP model for routing and an evolutionary algorithm for loading. First, we compared these models and a MIP model adapted from the literature by solving 24 small-size problems. Based on the comparison results, the hybrid model (CP&EA) gave the highest performance. Then, a computational study was conducted with Moura & Oliveira's (2009), Bortfeldt and Homberger's (2013), and Gendreau et al. (2006)'s benchmark instances to test the hybrid model's performance in large-size problems. We improved the solution values for 36 of 93 problems, and the CP&EA model performed better than the existing approaches for Moura & Oliveira's instances on average. For the rest of the problems, we achieved very competitive results. According to the computational results, thus, it has been shown that the proposed CP&EA model can help solve the problems in distribution logistics.

CHAPTER FIVE
NEW EVOLUTIONARY ALGORITHM FOR THE 3L-CVRP

5.1 Introduction

This chapter deals with the capacity vehicle routing problem with three-dimensional loading constraints (3L-CVRP), representing a combination of vehicle routing and three-dimensional bin packing problems introduced by Gendreau et al. (2006). As explained in the previous chapters, it combines the traditional vehicle routing problem (CVRP) with the additional constraint of loading three-dimensional boxes into the vehicles in a way that maximizes capacity utilization while adhering to specific loading constraints. The 3L-CVRP is a challenging optimization problem that arises from integrating vehicle routing and three-dimensional loading of goods. Due to its NP-hard nature, solving it optimally for large instances is computationally intractable, but heuristic and metaheuristic methods offer viable approaches to finding good solutions in practice.

A new Evolutionary Algorithm (EA) is proposed for the 3L-CVRP. Initially, a description of the EA is given. Following, the details of the proposed solution method are provided. Finally, a computational study is given.

5.2 A Novel Elitism-Based Evolutionary Algorithm for the 3L-CVRP

Metaheuristic algorithms are search processes that aim to generate reasonable solutions for optimization problems that are intricate and demanding to solve for optimality. They usually strive to attain a quasi-optimal solution for inadequate resources founded on partial or imperfect information. The emergence of metaheuristics is considered a paramount accomplishment in operations research in recent years.

Metaheuristics aim to develop strategies that direct the search for optimal or desired values from all available options of a system's parameters. This approach implements stochastic optimization, which frequently utilizes random variables generated from metaphors and natural world inspirations, resulting in solutions that are often good enough to maximize or minimize the output. Drawing upon natural behaviours as a foundation to craft novel optimization algorithms has greatly impacted the history of the metaheuristic domain. (Sörensen et al. 2017). For instance, EAs are a type of metaheuristic techniques that draw inspiration from the natural evolution process. They are designed to handle intricate optimization problems that typically entail non-convex and extremely non-linear resolution spaces. Conventional mathematical programming methods struggle to configure a solution due to their complex computational requirements (Talbi, 2013). Many classical EAs have been extensively studied due to the increasing complexity of real-world optimization problems, including Evolutionary Programming (EP), Genetic Algorithms (GA), Evolution Strategies (ES), and Genetic Programming (GP).

Moreover, novel EAs such as Ant Colony Optimization Algorithm (ACO), Firefly Algorithm (FA), and Harmony Search Algorithm (HS) have gained attention in recent years. No matter how many methods related to Evolutionary Algorithms emerge, the fundamental concept behind all these techniques remains the same: when a group of individuals inhabit an environment where resources are limited, competing for these resources' triggers natural selection, leading to improved fitness of the population with each subsequent iteration (Eiben & Smith, 2003).

The chapter introduces a method that proposes the generation of a random set of candidate solutions related to a sine wave by adjusting a quality function to the EA, similar to the classical approaches mentioned. The sine wave is a geometric waveform that undergoes periodic oscillations, alternating up and down, and is defined by the function $y = sin\,x$. In essence, it portrays a smooth, s-shaped wave that fluctuates above and below zero.

Joseph Fourier, a French mathematician and physicist, demonstrated that sine waves can function as fundamental basic units to depict and approximate any recurrent waveform (North & Helliwell, 2013). For instance, by breaking down any recurrent activity into sinusoidal wave components, it became less challenging to formulate heat transfer models that depict how heat spreads in a specified area. Consequentially, the Heat equation evolved into a critical constituent of several scientific fields of applied mathematics, such as probability theory. A sound wave can be characterized by the amplitudes of the sine waves that compose it. This series of numbers produces the harmonic content of the sound, termed the harmonic spectrum. Therefore, a sine or sinusoidal wave presents the most realistic portrayal of natural periodic state changes. Elitism is a method of guaranteeing that the most exceptional members of an Evolutionary Algorithm are retained in the succeeding generations, with its proportion controlled via parameters. The sine wave in this research demonstrates how the amplitude of the elitism rate varies over time. A high rate of elitism implies that genes from superior individuals are passed down to new generations, whilst a low rate implies that inherited individuals are less likely to excel but offer a diverse range of traits. When transferring traits of a particular group of individuals across generations in an EA population, it entails performing a local search within the solution set that group belongs to. However, when the rate of elitism is low, randomly selecting individuals for the population's continuity may produce new individuals with significantly different characteristics from the existing ones. Therefore, the outcome would be the same as a random search for the solution set.

5.2.1 (μ, λ)-ES

EA is a subfield of computational intelligence that encompasses ideas, approaches, methods, and tools for creating intelligent systems capable of exhibiting sophisticated behavior in intricate settings. Its applications are found in problem-solving systems that leverage computational models with evolutionary dynamics as the key component of the design. EA aims to capitalize on the collective dynamics of adaptive populations that enable birth, death, variation and selection to occur in an iterative intergenerational cycle.

EAs developed independently after the second half of the 1900s have four prominent sub-components (see Figure 5.1): *Genetic algorithms (GA)* are a form of evolutionary algorithm which mimic the process of natural selection, developed by Holland (1975). They evolve populations of candidate solutions over generations by using genetic operators including selection, crossover, and mutation. *Evolutionary Strategies (ES),* which were introduced by Rechenberg (1971) and Schwefel (1975) in the early 1970s, aim to optimize real-valued functions. ES utilize mutation operations as the primary means to explore the solution space. *Evolutionary Programming (EP),* first introduced by Fogel, Owens & Walsh (1965), is one of the earliest techniques in evolutionary computation. EP primarily targets numerical optimization problems and relies significantly on mutations and self-adapting algorithms. It is paradoxically well-suited for continuous function optimization, which distinguishes it from other evolutionary computation techniques. *Genetic Programming (GP),* developed by Koza (1992), extends the principles of genetic algorithms to develop computer programs or symbolic expressions. In GP, program structures represent the "genes," and evolution improves the structure and behaviour of these programs. Although each solution method is different, they are all inspired by the same principles of natural evolution.

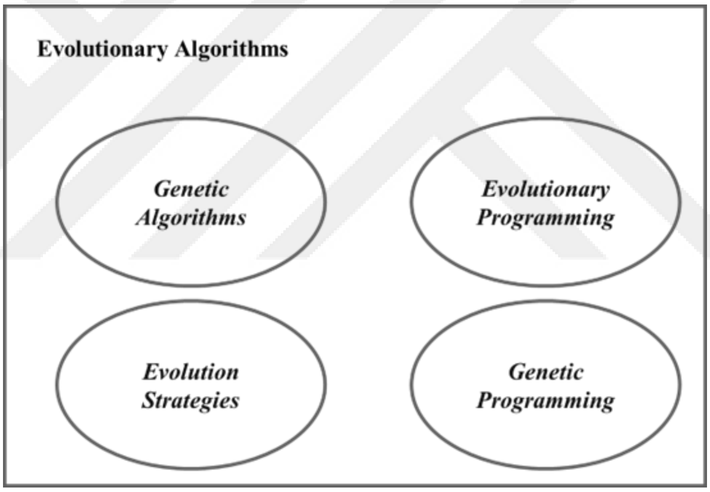


Figure 5.1 Evolutionary algorithms

Evolution Strategies (ES) are primarily used in this section as the main method for achieving solutions. Initially, it was applied to optimization processes that involve continual variables that are real-valued. Specific mutation operators that involve normally distributed variables are commonly utilized, while crossover is not frequently employed. There are only two principal principles for ES: Firstly, all the variables are slightly and randomly modified simultaneously. If the new set of variables enhances the function's performance, retain it; otherwise, revert to the previous state. The initial rule mirrors natural mutations, and the second reflects the principle of natural selection, which accurately models "survival of the fittest" (Darwin, 1859).

The most basic form of ES is referred to as $(1 + 1) - ES$, which involves a solitary parent and one offspring. Two additional variations of ES exist, namely $(\mu + \lambda) - ES$ and (μ,λ)-ES. In the $(\mu + \lambda) - ES$ variation, $\lambda \geq 1$ offspring is generated per generation, and the poorest λ of all $\mu + \lambda$ individuals is eliminated to maintain population size stability (Beyer & Schwefel, 2002). In $(\mu, \lambda) - ES$, there is a disparity between the population of μ-size parents and the population of λ-size offspring (where λ is greater than or equal to μ). Some more recent versions of ES use the additional operator ρ to describe both the parent population μ and the $(\mu/\rho, \lambda) - ES$, where the new generation is chosen from the best λ offspring and the population size remains constant. Nonetheless, regardless of the parents' fitness, they cannot be included in the next generation (see Figure 5.2).

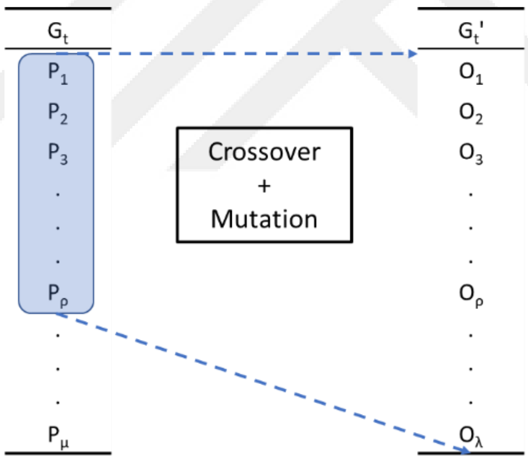


Figure 5.2 Representation of (μ/ρ, λ)-ES

The basic algorithm of a $(\mu, \lambda) - ES$ is summarized in Algorithm 5.1. The method begins by generating a population of μ parents through random sampling at intervals achievable for objective variables. As the first generation, the population's fitness values are established. Subsequently, generations proceed to create offspring guided by the search strategy, with their fitness values being determined and the best offspring being added to the population until the stopping criterion is met. ES implements selfadaptation by improving the strategy parameters of the selection and replacement operators while enhancing the solution. As such, the $(\mu, \lambda) - ES$ approach was selected to implement our novel optimization method due to its high customizability.

Algorithm 5.1 *(μ, λ)-ES Template (Talbi, 2009)*

Assign a randomly generated population containing μ individuals
Calculate the fitness of each individual in the population
While *not termination criterion is met,* **do**
    Select μ/ρ parents from the population
    Generate λ offspring from μ/ρ parents
    Calculate the fitness of λ offspring
    Replace the population with μ individuals from the best of offspring and parents
End while
Provide the best performing member of the population.

Overall, ES is a robust and flexible optimization algorithm that has found successful applications in various fields, including engineering design, machine learning, and parameter optimization in complex systems. Like any optimization method, choosing appropriate algorithm parameters and fine-tuning the settings for specific problems is essential to achieve good performance.

For example, the ES algorithm described above cannot be directly applied to combinatorial optimization problems since the discreteness of the solutions of combinatorial optimization problems is not taken into account. Therefore, the components of ES need to be modified for these types of problems. Components such as the selection mechanism, the representation of individuals, and the local search performed through secondary heuristics in the solution space can be given as examples of those to be modified.

The VRPTW has been successfully solved using ES developed by Homberger & Gehring (1999) for combinatorial optimization problems for the first time. Figure 5.3 shows the ES-based reproduction process of their algorithm. Population with μ individuals and λ offspring is produced starting from $P(t)$. To create the offspring, several individuals, called parents, are selected from $P(t)$. The selections are random, and the probability of selection is the same for each individual and is independent of the individuals' $F^*(X)$ fitness values. Exactly one offspring is produced by recombining the selected parents and then being subjected to a mutation. Then μ individuals of the next population $P(t+1)$ are selected from the set of offspring λ according to their $F^*(X)$ fitness values. Since parents are not involved in the selection process, disruptions are allowed during the search. Therefore, the search may escape the local optimum.

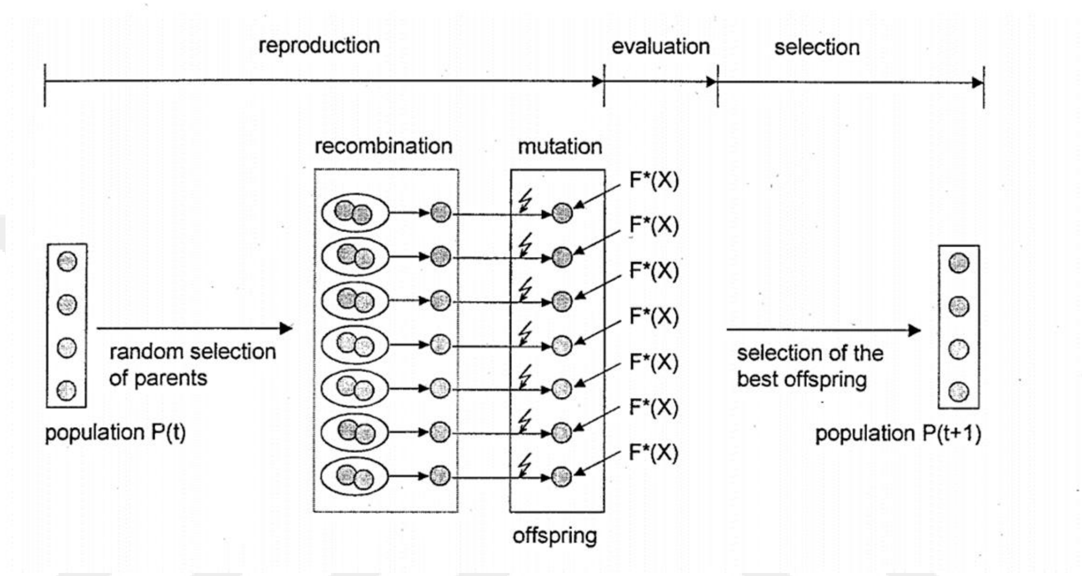


Figure 5.3 The ES-based reproduction process (Homberger & Gehring, 1999)

5.2.2 Elitism

In this chapter, the search strategy is based on the concept of elitism. Sociologically, the term "elite" typically denotes individuals or groups belonging to the highest echelons of society, although the precise criteria for what constitutes the elite may vary. Sociologically, the term "elite" typically denotes individuals or groups belonging to the highest echelons of society, although the precise criteria for what constitutes the elite may vary. Moreover, these criteria may evolve over time or in response to cultural factors. Although wealth may play a role in determining social class, personal achievement can be even more significant in certain cultures. Within the context of evolutionary algorithms, elitism involves preserving the "top" solutions obtained during the search process. This guarantees that the most exceptional individuals are consistently present in the population, generation after generation.

The application of elitism, initially suggested by De Jong (1975), frequently enhances the performance of EA. There are several methods for implementing it, but two commonly used options, as described by Simon (2013), are as follows: firstly, elitism can be achieved by producing only $(\lambda = \mu - \varepsilon)$ offspring per generation(G_t), where μ is the population size and ε is the number of elite individuals defined by the user. When striving to keep the top ε members of the μ-sized population for the next generation, through combining and changing genes to create progeny, the best ε members conjoin with the $(\lambda = \mu - \varepsilon)$ offspring to create the subsequent generation (G_{t+1}) of μ individuals (see Figure 5.4).

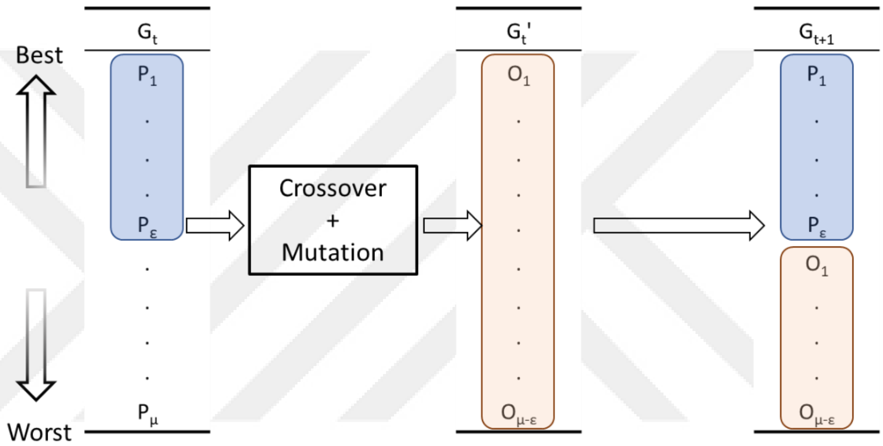


Figure 5.4 Elitism option 1

The second way to implement elitism is to produce $(\lambda = \mu)$ offspring and replace the worst offspring with the best ε individuals of the previous generation (G_t). Figure 5.5 shows this option. Reasonable outcomes are generally expected by means of these options; nevertheless, extra enhancement steps may be required to obtain plausible solutions for multi-modal optimization problems.

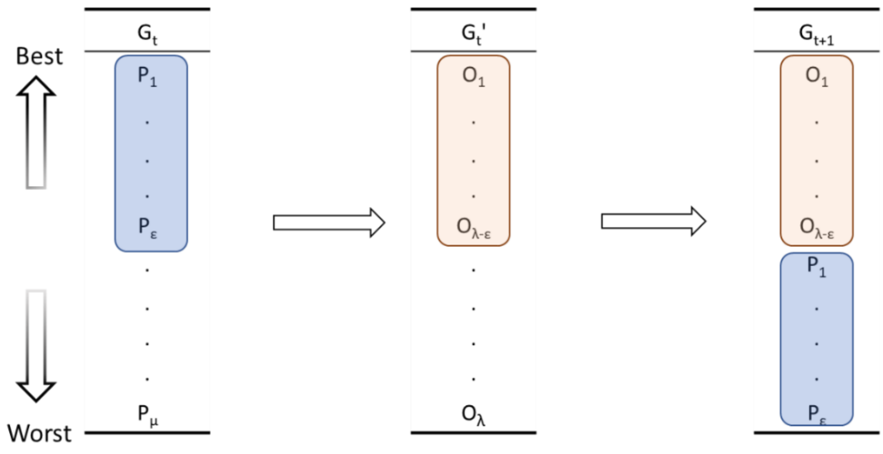


Figure 5.5 Elitism option 2

In this dissertation, an alternative approach to elitism is presented. The proposal balance on two primary components: diversification and intensification, also commonly known as exploration and exploitation. Despite appearing contradictory, the synergistic integration of these elements is pivotal to achieving optimal performance in any meta-heuristic algorithm. Especially in multi-modal optimisation problems, proper diversification enables the search in the solution space to efficiently explore as many places and regions as possible. Moreover, it ensures that the search algorithm does not become trapped in a local optimum. In practice, diversification is often depicted through randomization and/or an additional stochastic component superimposed on deterministic components. If diversification is excessively strong, it may randomly discover an excessive number of locations and impede the algorithm's convergence. If diversification is inadequate, the searched solution space may be limited, leading to a local optimum and possibly unfeasible outcomes (Yang, 2009).

On the other hand, proper intensification or exploitation seeks to capitalize on the past events and knowledge gained throughout search process. When necessary, limiting diversification and reducing the search region aims to accelerate convergence. The result of this is that intensification is often accomplished using narrative memory, as in Taboo Search (TS) or elitism. However, strong intensification may cause early convergence, resulting in local optima as the search area is not well explored. In a given set of problems, fine-tuning parameters or self-adaptive mechanisms are often needed to increase the efficiency of an algorithm. Therefore, an optimal balance between diversification and intensification is required, and such a balance is itself often an optimization process.

Metaheuristic optimization algorithms are general-purpose optimization techniques designed to tackle a wide range of problems. However, these algorithms do make implicit assumptions about the types of problem functions on which they are expected to perform well. Making these assumptions explicit can lead to more effective and efficient optimization solutions. In this context, the "No Free Lunch" (NFL) theory has emerged as a significant paradigm in academic literature. It was introduced by Wolpert & Macready (1995) to assess the effectiveness of optimization methods. The NFL theorem emphasizes the significance of comprehending the problem landscape and altering optimization algorithms to utilize certain problem structures or heuristics. It serves as a reminder that no single "best" optimization algorithm exists, surpassing all others globally. Instead, the optimization method should be determined by the problem's features, and insights specific to the problem can substantially enhance optimization's efficiency and effectiveness.

Given enough time, if any random search algorithm is used to explore the entire search space, the global optimum is eventually found (Simon, 2013). The following theorem guarantees finding the global optimum where $f(x)$ is a continuous function defined in a closed space with the global optimum $f^*(x)$ and t is a generation number.

$$\lim_{t\to\infty} f(x) = f^*(x) \tag{5.1}$$

At this point, it can be stated that random search presents the most significant diversification rate, while local search unveils the most notable intensification rate. Additionally, in the context of Evolution Strategies, where $\lambda = \mu = \rho$, non-elitist search denotes random search and displays the highest diversification rate. Elitism is invariably selecting the foremost individuals from parents and offspring, leading to rapid convergence comparable to local search. Therefore, adjusting the elitism rate can promote a balance between diversification and intensification. We have suggested using a novel elitism method that upholds the aforementioned equilibrium with a selection process based on parent and elite ranking, as illustrated in Figure 5.6. Intense colors in the figure indicate the probability that sets of parents and offspring ranked by fitness value will be selected for the next generation. For example, the intense blue area at G_{t+1} indicates that the parents are more likely to be included in the next generation, and the intense brown area indicates that the offspring are more likely to be selected. Light colors indicate that the selection of individuals to be transferred to the next generation is made randomly among the parents and offspring.

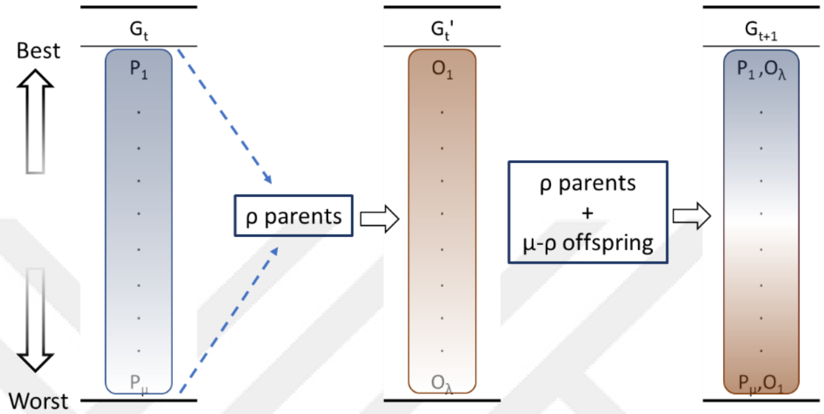


Figure 5.6 Proposed elitism option

The suggested elitism plan involves generating λ offspring from ρ parents in which μ represents the population size, forming an equal number of elite and parent individuals. The rank-based parent selection strategy is a straightforward option that offers higher selective pressure control and enables significant population differentiation, even in cases of closely scored fitness values in subsequent generations. The selection process may be completed using a predefined probability distribution function, which can assume a simple linear form or a more intricate structure. This will limit the likelihood of selecting any individual in the ranking, as the probability of selecting the fittest individual is restricted to double that of any other. In the end, the replacement mechanism follows a reverse selection process to replace the current population's members, with a preference for the fittest individuals. (Husbands et al., 2007).

5.2.3 *Sine Wave Optimization*

Sine Wave Optimization (SWO) was proposed as a means to simulate the undulatory shape that depicts balance between diversification and intensification over a timeline. Some optimization approaches are also derived from wave or trigonometric function theories in the literature. The Harmony Search Algorithm (HSA) is the most widely recognized example of such approaches, which was first proposed by (Geem et al., 2001). HSA is a meta-heuristic algorithm that simulates musicians' improvisation to find a harmonious melody with specific frequency relationships between multiple waves of different frequencies. The Water Wave Optimization (WWO) is a meta-heuristic proposed by Zheng (2015) that takes inspiration from the movements of water waves to design search methods for global optimization problems with high dimensions. The WWO consists of three operators. High-fitness waves can explore smaller areas thanks to propagation, whereas low-fitness waves are better suited to scanning larger areas. Diversity is enhanced by refraction, which reduces premature convergence. Breaking efficiently exploits the local space surrounding a promising location. Lastly, Mirjalili (2016) introduced the Sine Cosine Algorithm (SCA), which is a population-based optimization method. The SCA algorithm generates multiple initial random solutions, which move towards or away from the optimal solution using a mathematical model based on sine and cosine functions. The algorithm also includes various random parameters and an adaptive mechanism to utilize the entire search space during optimization at different stages.

When solving optimization problems, single-solution-based meta-heuristics commonly generate a solution that can be considered as "walks" through the problem's search space neighborhoods. These "walks" entail iterative procedures that move a certain distance in search space from the current solution towards the next. As a general rule, the trigonometric approaches underpinning the proposed solution methods are also associated with neighborhood searches. However, the SWO algorithm distinguishes itself by seeking the optimal solution via the degree of elitism established for the transitions between generations. A new solution method has been developed by integrating the SWO algorithm with the ES algorithm, whose general structure is given in Algorithm 5.1.

In the developed solution method, a trigonometric function $F(t)$ is proposed to determine the degree of elitism over generations to switch between exploration and exploitation-oriented searches. The function $F(t)$ given in Equation (5.2), where t is the generation number, is developed to determine the degree of elitism below.

$$F(t) = A(1 - (\frac{sin(\pi(ft + 0.5)) + 1}{2})^{\tau}) + C \tag{5.2}$$

The proposed solution method, whose broad definition is sine function-controlled elitism-based (μ, λ)-ES, is called SWO for short. As seen in the above equations, there are four main parameters in SWO.

$A \in \mathbb{R}^{+}$	Amplitude, the amount of increase from the centre of the peak to its highest point.
$f \in [0,1]$	Frequency, number of oscillations per iteration
$C \in \mathbb{R}^{+}$	Center amplitude, the lowest bound of the function
$\tau \in \mathbb{R}^{+}$	Convergence of waveform

The process of rank-based randomization guarantees that the resultant solution typically comprises distinct elite solutions, ultimately leading to a powerful diversification effect. In order to accomplish this, a random number, r, uniformly distributed on [0,1] with $F(t)$ is generated. This process is formulated in Equation (5.3). The population comprises of a total of individuals denoted by μ. The p^{*} real number is rounded off to the nearest integer to determine the selection of the rank-ordered population's (p*th best individual). The p* real number is rounded off to the nearest integer to determine the selection of the rank-ordered population's (p^{*}th best individual). Alternatively, the best offspring may replace the p^{**}th best individual. Equations (5.4) and (5.5) give formulations of $p^{*}$and p^{**} respectively. This selection and replacement process is implemented as specified in Algorithm 5.1 ($\text{e.g.}; \lambda = \mu, \rho = \varepsilon$).

$$r = (rand(0,1))^{F(t)} \tag{5.3}$$

$$p^* = round(r\mu) \tag{5.4}$$

$$p^{**} = round((1 - r)\mu) \tag{5.5}$$

Figure 5.7 shows how elitism varies across generations on the left. If the $F(t)$ value is higher, the search strategy becomes more similar to Local Search (LS), while if it is lower, it is more similar to Random Search (RS). On the right, the parental selection and offspring replacement of the population are displayed in order of ranking from highest to lowest. The blue dots indicate the location of parent selection in the population, while the brown dots indicate the location of individuals to be removed from the current population for placement of selected offspring. The concentration of blue dots above and brown dots below indicates the high degree of elitism and that the search is similar to LS. The random distribution of the points means that the search is similar to RS.

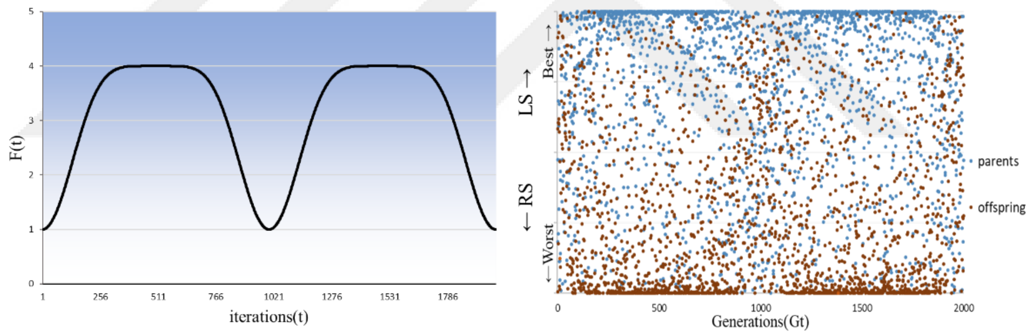


Figure 5.7 F(t) graph and parents' selection of SWO

To understand this algorithm, a simple SWO instance and its selection strategy are given;

Parameters: $A = 2$, $C = 1$, $f = 1/100$, $\tau = 1$. $\mu = 100$, $Iteration(t) = 30$,

Step 1:

$$F(30) = 2(1 - (\frac{sin(\pi(0.01(30) + 0.5)) + 1}{2})^1) + 1$$

$$F(30) \cong 1.41$$

Step 2:

$Randomly\ select\ a\ number;\ random\ (0,1)\ = 0.2$

$$r = 0.2^{F(30)} \cong 0.1030$$

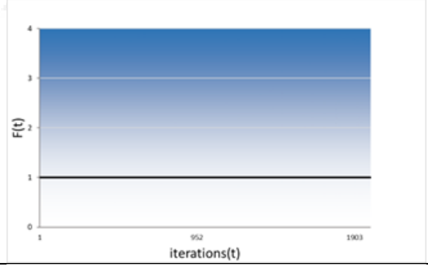
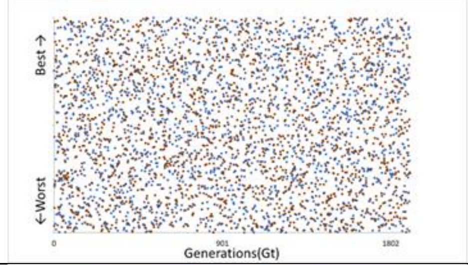
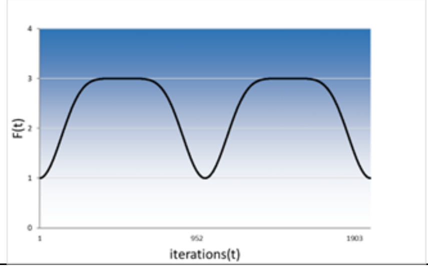
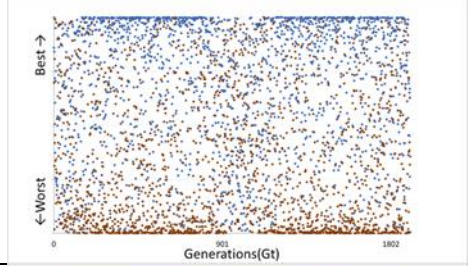
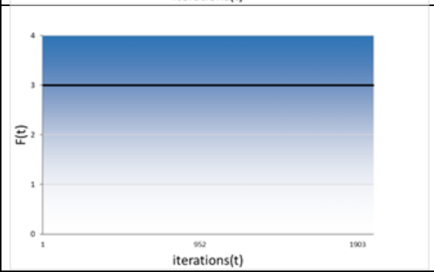
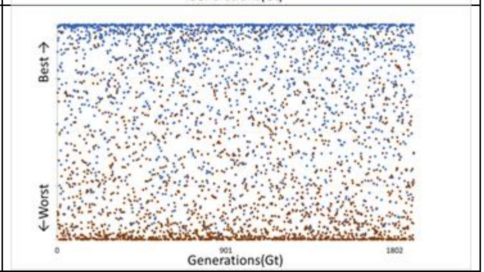
Step 3:

$p^* = round(0.1030 * \mu) = 10$ *(The parent selected was the individual with the 10th highest fitness score)*

$p^{**} = round(0.8970 * \mu) = 90$ *(One of the best fitness offspring will replace the 90th fitness offspring)*

As is typical of classical methods, the effectiveness of SWO is significantly influenced by adjusting internal parameters, particularly the degree of elitism. A plethora of variations can be created through the SWO parameters, yet this research concentrates on three significant variants. These parameter variations and their impact on the elitism degree are given in Table 5.1.

Table 5.1 SWO variants

Variants		Param.	Sine-wave function	Selection and Replacement in Population
1	No-Elitist (RS)	A=0, f=n/a C=1, τ=n/a		
2	Sine-Elitist	A=2, f=1/500 C=1, τ=3		
3	Elitist (LS)	A=0, f=n/a C=3, τ=n/a		

In Variant 1, there is no fluctuation because parameter A equals zero. This variant employs a completely random search, where the progeny of randomly selected parents are arbitrarily assigned within the population during the entire search procedure. Variant 2 aims to eliminate the local optimum at specified intervals by selecting individuals as parents, regardless of their fitness values, and replacing offspring with random individuals. Variant 3, akin to the LS technique, chooses the strongest individuals as parents during the procedure. The parameters of variation 2 were developed through the amalgamation of the other two variants to illustrate the oscillating impact of the sine wave.

In order to compare the performances of these parameter variations in routing and loading problems, they were tested with Solomon (1987) and Bischoff & Ratcliff (1995) instances. In 100 customer instances of Solomon (1987), customer locations are grouped as cluster (C), random (R), and mixed (RC). Every instance has been run ten times, and the evaluation is based on the average performance over the ten runs. The results for the Solomon (1987) instances, which are solved only by using the developed heuristic, with the objective function defined as the minimization of the total travel distance, are shown in Table 5.2. As can be seen, variant 2 achieves good results in random-type problems, and variant 3 achieves good results in cluster-type problems. Examining the results of all groups, it seems that variant 2 is slightly better than the others.

Table 5.2 Results of the parameter variants tested with Solomon (1987) instances

Group type	Num.of Ins.	variant 1	variant 2	variant 3
C1	9	1086.74	986.87	**982.94**
C2	8	757.19	757.57	**718.10**
R1	12	1328.13	**1263.35**	1280.99
R2	11	1070.05	**1009.17**	1033.79
RC1	8	1476.19	**1450.91**	1464.80
RC2	8	1335.46	1266.19	**1255.49**
Average		1175.63	**1122.34**	1122.69

On the other hand, when the convergence of the variants presented in Figure 5.8 is examined, it is seen that variant 3 achieves good results in a relatively short time, and variant 1 needs a very long number of iterations for good results. However, variant 2 achieves the best result in a reasonable time. The occurrence of this result can be interpreted as the avoidance of local optima due to the sinusoidal curve in variant 2.

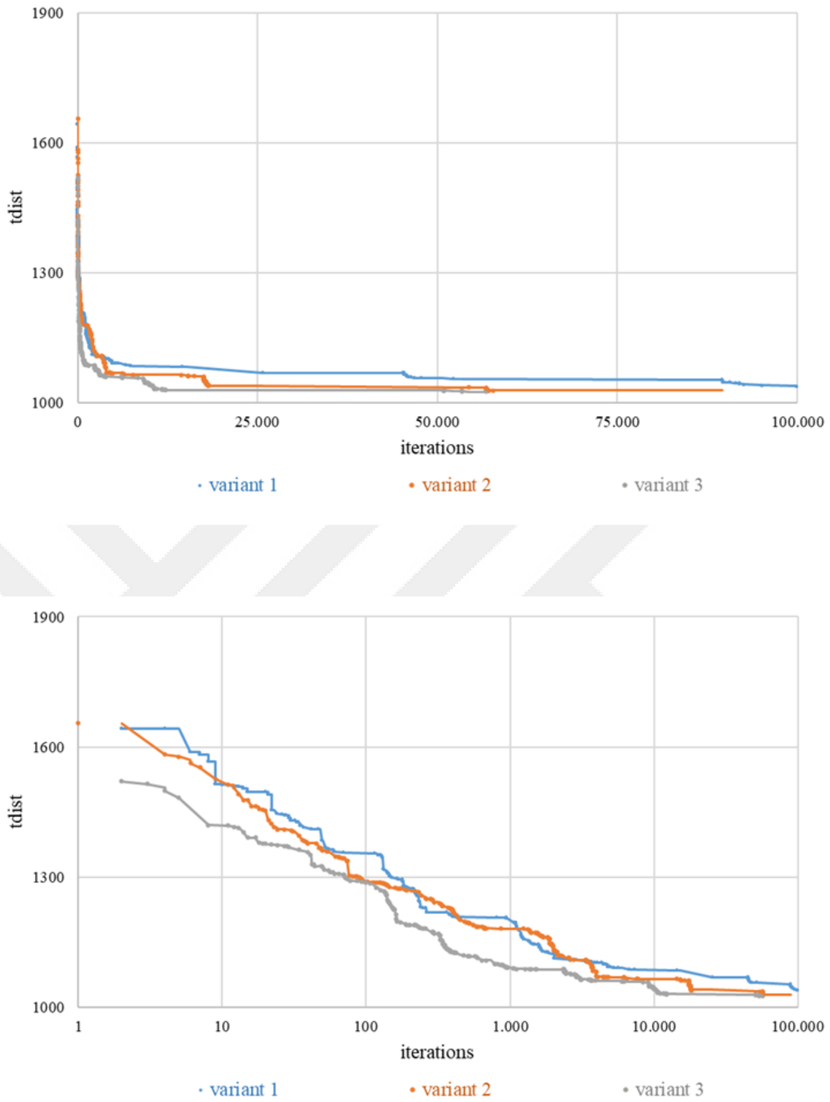


Figure 5.8 Comparison convergence of the variants' results (R104)

Then, the developed method was tested with Bischoff & Ratcliff (1995) instances to compare the effect of parameter variables on loading problems. The problem set, 700 of which are solved in this section, is divided into 7 test sets, each containing 100 problem examples from BR1 to BR7. The number of different box types is given as 3, 5, 8, 10, 12, 15, and 20 for each test. According to the decreasing average number of boxes per box type, the problem character in the test sets gradually changes from weak heterogeneous to strongly heterogeneous. The objective function is to maximize the use of the given container in the tests. The results are presented in Table 5.3. Variant 3 performed better in the results obtained. As a result, using the parameter settings of variant 2 when solving the routing part of the problem and variant 3 when solving the loading part will increase the quality of the results. However, in other loading problems of different types, the parameters may need to be adjusted.

Table 5.3 Results of the parameter variants tested with Bischoff & Ratcliff (1995) instances

	BR1	BR2	BR3	BR4	BR5	BR6	BR7	Average
No. Of Box types	3	5	8	10	12	15	20	
variant 1	0.865324	0.862831	0.817464	0.81609	0.795298	0.777763	0.765749	0.814360
variant 2	0.891711	0.882249	0.863378	0.84203	0.837064	0.829699	0.800055	0.849455
variant 3	**0.900325**	**0.887966**	**0.871943**	**0.851219**	**0.840549**	**0.832150**	**0.803811**	**0.855423**

5.3 Encoding and Decoding Procedure for the 3L-CVRP

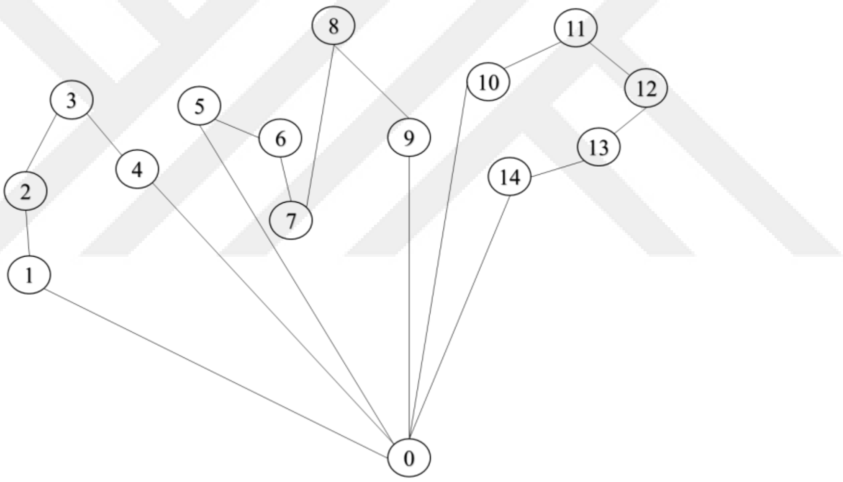
This section deals with the 3L-CVRP introduced by Gendreau et al. (2006), which was given in detail in the previous chapter. The encoding procedure is given in Section 4.3, for the three-dimensional box loading problem. Therefore, in this section, only the encoding procedure of the vehicle routing problem is given in the following text.

EAs are founded on the evolution of a community of agents. This evolution facilitates the exploration of at least a near-optimal solution for a specific problem. A fitness measure administers the population's evolution in the correct direction. The objective is to guide the population to discover the most optimal solution to a specific problem. In general, EA is associated exclusively with GA, however, this is not the sole plausible scenario. In this doctoral thesis, we present our novel elitism-based $(\mu, \lambda) - ES$ approach, which utilises its memory to explicitly tackle the Vehicle Routing Problem (VRP).

The suggested approach relies on a group of agents, residing within the solution space for a particular VRP instance. Each agent retains data about the current solution. The VRP instance solutions take the form of a chromosome representation recognised from the standard GA methods for the VRP. Genetic crossover or mutation operators are utilised to enact solution modifications.

Solution S of the VRP provides data regarding the routes taken by each vehicle $(route\ i)$ utilized within the solution. The vehicles all commence and terminate their routes at the depot. They then proceed to serve their customers, thereby developing a list of all customers to be visited in a specific order. The customers are denoted by a unique integer number $(n > 0)$, while the depot is denoted by 0. Each solution is a sequence of serviced customers for all vehicles separated with 0. For example, sequence s in form;

$S = \{0, 1,2,3,4,0,5,6,7,8,9,0,10,11,12,13,14, 0\}$ describes a solution to the VRP problem that uses three vehicles to service 14 customers, as illustrated in Figure 5.9.



	Route 1					Route 2						Route 2					
0	1	2	3	4	0	5	6	7	8	9	0	10	11	12	13	14	0

Figure 5.9 Chromosome Representation of the VRP solution

Each chromosome is assigned a fitness value that measures its solution quality. Since VRP has two objectives, chromosome fitness is generally evaluated according to two objective functions, assigning two values to each chromosome. The first shows the fitness value according to the number of vehicles used, and the second shows the fitness value depending on the total distance traveled.

Each chromosome, with its unique variables such as its sequence and fitness value, becomes a potential solution and is called a "candidate solution" or "individual." A group of individuals is referred to as the "population" of the EA.

Table 5.4 Representation of the population with μ individuals

#		Route 1				Route 2				Route 3			
S_1	0	1	…	4	0	5	…	9	0	10	…	14	0
S_2	0	5	…	4	0	12	…	1	0	8	…	3	0
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.	.	.
$S_{\mu-1}$	0	2	…	1	0	8	…	3	0	13	…	14	0
S_{μ}	0	10	…	14	0	10	…	6	0	4	…	12	0

5.4 Elitism-based (μ, λ)-ES algorithm for the 3L- CVRP

The new search algorithm for solving the 3L-CVRP consists of the elitism-based (μ, λ)-ES algorithm to obtain optimal vehicle routes and load the boxes of all customers assigned to a route into the vehicle's loading area.

The general procedure of elitism-based ES is shown in Figure 5.10. The first phase begins with generating a random initial population and determining the best solution based on that population. The generation number starts with $IterNum = 0$ and as the $IterNum$ value increases, the degree of elitism depends on the changes of the parameter values. In generations where the degree of elitism ($F_{IterNum}$) is high, the best individuals in the population are used as parents, and as the $F_{IterNum}$ value gets smaller; parents can be selected from low-quality individuals in the population and transfer their characteristics to new generations. With the change in $F_{IterNum}$ value, it provides opportunities for individuals at different levels to survive and become parents in terms of the fitness value of the population. Local search is supported by crossover and mutation among the best while searching across the population is supported by random search. At this stage, Varun Kumar & Panneerselvam's (2017) single point crossover operator and Yu, Yang & Yao's (2009) 2-opt mutation operator are referred to.

Evolutionary computing necessitates that the best offspring, as many as ρ, are added to the population in each generation; on the other hand, the same number of non-elitist individuals are discarded from the population (see Figure 5.6). This process, called replacement, is the most critical element of SWO because the selection of the individuals to be placed and to be removed will affect the evolution of the population to give the best solution.

The best solution achieved at the end of the $IterMax$ number of generations is the input for packing check in the second phase. The packing heuristic establishes feasible packing plans for each route, as detailed in section 4.3. To further speed up the search, all routes tried since the beginning of the algorithm with feasible packing plans are collected in a vector called $SolCache$. When a route is checked for packing, it is first searched in the cache. If it is not found, the packing heuristic is called, and the route is now cached with the result of the packing test. Among the routes in the cache, those with an appropriate packing plan are directly accepted. If a feasible packing plan is not found for a route, the packing control for the remaining routes is stopped. Any solution with an infeasible route is penalized for staying below the fitness ranking in the population. The infeasible route vector is cached and considered in the fitness evaluation of new individuals. The algorithm ends when a feasible packing plan is created for all routes.

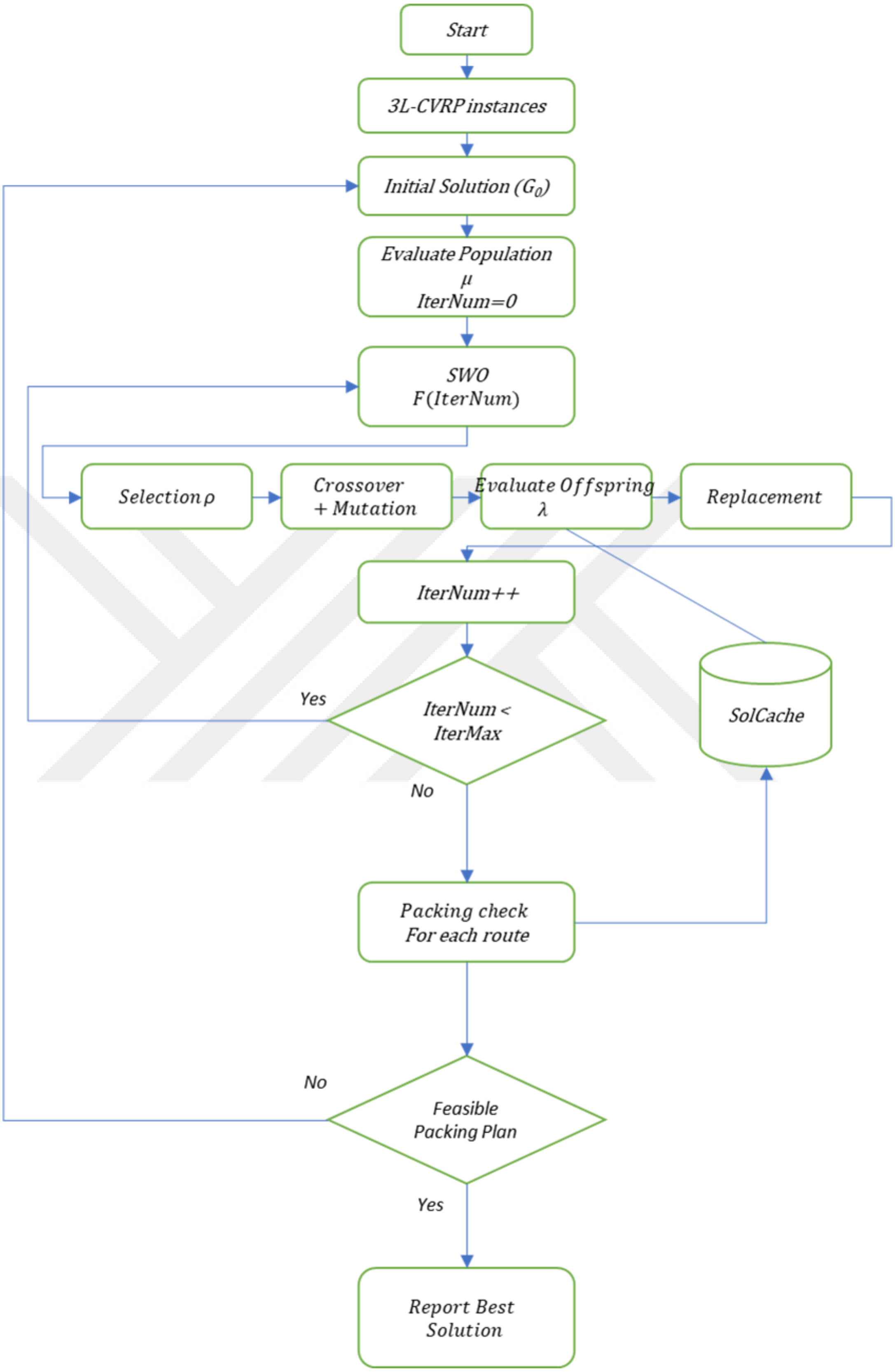


Figure 5.10 Flowchart of the Elitism-based ES for the 3L-CVRP

5.5 Computational Study

We used Gendreau et al.'s (2006) benchmark instances to examine the elitism-based (μ, λ)-ES algorithm's performance. The algorithms were coded in C# using Microsoft Visual Studio 2019, and all problem instances were solved on an Intel Core i5-7400 processor at 3.0 GHz using 12 GB of RAM. There are 27 problem instances, with the number of customers varying between 15 and 100. The nodes' coordinates, the weights of the items demanded by the customers, and the vehicle weight capacities were taken from the CVRP instances of Toth and Vigo (2002). These instances can be downloaded at https://data.mendeley.com/datasets/kzv3bd2bm2/1.

The proposed method SWO is compared with Gendreau et al.'s (2006) TSA, Tarantilis et al.'s (2009) hybrid algorithm (HA), Fuellerer et al.'s (2010) ACO algorithm, and Bortfeldt's (2012) VRLH in Table 5.5 using the instance set. The best overall travel distance per instance for each method is shown in the $tdist\_best$ column in the table. For ACO, HA, VRLH, and SWO, the average travel distance is computed as the total distance traveled averaged over the number of runs per instance, presented in the $tdist\_ave$ column. Also, for the ACO, HA, VRLH, and SWO, the average computation time per instance, $atime$, is computed as the total computation time averaged over the computed runs, given in seconds. Finally, the best total travel distances $(tdist\_best)$, the average total travel distances $(tdist\_ave)$ and total computation times $(atime)$ are averaged over 27 instances. The resulting average distance values are denoted by $tdist'$. Comparison of relative distance traveled with respect to SWO is given as the gap defined in Equation (5.6). For example, the gap between the SWO and ACO of Fuellerer et al. (2010) is calculated according to $gap = 100(tdist'\_SWO - tdist'\_ACO)/tdist'\_ACO$.

$$\%gap = 100\left(\frac{tdist'\_SWO - tdist'}{tdist'}\right) \tag{5.6}$$

For each method, the total number of vehicles used in the best solutions is shown in the nv line. Finally, in the $trate$ line, the average total computation time of the SWO across all instances is given as a percentage of the time of the average total computation time of the methods ($atime$') being compared (5.7). For example, $trate$ value between the SWO and ACO of Fuellerer et al. (2010) is calculated according to $trate = 100(atime'\_SWO/atime'\_ACO)$.

$$\%trate = 100\left(\frac{atime'\_SWO}{atime'\_}\right) \tag{5.7}$$

Table 5.5 shows the results that the new SWO method, which includes the sine-function controlled elitism-based (μ, λ)-ES algorithm, is clearly superior to the other methods in terms of solution quality. Of these, only the VRLH method has obtained results close to our proposed solution method in terms of solution quality and computation time. Evaluating all 27 instances, SWO achieved a new best solution for 17 instances. When the number of customers and loads in the instances increases, it becomes clear that more computation time is needed.

Our proposed solution method performs better in all aspects of solution quality, while the computational effort required by SWO and VRLH is much lower than other methods. This is evident when looking at the $atimes$ and $trates$. The time ratios are meaningful only if the processors used are of comparable power. To make a point, it can be said that the total computation time consumed by SWO and VRLH is approximately 10% of the total computation time required by the competitors.

In the second part of the computational study, the effects of the fragility constraint (FR), LIFO constraint, and support constraint (ST) on the solution quality were investigated, and the results for the five different loading configurations are shown in Table 5.6.

In the first configuration, all loading constraints were considered; in the next three configurations, the FR, LIFO, and ST constraints were ignored, respectively. In the last configuration, all these constraints were ignored. For each configuration and problem instance, the proposed algorithm was run ten times, showing the average total travel distance ($tdist$) and average computation time ($atime$) across all the runs.

The $gap(\%)$ value for a configuration represents the relative improvement of the total travel distances compared to the $all - constraints$ (all) configuration and is calculated according to the formulation shown in Equation (5.8). The total number of vehicles over the solutions obtained in the calculation for each loading configuration is indicated in line nv.

$$\%gap = 100x\left(\frac{tdist\_ - tdist\_\text{all}}{tdist\_}\right) \tag{5.8}$$

Results on the effect of restrictions are similar to findings in the literature; removing restrictions results in significantly shorter overall travel distances. Only removing the SP constraint presents unnecessary loading preferences due to the DBFL loading method, preventing it from finding a viable solution in a certain number of iterations. Therefore, almost the same results were obtained in the SP discarded configuration compared to the $all - constraints$ configuration. Moreover, removing restrictions leads to smaller vehicle numbers nv and lower average computation times$(atime)$.

In Table 5.6, the results of the five loading configurations obtained with SWO and competing methods are compared to show the best overall travel distance $tdist\_best$ and average total computation time $atime$. Additionally, the average total travel distance $tdist\_ave$ over the runs performed for the algorithms, if any, is presented. All values are averaged over the relevant configuration examples, and the best distance values are in bold.

Table 5.5 Results for the benchmark instances by Gendreau et al. (2006)

#	n	num. of boxes	Gendreau et al. (2006)		Tarantilis et al.(2009)		Fuellerer et al. (2010)			Bortfeldt (2012)			SWO		
			tdist	**atime**	**tdist**	**atime**	**tdist_best**	**tdist_ave**	**atime**	**tdist_best**	**tdist_ave**	**atime**	**tdist_best**	**tdist_ave**	**atime**
1	15	32	316.32	1800	321.47	13.20	304.13	305.35	12.00	302.02	302.02	72.30	**282.95**	286.64	28.94
2	15	26	350.58	1800	334.96	11.50	334.96	334.96	0.60	334.96	334.96	0.90	**334.96**	339.05	30.28
3	20	37	447.73	1800	430.95	540.60	399.68	409.79	121.80	392.63	401.44	182.00	**364.20**	371.36	153.11
4	20	36	448.48	1800	458.04	323.50	440.68	440.68	5.40	437.19	437.54	16.10	**430.88**	434.62	31.52
5	21	45	464.24	1800	465.79	99.60	450.93	453.19	30.90	443.61	451.03	182.60	**418.94**	422.32	96.45
6	21	40	504.46	1800	507.96	1212.40	498.32	501.47	18.40	498.16	498.38	23.60	**495.85**	497.58	31.03
7	22	46	831.66	1800	796.61	364.80	792.13	797.47	67.40	769.68	772.49	133.10	**742.23**	746.52	140.09
8	22	43	871.77	1800	880.93	230.00	820.67	820.67	78.60	810.89	821.35	139.10	**755.71**	758.21	192.36
9	25	50	666.1	1800	642.22	982.20	635.50	635.50	16.30	630.13	645.81	24.30	**630.13**	634.45	32.53
10	29	62	911.16	3600	884.74	1308.40	840.75	841.12	246.70	820.35	827.29	175.10	**758.18**	763.20	295.39
11	29	58	819.36	3600	873.43	522.50	818.87	821.04	199.80	803.61	815.62	136.40	**699.80**	708.49	328.07
12	30	63	651.58	3600	624.24	294.60	626.37	629.07	48.20	614.59	630.46	14.00	**610.00**	615.15	32.58
13	32	61	2928.34	3600	2799.74	2193.10	2739.80	2739.80	308.80	2645.95	2694.81	268.40	**2459.09**	2499.16	227.35
14	32	72	1559.64	3600	1504.44	4581.30	1466.84	1472.26	642.80	1368.42	1413.59	311.60	**1322.44**	1346.77	231.31
15	32	68	1452.34	3600	1415.42	2528.30	1367.58	1405.48	656.80	1341.14	1355.50	311.50	**1332.41**	1335.01	299.50
16	35	63	707.85	3600	698.61	4256.50	698.92	698.92	14.80	698.61	705.05	3.40	**698.61**	710.62	33.46
17	40	79	920.87	3600	872.79	2096.00	868.59	870.33	14.90	866.40	917.96	2.50	**861.79**	863.59	34.34
18	44	94	1400.52	3600	1296.59	2275.20	1255.64	1261.07	2209.80	**1207.72**	1228.98	309.50	1226.49	1231.61	347.19
19	50	99	871.29	7200	818.68	2509.00	777.18	781.29	623.60	**741.74**	753.87	416.50	767.90	776.47	280.21
20	71	147	732.12	7200	641.57	1940.90	604.28	611.26	3901.00	**587.95**	596.42	427.00	641.74	652.68	285.39
21	75	155	1275.2	7200	1159.72	2823.40	1110.09	1124.55	5180.60	**1090.22**	1107.00	443.40	1111.72	1113.22	432.06
22	75	146	1277.94	7200	1245.35	2685.60	1194.18	1197.43	2290.30	**1147.80**	1171.49	423.50	1188.91	1194.88	408.88
23	75	150	1258.16	7200	1231.92	4659.10	1158.51	1171.77	3727.60	**1130.54**	1135.46	425.80	1154.35	1155.82	432.37
24	75	143	1307.09	7200	1201.96	4854.10	1136.80	1148.70	1791.50	**1116.13**	1128.82	411.10	1179.16	1193.06	395.41
25	100	193	1570.72	7200	1457.46	5725.80	1429.64	1436.32	8817.10	**1407.36**	1428.80	453.00	1523.46	1538.77	479.97
26	100	199	1847.95	7200	1711.93	6283.10	1611.78	1616.99	6904.30	**1600.35**	1625.31	430.60	1706.66	1713.87	440.81
27	100	198	1747.52	7200	1646.44	9915.70	1560.70	1573.50	10483.90	**1529.86**	1550.85	435.00	1660.70	1680.51	409.98
	average		1042.26	4200	997.18	2415.90	960.87	966.67	1793.10	**938.44**	953.79	228.60	939.23	**947.54**	**227.06**
	gap(%)		-8.5		-5.9			-1.3			0			0	
	nv		n/a		295			288			274			281	
	Pro.		Pentium iv (3.0 GHz)		Pentium iv (2.8 GHz)			Pentium iv (3.2 GHz)			Intel (3.17 GHz)			Intel Core i5-7400	
	nofruns		1		?			10			10			10	
	trate		5.4		9.3			16.6			99			100	

Table 5.6 Average results of competing methods and five loading configurations.

Loading configuration	Gendreau et al. (2006)		Tarantilis et al. (2009)		Fuellerer et al. (2010)			Bortfeldt (2012)			SWO		
	tdist_best	**atime**	**tdist_best**	**atime**	**tdist_best**	**tdist_ave**	**atime**	**tdist_best**	**tdist_ave**	**atime**	**tdist_best**	**tdist_ave**	**atime**
All constraints	1042.26	4200.0	997.18		960.87	966.67	1793.1	**938.44**	953.79	228.6	939.23	**947.54**	267.19
No fragility	1014.49	4200.0	965.14		940.3	945.04	1375.1	**920.21**	**934.38**	232.4	928.43	937.47	256.69
No LIFO	951.19	4200.0	950.59		910.34	916.25	716.8	**900.55**	912.05	205.6	904.32	**911.44**	254.70
No support	939.53	4200.0	934.96		914.72	919.69	1376	**889.64**	**902.55**	237.3	943.25	949.90	256.87
3D loading only	876.31	4200.0	876.39		**854.39**	**856.67**	741.5	854.45	864.53	209.4	871.95	880.48	252.71

For four of the five loading configurations, the VRLH method obtains the best $\_best$, as well as the smallest $atime$ over all runs. The ACO method performs better in terms of $tdist\_best$ and $atime$ if all constraints are removed. However, SWO achieved better results on $tdist\_ave$ for the three configurations, and the other values obtained were very close to the best ones.

In general, removing some of the limitations of the 3L-CVRP results in better and faster results. The SWO, like other competing methods, increased its performance as the constraints reduced and was able to maintain this performance more consistently than others.

5.6 Chapter Summary

This chapter discusses the newly developed EA-based solution method. This solution method balances diversification and intensification by increasing and decreasing the elitism ratio with respect to a newly defined sine wave function. The parameters that determine the shape of the sinus curve change depending on the number of iterations, thus avoiding the local optima by performing an LS in some periods of the search and an RS in others. Contrary to classical EAs, in this method, more parent selection and offspring replacement come to the fore than crossover and mutation operators. By making the elitism dynamic, it is possible to search different parts of the solution space by means of individuals. In solving different types of problems, satisfactory solutions have been obtained in a short time with the flexible parameter settings of the new method. First, it was tested with Solomon (1987) instances, which are often used to evaluate the performance of studies in the VRP literature. When the results obtained with different parameter values were compared, it was seen that the $sine - elitist$ version achieved the best result, and this result was of a quality that could compete with the best-known results. Then, in order to measure the performance of the method in other problem types, the developed method was tested with Bischoff & Ratcliff (1995) instances, which are the leading test examples of the CLP literature related to our thesis topic. According to the results, the flexible structure of the developed model has proven that it can be used effectively in loading problems.

In this chapter, Gendreau et al. (2006) instances, one of the well-known instances set of the 3L-CVRP, are solved with the developed solution method. The instances solved using the parameter settings obtained from the solutions of the routing and loading problems in the previous section achieved very competitive results. For the problem set, five different loading configurations are specified. One of these configurations, considering all the constraints, was compared with other studies in the literature, and according to the quality of the results obtained, the performance of the new method was found to be superior and stable than the other methods. When the configurations are compared within themselves, reducing the constraints in parallel with other studies has led to an increase in the quality of the solution and a shorter (about 10-20%) solution time.

CHAPTER SIX
PICKUP AND DELIVERY PROBLEMS WITH LOADING CONSTRAINTS (3L-PDP)

6.1 Introduction

This chapter focuses on the generalized version of the PDP to an integrated routing and three-dimensional loading problem. The 3L-PDP involves receiving requests as sets of three-dimensional rectangular elements and loading them into vehicles with three-dimensional rectangular compartments.

The 3L-CVRP with LIFO restrictions ensures that the boxes are not moved after loading or before unloading. However, the 3L-PDP has more intensive loading and unloading activities; therefore, additional constraints are needed to reduce unnecessary effort. These new constraints result in the set of 3L-PDP variants described in the following sections.

The rest of this chapter is organized as follows: Section 6.2 describes the studied problem. The solution approaches in the literature are presented in Section 6.3 of solution approaches in the literature. The solution representation of the problem and the procedure of the packing phase are given in section 6.4 and section 6.5, respectively. Section 6.6 describes the crucial features of the proposed evolutionary algorithm. In Section 6.7, the results of the computational study are presented and analyzed. Finally, Section 6.8 outlines the chapter and provides an overview of future research.

6.2 Problem formulation

The PDP formulation can be used as a starting point to model the 3L-PDP. Like all distribution problems, the PDP has the following assumptions and limitations;

- each of the routes starts and ends at a single depot,
- each request is served on a single route,
- the pick-up point is visited before the delivery point,

- the loaded set of goods never exceeds the capacity of the vehicles,
- the number of routes does not exceed the number of vehicles given,
- the shipping cost given by the total travel distance is minimized.

The 3L-PDP, derived from Bartók & Imreh (2011) and Männel & Bortfeldt (2015), can be formulated as the following;

The PDP can be defined on a directed graph $G = (N, A)$ with node set $N = \{0, \dots, n+1\}$ and arc set A. Nodes 0 and $n+1$ represent the origin and destination depots (which may have the same location), while subsets $P = \{\dots\}$ and $\mathrm{D} = \{\dots\}$ represent pick-up and delivery nodes, respectively.

Let r denote the number of requests to satisfy. Each request $r \in R$ is associated with a pick-up node $P(r)$ and a delivery node $D(r)$. Each node $i \in N$ is associated with a load q_i and a non-negative service duration d_i satisfying $d_0 = d_{n+1} = 0$, $q_0 = q_{n+1} = 0$. An unlimited fleet of identical vehicles with capacity C and cost cv is available to serve the requests. Each $arc\ (i, j) \in A$ is associated with a routing cost c_{ij} and a travel time/distance t_{ij}.

Moreover, each node i also has a time window $[a_i, b_i]$, $i \in I \cup D$, where a_i and b_i indicate the possible earliest and latest service start times, respectively. The service may begin between these times at node i. The depot usually has time windows represented as $[a_0, b_0]$ and $[a_{n+1}, b_{n+1}]$. Vehicles can leave and return to the depot within time windows.

There is also a set of boxes that must be delivered to customers $N' = \{1, \dots, n\}$. Each box t is characterized by the length (l_t), width (w_t), and height (h_t) dimensions. The loading space (L, W, H) of each vehicle is embedded in the first octant of a Cartesian coordinate system such that the length, width, and height of the loading space lie parallel to the x-, y-, and z-axis. The zx_{tk}, zy_{tk}, and zz_{tk} coordinates give the placement of box t in the vehicle k's loading space. They indicated the distances from the corner of the box closest to the origin of the coordinate system (see Figure 6.1); in addition, an orientation index β_{tps}^{k} indicates which of the possible spatial orientations of box t for vehicle k is selected ($p \in \{l, w, h\}$ and $s \in \{x, y, z\}$). The spatial orientation of a box is determined by a one-to-one mapping of the three box dimensions and three coordinate directions. A detailed explanation of the mathematical representation of the orientation of the boxes is given in section 4.2.

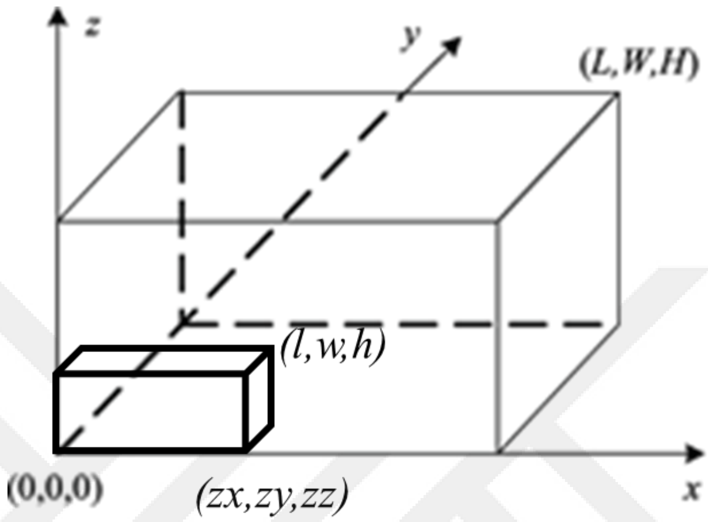


Figure 6.1 Replacement of a box in the loading space of the vehicle

Q_i^k represents the loading plan of vehicle k when leaving node i. A loading scheme for a designated loading area encompasses one or multiple placements and is considered viable provided the succeeding requirements are met: each box is fully contained within the valid loading space; no two boxes scheduled within the same truck-loading area occupy the same space; each box is aligned parallel to the loading space's surface area.

In addition, the following routing and packing constraints are satisfied:

Weight constraint (WC): Each box $t \in O$ has a positive weight and the total weight of all the boxes in a loading plan Q_i^k must not exceed the maximum load weight C.

Orientation constraint (OR): Each box has six possible placements, but some boxes may have OR. Thus, only two of the six values are allowed for the orientation index β_{tps}^{k} of the placement for those boxes. For instance, if box t in vehicle k has the OR, the variable β_{t33}^{k} must be equal to one. Thus h_t' determined as the initial and unrotated height value becomes the precise height value of the box $(h_t' = h_t)$. On the other hand, β_{t11}^{k} or β_{t12}^{k} values can be one (See Figure 6.2).

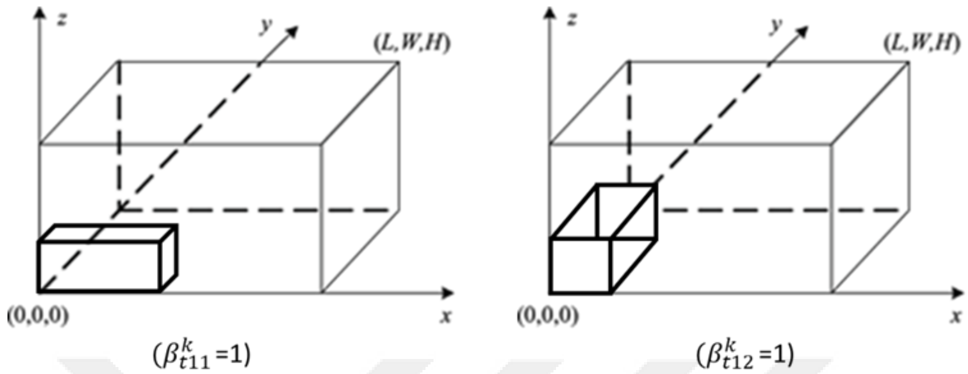


Figure 6.2 Orientation constraint of box t in vehicle k

Stability (Support) constraint (ST): If a box is not placed on the floor, a certain percentage of its base area must be supported by other boxes.

Stacking constraint (FR): A fragility attribute f_t is assigned to each box t. If a box is fragile $f_t = 1$, only other fragile boxes may be placed on its top surface, whereas both fragile and non-fragile boxes may be stacked on a non-fragile box ($f_t = 0$).

The objective function in this problem is to transport all requests with a minimum cost solution given an available number of vehicles without violating the defined constraints. The cost of the solution is assumed to be the total cost of the routes traveled by the vehicles and the cost of vehicles. Then, the objective function is:

$$Minimize \sum_{k \in V} \sum_{i \in N} \sum_{j \in N} c_{ij}\ x_{ij}^{k} + \sum_{k \in V} cv\ y^{k} \tag{6.1}$$

Where: $x_{ij}^{k} = 1$ if arc (i,j) is traversed by vehicle k, and 0 otherwise; $y^{k} = 1$ if vehicle k is used, and 0 otherwise.

6.3 Literature review for the 3L-PDP

Since the PDP was first introduced by Desrochers et al. (1988), several studies have focused on solving practical examples of the PDP using a variety of techniques, such as branching methods (Ropke & Cordeau, 2009), decomposition methods (column building, Desrosiers & Dumas (1988); row generation, Dumas, Desrosiers & Soumis (1991)), dynamic programming (Jun & Lee, 2022), heuristics (Ropke & Pisinger, (2006); Li & Kuhl (2017)), and meta-heuristics (Curtois, Landa-Silva, Qu & Laesanklang (2018); Zhu, Xiao, He, Ji & Sun (2016)).

In most static applications, the typical objective functions to be minimized are, first, the fleet size needed to satisfy the demand with a determined level of service and, second, the total distance traveled by all vehicles, provided that the fleet size is fixed. The static PDP constraints can be classified as visiting constraints (each pick-up and delivery must be visited exactly once), vehicle capacity constraints, depot constraints, coupling constraints (stating that for a given request, the same vehicle must visit the pick-up and delivery stops), precedence constraints, resource constraints on the availability of drivers and vehicles, and time window constraints to be satisfied at each stop if time windows are explicitly defined (Desrosiers et al., 1995). For a survey on the PDPs until 2008, see Parragh, Doerner & Hartl (2008).

Savelsbergh & Sol (1995) classified the pick-up and delivery problem into static and dynamic cases, with single and multi-vehicles and with and without time windows. Solomon & Desrosiers (1988) also published an excellent review of the Pick-up and Delivery Problem with Time Windows (PDPTW) as a special case of the vehicle routing problem with time window constraints. Subsequently, Toth & Vigo (2002) surveyed the methods proposed to solve the PDPTW.

Li & Lim (2001) proposed a tabu-embedded simulated annealing algorithm that restarts a search procedure from the current best solution after several non-improving search iterations. They proposed six newly generated datasets with various distribution properties for the PDPTW problem. Bent & Van Hentenryck (2006) proposed a two-stage hybrid algorithm for the PDP, wherein in the first stage using the Simulated

Annealing (SA), the number of vehicles is minimized; in the second stage, the total travel cost is minimized by a large neighborhood search. Ropke & Pisinger (2006) proposed an adaptive large-neighborhood search heuristic. Nagata & Kobayashi (2010) successfully applied a Guided Ejection Search (GES) algorithm to the PDPTW. Nalepa & Blocho (2016) proposed a parallel GES algorithm to solve the PDPTW. In their approach, parallel processes cooperate periodically to enhance the quality of the results and accelerate the convergence of computations. Küçük & Yildiz (2019) introduced a novel Constraint Programming model for the PDPTW. Nasiri, Keedwell, Dorne, Kern & Owusu (2022) proposed the application of a sequence-based selection hyper-heuristic to the one-to-one, static, and deterministic variant of the PDPTW.

Bartók & Imreh (2011) first discussed the PDP with three-dimensional loading constraints. Their algorithm is summarized as follows: In the initialization step, routes are created on the graph given for each vehicle, considering the pairs of vertices with requests. This step is repeated until there is a pair of unassigned vertices. After this step, a simple local search method is applied, in which vertices are swapped along a route as long as the total distance of a route is reduced while maintaining the linear ordering of the vertices. After this simple local search step, the packing algorithm is executed. The above steps are repeated as long as there are no uninitialized vehicles and at least one unmoved item. The final step is completed by applying an advanced local search method, in which vertices are swapped between routes while maintaining the feasibility of packing.

Männel & Bortfeldt (2016), named the integrated PDP and three-dimensional loading problem "PDP with three-dimensional loading constraints (3L-PDP)". They also rearranged the definition: In the 3L-PDP, with a given request group and a homogeneous fleet of vehicles, a set of routes is determined by the minimum total length for each demand to be transported from a loading area to a corresponding unloading area. Each request is given as a series of three-dimensional rectangular items, and vehicle capacity is replaced by three-dimensional loading space. They also explored the constraints that would be used to ensure that no boxes were moved after loading or before unloading. For this, they proposed a hybrid algorithm to solve the 3

L-PDP, which consists of a large neighborhood search for routing and a tree search heuristic for the packing procedure.

Männel & Bortfeldt (2018) are the second in a series of articles dealing with the constraints on the 3L-PDP that ensure that there will be no reload effort as in the first. Here, they focused on the ban on reloading, a packing constraint that allows identical boxes to be placed in different packing schemes. This constraint provides better solutions regarding travel distance than a routing constraint used in the first study to avoid any reloading effort. The article's primary focus is the packing procedure designed as a tree search algorithm and the corresponding concept of packing controls. A large neighborhood search procedure for routing and the packing procedure creates a hybrid algorithm for the 3L-PDP. Computational experiments were performed using 54 3L-PDP benchmark instances.

Moreover, Dupas, Grebennik, Lytvynenko & Baranov (2017) proposed a mathematical model and a solving strategy, including the GenBase algorithm (GBA) for the 3L-PDP. Wu, Zheng, Huang, Cai, Feng & Zhang (2019) proposed an improved LNS algorithm. They also added the tabu strategy to the neighborhood search to improve the search capability of the algorithm and added a new evaluation function to the loading algorithm. Chi & He (2023) discussed the differences between the pickup and delivery scenarios in detail and introduced the Pickup Capacitated Vehicle Routing Problem with Three-Dimensional Loading Constraints (3L-PCVRP). They solved the 3L-PCVRP with a branch-and-price-based algorithm and used a label-correcting-based algorithm (LCA) for routing and TRSA for the loading part of the problem. An overview of the literature on the 3L-PDP and the solution methods for each study, the problem characteristics, and the constraints considered are summarized in Table 6.1.

In this chapter, we adapted the elitism-based ES algorithm that was developed in Chapter 5 for the 3L-PDP with time windows. The solution method has been applied to both the routing and loading parts of the problem. Weight Constraint (WC), Orientation Constraint (OR), Stability (Support) Constraint (ST), and Stacking Constraint (FR) have been taken into account in solving the problem.

Table 6.1 Detailed information about the literature on the 3L-PDP

Authors	Solution methods		ToV		Problem characteristics and constraints								
	Heuristic	Exact	Hmg	Het	TW	CLC	WL	WD	LIFO	FR	PD	BH	RB
Bartók & Imreh (2011)	LS		√			√	√				√		
Männel & Bortfeldt (2016)	ALNS		√			√	√		√	√	√		
Männel & Bortfeldt (2018)	LNS+TRSA		√			√	√		√	√	√		√
Dupas et al. (2017)	MIP+GBA		√			√	√				√		
Wu et al. (2019)	LNS		√			√	√		√	√	√		
Chi & He (2023)	LCA+TRSA		√			√	√			√	√		
Chapter 6	ES		√		√	√	√	√	√	√	√		

ToV: Type of Vehicle fleet Het: Heterogeneous, Hmg: Homogeneous TW: Time Windows, CPC: Classical Loading Constraints, WL: Weight Limit, WD: Weight Distribution, LIFO: LIFO Loading, FR: Stacking (Fragility), PD: Pickup and Delivery, BH: Backhauls, RB: Reloading Ban

6.4 Proposed elitism-based ES algorithm for the 3L-PDP

In this section, the application of the elitism-based ES presented in Chapter 5 to the 3L-PDP is explained in detail. First, the order of visiting customers in the PDP is presented in the form of an array, and the effect of considering LIFO and RS constraints on both routing and loading is explained with the help of figures.

6.4.1 Solution representation

The representation of the PDP problem in the form of an array S should contain information about the routes of all the vehicles used in the solution. Each vehicle starts and ends its tour at the depot. It then serves its customers and creates an ordered list of all customers to be visited. Customers are represented by a unique integer ($n > 0$), while the warehouse is denoted by 0. Each representation is the array of customers served for all vehicles separated by 0.

For example, $S = \{0, P1, P2, D2, D1, P5, D5, 0, P3, D3, P4, D4, 0\}$ describes a solution representation of the PDP seen in Figure 6.3 that uses two trucks to service ten customers.

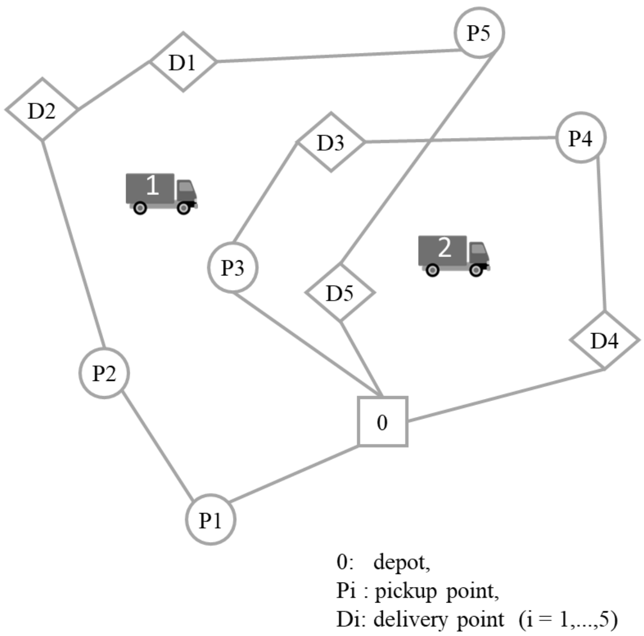


Figure 6.3 Solution representation of the PDP

As mentioned in Section 6.1, the PDP is tasked with homogeneous vehicles for a series of requests from a pick-up point to a delivery point, whereas in the 3L-PDP, the requests consist of sets of boxes and are transported in three-dimensional compartments of the vehicles. Furthermore, for rear-loading vehicles, it is assumed that goods are loaded and unloaded from the rear only by movements in the vehicle's longitudinal direction. Therefore, lifting or moving the boxes in the width direction is prohibited during loading or unloading.

There are different practical ways to prohibit the reloading of goods during a pick-up and delivery tour, such as lack of workforce, hard work time, lack of equipment, and extra heavy, fragile, or even dangerous goods. To eliminate any reloading effort, the first way to load the respective boxes is by applying the well-known Last-In, First-Out (LIFO) constraint.

For instance, suppose that there are two pickup points (A, B) and two delivery points (A', B') related to customers A and B. Let us call the boxes belonging to customer A as $box\ set\ A$, and the ones belonging to customer B as $box\ set\ B$. At delivery point A, the LIFO constraint ensures priority unloading of $box\ set\ A$ from the rear of the vehicle. It requires that $box\ set\ B$, which needs to be unloaded later, is not located between $box\ set\ A$ and the rear of the vehicle. Likewise, any of the B boxes must not be supported by the A boxes (For details, see Section 2.6). Otherwise, for $box\ set\ A$ to be unloaded, it becomes necessary to reload $box\ set\ B$.

Since pick-up points also occur on a pick-up and delivery tour, there must be a request sequences (RS) constraint at loading locations to prevent the reloading of goods. The RS constraint at a pick-up point requires no loaded $box\ set\ B$ at a previous pick-up point between the location of a newly loaded $box\ set\ A$ and the rear of the vehicle (Männel & Bortfeldt, 2018). The idea of an RS for the 3L-PDP consisting of a series of sub-tours is quite simple. Each sub-tour is a series of pick-up points followed by corresponding delivery points in reverse order.

Table 6.2 contains data on the illustrative 3L-PDP example of the five requests. The problem was solved using these data in two ways: with the RS constraint and without this constraint. As seen in Figure 6.4, when the RS constraint is not taken into account, the arrows of the start and distribution pairs intersect with each other, and when the RS constraint is taken into account, independent sub-tours are formed within the vehicle's tour. Figure 6.5 illustrates the 3L-PDP instance and the route of a possible solution without the RS constraint. The total distance is approximately 109. Figure 6.6 shows the loading space left during the departure from each customer. Both figures show that the overall distance is small, and the loading area is used very efficiently. However, the reloading effort is very high at almost all the loading points.

Table 6.2 Description of the 3L-PDP small-size instance derived from Männel & Bortfeldt (2015)

Maximum number of routes	Number of requests	Maximum load weight of vehicle	Length of loading space (L)	Width of loading space (W)	Height of loading space (H)
10	5	45000	60	25	30

Number of depot (= 0)	X-coordinate of depot	Y-coordinate of depot	Number of depot (= 0)	X-coordinate of depot	Y-coordinate of depot	Maximum duration of route
0	70	70	0	70	70	1351

Number of Request	pick-up node			delivery node			Sum of weights of the boxes	#boxes	Box 1				Box 2				Box 3				Box 4				Box 5			
	X-coordinate	Y-coordinate	Service time	X-coordinate	Y-coordinate	Service time			l	w	h	f	l	w	h	f	l	w	h	f	l	w	h	f	l	w	h	f
1	67	83	40	66	96	40	31806	4	30	11	10	0	25	10	12	1	23	14	9	1	26	6	9	1				
2	59	65	40	74	95	40	3388	1	22	11	14	1																
3	60	85	40	81	81	40	6828	3	18	11	12	0	24	12	7	0	29	14	6	1								
4	57	67	40	60	70	40	10296	1	24	13	11	1																
5	73	84	40	74	63	40	13794	5	22	12	15	0	34	9	9	0	18	12	10	0	12	14	15	1	30	10	8	1

#boxes: Number of boxes, l: Length of box, w: Width of box, h: Height of box, f: Fragility of box

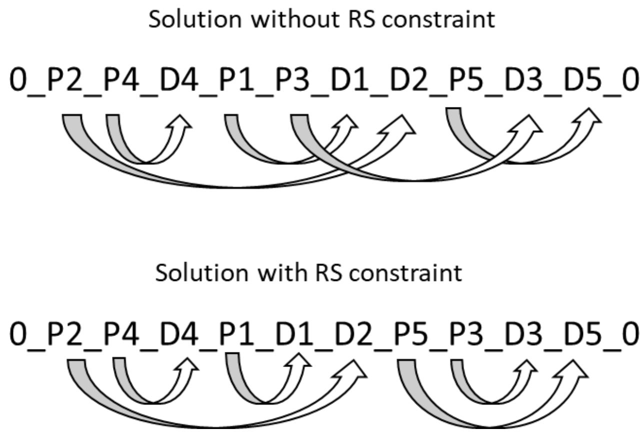
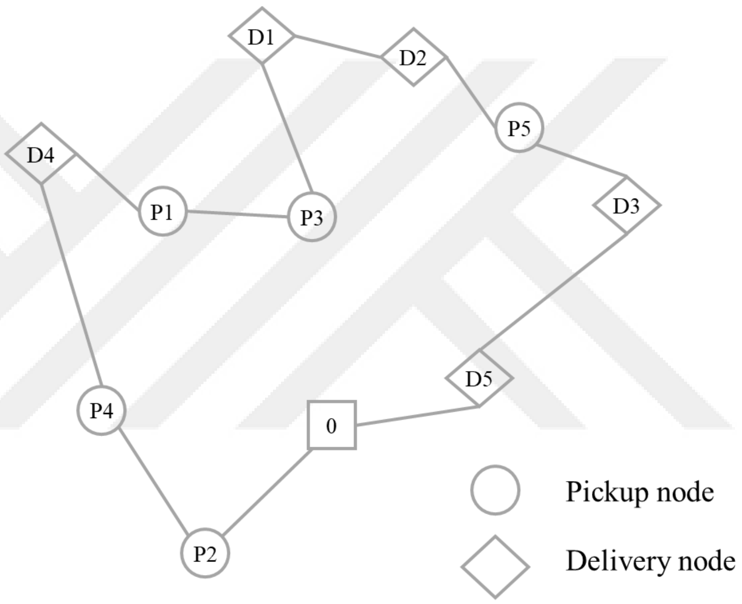


Figure 6.4 Solutions with and without RS constraint



0_P2_P4_D4_P1_P3_D1_D2_P5_D3_D5_0

Figure 6.5 The vehicle route for the illustrative example without the RS constraint

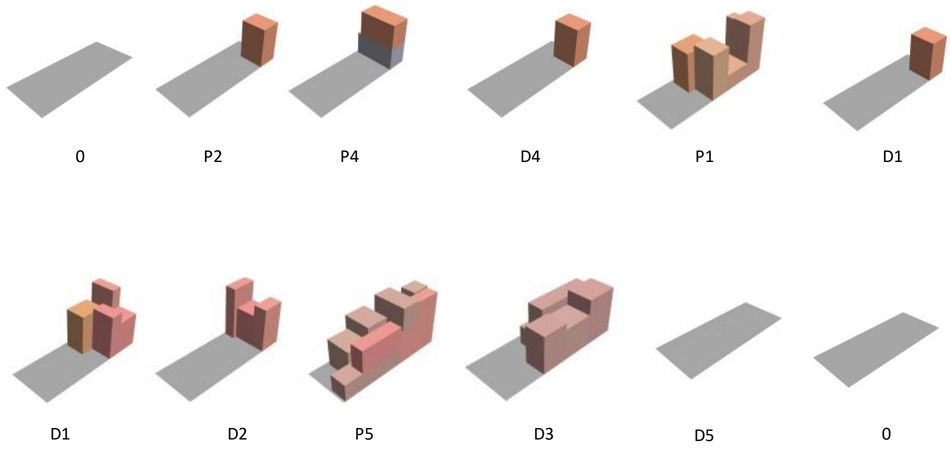


Figure 6.6 Illustration of the loading space while departing from each customer (without the RS constraint)

Figure 6.7 and Figure 6.8 show the solution to the example problem with RS and LIFO constraints that require no reloading effort. The total travel distance is approximately 127. These figures show that if a route meets the RS constraint, the loading area will be completely empty after a sub-tour (P2_P4_D4_P1_D1_D2 and P5_P3_D3_D5) is completed. Therefore, the packing plans required for the next sub-rounds are independent of each other. For example, a request box should only be stacked with request boxes from the same sub-tour. Thus, only one packing plan is required per request, and there is no need to reload any box.

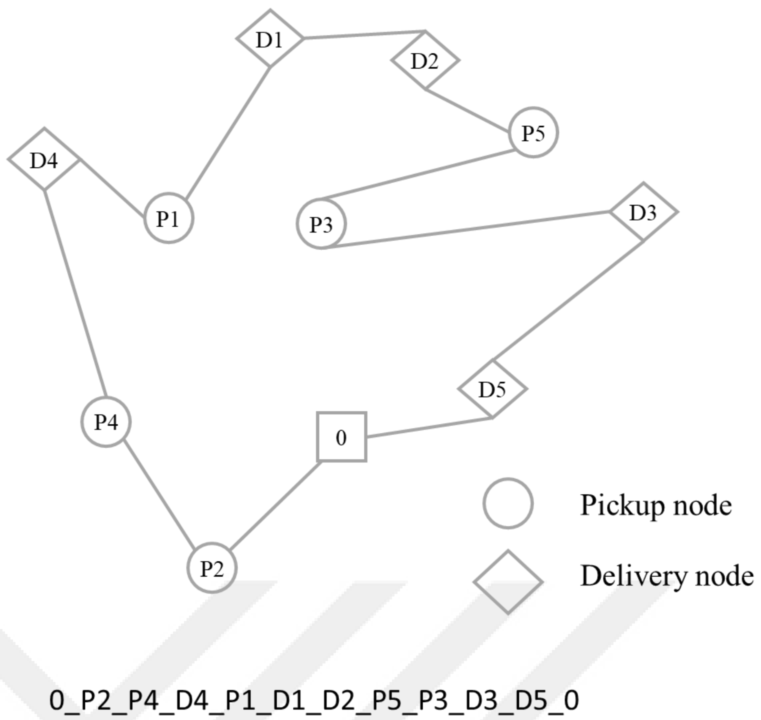


Figure 6.7 The vehicle route for the illustrative example with the RS constraint

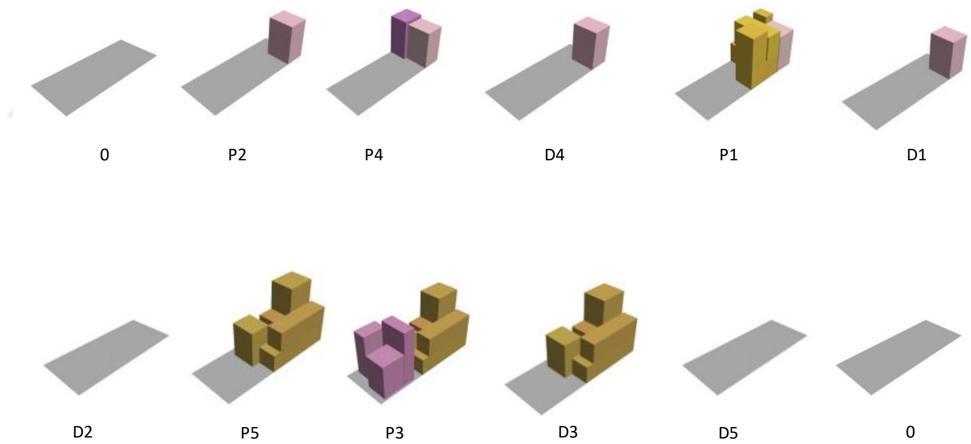


Figure 6.8 Illustration of the loading space while departing from each customer (with the RS constraint)

6.4.2 Routing procedure

The routing procedure is summarized in the following Algorithm 6.1.

Algorithm 6.1 *Routing procedure of the 3L-PDP*

```
  P ← population size
  G ← generation size
  Find the initial population (G_0)
  it ← 1                          ⇒iteration/generation number
 while it ≤ G do
    Apply the elitism procedure to obtain SWO rate
    i ← 1                         ⇒population number
    while i ≤ P do
       Select parents using SWO rate
       Reproduction and mutation offspring
       Compute fitness value of offspring
       i ← i + 1
     end while
    Apply replacement mechanism using SWO' rate
     it = it + 1
 end while
```

6.4.2.1 Population Initialization

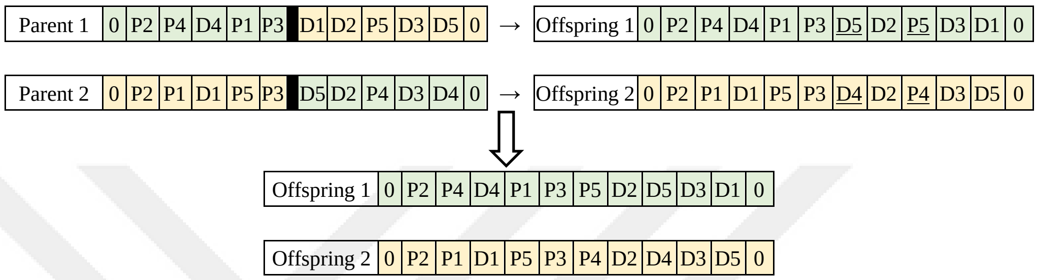
First, an initial population (G_0) is randomly generated. It is expected that a sufficiently structured population of solutions will progress to high-quality solutions within a few generations of the EA. Nonetheless, a potential drawback is that such a population may have a limited diversity, which would prevent the ability to obtain near-optimal solutions that are as good as other search methods. Due to limited vehicle numbers, the initial solution or subsequent ones may not be able to accommodate all requests. In order to address this issue, a virtual request bank, as in the algorithm, is utilized.

The objective function of the PDP (given Eq. (6.1)) is to transport all requests using the provided vehicles, achieving the lowest cost solution and avoiding any breach of defined constraints.

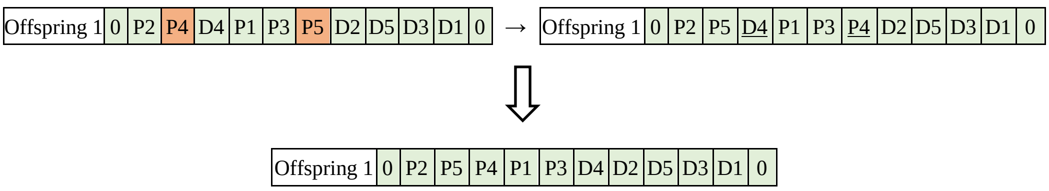
6.4.2.2 The process of reproduction

Two individuals are randomly selected as parents from the population using the elitism approach controlled by the sine function.

Offspring are created through a standard crossover procedure using both parent solutions. The most optimal outcomes were achieved using a one-point crossover, in which a single point on the chromosome is randomly selected. Offspring consists of the gene values inherited from Parent One, located to the left of the point, and Parent Two, located to the right of the point. The second offspring also consists of unselected gene values. After the formation of the offspring, the delivery points before the pickup points in the gene sequence are corrected by swapping.



Mutation processes within the context of evolutionary algorithms alter each offspring at a predetermined probability. The purpose of the mutation operation is to randomly mutate the offspring and thus produce a new solution that is not very far from the original one. In the proposed ES algorithm, the swap method is applied as a mutation operator.



The proposed evolutionary algorithm uses the elitist approach by adding eligible offspring to the population immediately when they are produced while simultaneously removing inferior individuals to maintain a constant population size.

6.4.3 Loading procedure

For a given request loading sequence and corresponding sets of boxes, the packing procedure tries to determine a complete solution of the loading plan that stacks all the given boxes. Similar to Section 4.3.3, each representation of the loading procedure is a permutation of a series of boxes (O_k). O_k is a dynamic list of the boxes and indicates the boxes currently available in the vehicles' loading spaces.

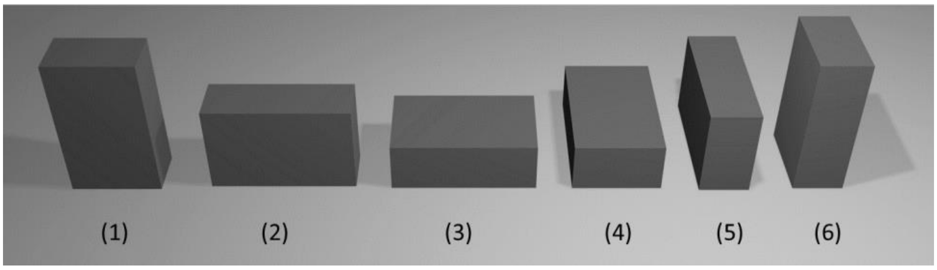


Figure 6.9 Six orientations of a box

In addition, there are O'_k permutations of a series where the packing order and orientation information of the boxes are stored. The O'_k series indicates which of the orientations shown in Figure 6.9 will be placed in the box and takes one of $\{1,2,3,4,5,6\}$. According to the random permutations of the packing order, the total number of genes in the first chromosome is 15. In the second chromosome, the genes are encoded in random orientations of the boxes, as shown in Figure 4.5.

Costumers	1						2				3				
Boxes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Packing Orders	3	2	1	4	6	5	10	8	7	9	11	12	15	14	13
Orientations	1	1	3	4	1	2	6	5	1	4	1	2	3	1	2

Figure 6.10 Encoding representation of the packing and orientation chromosomes

As a construction heuristic, a corner placement variant of the deep-bottom-left-fill (DBLF) heuristic method of Karabulut & İnceoğlu (2004) is adapted to the algorithm. The DBLF heuristic ensures the stability of the cargo while each box is entirely supported by the container or other boxes with fragility control. Loading the vehicle from the rear is the standard approach in the literature. A common and practical method in transportation is that when a customer is visited, the goods belonging to the customers following each other along the route can be unloaded from the rear of the vehicle with forklifts without the need to move them (Fuellerer, Doerner, Hartl, & Iori, 2009).

The proposed search method generates a random initial population; then, the single-point crossover operator (Figure 4.6) is applied to the selected parents to generate new candidate solutions. Crossover operators are used to transfer the packing orders from parents to offspring. Then, a swapping mutation operator (Figure 4.7) is applied to the output of the crossover operator. The box types and other specifications, such as orientations, are transferred to offspring. Selection and replacement procedures are performed with the SWO approach proposed in this thesis. These processes continue until the specified termination condition has been met.

Costumers	1						2				3				
Boxes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Parent 1	3	2	1	4	6	5	8	7	9	10	11	15	13	14	12
Parent 2	2	3	4	6	1	5	10	9	7	8	15	14	11	13	12
Offspring 1	3	2	1	4	6	5	8	7	9	10	15	14	11	13	12
Offspring 2	2	3	4	6	1	5	10	9	8	7	11	15	13	14	12

Figure 6.11 Single-point crossover operator

Costumers	1						2				3				
Boxes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Offspring 2	2	3	**4**	6	1	**5**	10	9	8	7	11	15	13	14	12

$\Downarrow$

Offspring 2	2	3	**5**	6	1	**4**	10	9	8	7	11	15	13	14	12

Figure 6.12 Swapping mutation

All previous loading sequences of requests are stored in a cache to enhance search speed. Before being searched, arrays are first checked within the cache. To expedite the search, a dedicated table is created for each request that details the specific locations of all stored arrays, which includes the current request. For example, if box sets of customers 5, 3, and 6 to be visited respectively could be packed in the vehicle, they are saved as $BS_{5\_3\_6} = true$, otherwise as $BS_{5\_3\_6} = false$. We call these parts of requests the request thread (RT).

Thus, a RT is located solely by inspecting the cache locations of its initial request. The packing algorithm will only be called upon if the array is not present in cache, and subsequently the array is appended to the cache together with the outcome of the packing test. This ensures that the duplicated request threads are not checked repeatedly.

6.4.4 *The framework of the proposed algorithm for the 3L-PDP*

The main steps of the elitism-based ES algorithm for the 3L-PDP are given in Algorithm 6.2. S_{best} is the best solution found. It is obtained after two main procedures: the routing and loading stages. The first part consists of solving the PDP. The input parameters $iterMax$ and $iterMax'$ represent the maximum number of iterations of the EA. SWO parameters are also user-specified input data. To speed up the search process, an empty directory is created to save the information and availability of the loaded box sets. In this directory, the box sets belonging to the customers that are transported together are saved by specifying whether the installation was successful or not, using the variables including the customer number. For example, if box sets of customers 5, 3, and 6 to be visited respectively could be packed in the vehicle, they are saved as $BS_{5\_3\_6} = true$, otherwise as $BS_{5\_3\_6} = false$.

Table 6.3 Notation of Algorithm 6.2

it	: Iteration number
$iterMax$	: Maximum iteration number
S_{best}	: The best solution
$BS\ memory$	: Directory created to store the status of box sets belonging to customers that passed the loading test
$Packing\_ok$	: Binary variable for packing feasibility
$SWO\ \ value$	: Elitism ratio according to iteration ($F(it)$)
I_{max}	: Maximum vehicle number

The routing stage begins with the random generation of the initial population. Then, iterations continue until the maximum number is reached. In each iteration, the degree of elitism ($F(it)$) is determined, and accordingly, parent selection and offspring production are performed. The best solution obtained is an input to the next stage for packing check. In the packing control section, the developed algorithm is applied in the same way. At each pick-up point, the loads on the vehicle are placed in line with the constraints considered. If a feasible solution is obtained, the iterations are stopped, and the same process is started for the next pick-up point. If no feasible solution is found until the $iterMax'$ number, the search process at the pick-up points is terminated, and the search is restarted in the routing section. In the following iterations, penalty points are added for the unfeasible ones by using the solutions in memory, and recalculations are prevented for the feasible ones.

Algorithm 6.2 *Main steps of the Elitism-based ES for the 3L-PDP*

```
Input: instance data of the 3L-PDP, SWO parameters, BS memory,
Output: S_best ←Best solution
While not a feasible solution is found, do
    Generate μ initial population (G0) for the PDP part of the problem
    it← 0
    While it ≤ iterMax, do
        Obtain SWO value, F(it)
        Select ρ parents
        Reproduction and mutation λ offspring
        Evaluate offspring with BS… in memory
        Replace (λ-ρ) offspring while retaining the elites
        Sort individuals by their scores
        It←it+1
    End While
    Packing_ok← true, k← 0,
    While k ≤ I_max and Packing_ok, do
        Initialize load data at I_k , BS…
        Generate μ' initial population (G0) for loading part of the problem
        It'← 0
        While it' ≤ iterMax', do
                Obtain SWO value F(it')
                Select ρ' parents
                Reproduction and mutation λ offspring
                Evaluate offspring (S_k')
                Replace of (λ'-ρ') offspring while retaining the elites
                Sort individuals by their scores, it'←it'+1
                If S'_k_best is feasible
                        it'← iterMax'
                Else
                        Packing_ok← false
                End if
        End While
        Save BS… to the directory
    End While
End While
```

6.5 Computational Experiments

For solving the 3L-PDP, Elitism-based SWO algorithms were coded in C# using Microsoft Visual Studio 2019. All problem instances were solved on an Intel Core i5-7400 processor at 3.0 GHz using 12 GB of RAM. The algorithm was first tested with the one-dimensional PDP samples proposed by Li & Lim (2001). It was then tested with Männel & Bortfeldt's (2016) benchmark instances of the 3L-PDP. These tests were done on three variants of the problem. Lastly, since there is no test set in which Weight Constraint (WC), Orientation Constraint (OR), Stability (Support) Constraint (ST), Stacking Constraint (FR), and Weight Distribution (WD) will be handled simultaneously, an appropriate sample set was prepared and applied.

All tests were performed with 10000 iterations and parameter settings obtained in Section 5.2.

6.5.1 Li & Lim (2001) Benchmark Instances

In order to compare with other algorithms using Li & Lim (2001) instances, the loading phase of the algorithm has been canceled. The primary objective of this PDPTW is to minimize the number of vehicles used; the secondary objective is to minimize the total travel distance. Geographical data is generated randomly in the LR1 and LR2 problem sets, while it is clustered in the LC1 and LC2 problem sets. The LRC1 and LRC2 problem sets feature a combination of random and clustered structures. LR1, LC1, and LRC1 problem clusters have a limited planning horizon and can only accommodate a small number of clients per route. Conversely, the LR2, LC2, and LRC2 sets have an extended planning horizon, which enables a single vehicle to serve numerous customers.

Li and Lim (2001) instances were tested with 50 and 100 requests, and the results of the proposed algorithm were compared with the results of Li and Lim (2001) and the Best-Known Solutions (BKN) for the PDPTW from the website (https://www.sintef.no/projectweb/top/pdptw/). This is the Sintef website featuring the Li and Lim instances, which present the best results achieved through different solution approaches.

The results of Li and Lim instances (50 requests) are shown in Table 6.4. The proposed algorithm and the best solutions obtained by Li & Lim (2001), Ropke and Pisinger (2006), and BKN solutions are presented, where nv is the number of vehicles used, $tdist$ is the total travel distance, and the best values are in bold.

The gap value that displays the percentile deviation from the BKN is computed as per Eq. ((6.2). For a single instance, where $tdist\_SWO$ represents the best-achieved travel distance by the proposed algorithm, and $tdist\_BKN$ represents the best-known travel distance for the given instance.

Table 6.4 The results of the Li and Lim benchmark instances (50 requests)

#ins	BKN		Li & Lim (2001)			Ropke and Pisinger (2006)			SWO		
	nv	tdist	nv	tdist	%gap	nv	tdist	%gap	nv	tdist	%gap
lc101s	10	828.94	10	828.94	0.00	10	828.94	0	10	828.94	0.00
lc102s	10	828.94	10	828.94	0.00	10	828.94	0	10	828.94	0.00
lc103s	9	1035.35	10	827.86		9	1035.35	0	9	1068.68	3.22
lc104	9	860.01	9	861.95	0.23	9	860.01	0	9	860.01	0.00
lc105s	10	828.94	10	828.94	0.00	10	828.94	0	10	828.94	0.00
lc106s	10	828.94	10	828.94	0.00	10	828.94	0	10	828.94	0.00
lc107s	10	828.94	10	828.94	0.00	10	828.94	0	10	828.94	0.00
lc108s	10	826.44	10	826.44	0.00	10	826.44	0	10	826.44	0.00
lc109q	9	1000.60	10	827.82		9	1000.60	0	9	1096.48	9.58
lc201s	3	591.56	3	591.56	0.00	3	591.56	0	3	591.56	0.00
lc202s	3	591.56	3	591.56	0.00	3	591.56	0	3	591.56	0.00
lc203*	3	591.17	3	591.17	0.00	3	591.17	0	3	591.17	0.00
lc204	3	590.60	3	591.17	0.10	3	590.60	0	3	590.60	0.00
lc205s	3	588.88	3	588.88	0.00	3	588.88	0	3	588.88	0.00
lc206s	3	588.49	3	588.49	0.00	3	588.49	0	3	588.49	0.00
lc207s	3	588.29	3	588.29	0.00	3	588.29	0	3	588.29	0.00
lc208s	3	588.32	3	588.32	0.00	3	588.32	0	3	588.32	0.00
lr101s	19	1650.80	19	1650.80	0.00	19	1650.80	0	19	1650.80	0.00
lr102s	17	1487.57	17	1487.57	0.00	17	1487.57	0	17	1487.57	0.00
lr103s	13	1292.68	13	1292.68	0.00	13	1292.68	0	13	1292.68	0.00
lr104s	9	1013.39	9	1013.39	0.00	9	1013.39	0	9	1013.39	0.00
lr105s	14	1377.11	14	1377.11	0.00	14	1377.11	0	14	1377.11	0.00
lr106s	12	1252.62	12	1252.62	0.00	12	1252.62	0	12	1252.62	0.00
lr107s	10	1111.31	10	1111.31	0.00	10	1111.31	0	10	1111.31	0.00
lr108s	9	968.97	9	968.97	0.00	9	968.97	0	9	968.97	0.00
lr109	11	1208.96	11	1239.96	2.56	11	1208.96	0	11	1208.97	0.00
lr110s	10	1159.35	10	1159.35	0.00	10	1159.35	0	10	1159.35	0.00
lr111s	10	1108.90	10	1108.90	0.00	10	1108.90	0	10	1108.90	0.00
lr112s	9	1003.77	9	1003.77	0.00	9	1003.77	0	9	1003.77	0.00
lr201	4	1253.23	4	1263.23	0.80	4	1253.23	0	4	1253.23	0.00
lr202s	3	1197.67	3	1197.67	0.00	3	1197.67	0	3	1197.67	0.00
lr203s	3	949.40	3	949.40	0.00	3	949.40	0	3	953.93	0.48
lr204s	2	849.05	2	849.05	0.00	2	849.05	0	2	849.05	0.00
lr205s	3	1054.02	3	1054.02	0.00	3	1054.02	0	3	1054.02	0.00
lr206s	3	931.63	3	931.63	0.00	3	931.63	0	3	931.63	0.00
lr207s	2	903.06	2	903.06	0.00	2	903.06	0	2	903.06	0.00
lr208s	2	734.85	2	734.85	0.00	2	734.85	0	2	734.85	0.00
lr209	3	930.59	3	937.59	0.75	3	930.59	0	3	1021.03	9.72
lr210s	3	964.22	3	964.22	0.00	3	964.22	0	3	964.22	0.00
lr211	2	911.52	2	927.80	1.79	2	911.52	0	3	918.06	0.72
lrc101s	14	1708.80	14	1708.80	0.00	14	1708.80	0	14	1708.80	0.00
lrc102	12	1558.07	13	1563.07		12	1558.07	0	12	1558.07	0.00
lrc103s	11	1258.74	11	1258.74	0.00	11	1258.74	0	11	1258.74	0.00
lrc104s	10	1128.40	10	1128.40	0.00	10	1128.40	0	10	1128.40	0.00
lrc105s	13	1637.62	13	1637.62	0.00	13	1637.62	0	13	1637.62	0.00
lrc106	11	1424.73	11	1424.73	0.00	11	1424.73	0	11	1424.73	0.00
lrc107	11	1230.14	11	1230.14	0.00	11	1230.14	0	11	1230.15	0.00
lrc108	10	1147.43	10	1147.43	0.00	10	1147.43	0	10	1147.43	0.00
lrc201	4	1406.94	4	1468.94	4.41	4	1406.94	0	4	1406.94	0.00
lrc202s	3	1374.27	3	1374.27	0.00	3	1374.27	0	4	1487.64	
lrc203s	3	1089.07	3	1089.07	0.00	3	1089.07	0	3	1090.21	0.11
lrc204	3	818.66	3	827.83	1.12	3	818.66	0	3	818.66	0.00
lrc205s	4	1302.20	4	1302.20	0.00	4	1302.20	0	4	1302.20	0.00
lrc206	3	1159.03	3	1162.91	0.33	3	1159.03	0	3	1159.07	0.00
lrc207	3	1062.05	3	1424.60	34.14	3	1062.05	0	3	1064.82	0.26
lrc208s	3	852.76	3	852.76	0.00	3	852.76	0	3	852.76	0.00
Average	7.18	1036.78	7.23	1039.08	0.87	7.18	1036.78	0.00	7.21	1042.99	0.44

$$\%gap = 100x\left(\frac{tdist\_SWO - tdist\_BKN}{tdist\_BKN}\right) \tag{6.2}$$

The proposed SWO algorithm finds nearly the same results as BKN for all 56 instances, which include those obtained mostly by exact methods. As the results show, the SWO implementation achieves the solution quality of the best available methods for the PDPTW. In particular, a higher solution quality was obtained compared to the metaheuristic proposed by Li & Lim (2001).

The results of Li and Lim instances (100 requests) are shown in Table 6.5. The proposed algorithm and the best solutions obtained by Ropke and Pisinger (2006), Männel & Bortfeldt (2016), and BKN are presented, where nv is the number of vehicles used, $tdist$ is the total travel distance, and the best values are in bold.

Table 6.5 The results of the Li and Lim benchmark instances (100 requests)

#ins	BKN		Männel & Bortfeldt (2016)			Ropke and Pisinger (2006)			SWO		
	nv	tdist	nv	tdist	%gap	nv	tdist	%gap	nv	tdist	%gap
LC1_2_1	**20**	**2704.57**	**20**	**2704.57**	0.00	**20**	**2704.57**	0.00	**20**	**2704.57**	0.00
LC1_2_2	**19**	**2764.56**	**19**	**2764.56**	0.00	**19**	**2764.56**	0.00	19	2768.95	0.16
LC1_2_3	**17**	**3127.78**	17	3128.61	0.03	17	3128.61	0.03	18	2796.11	
LC1_2_4	**17**	**2693.41**	**17**	**2693.41**	0.00	**17**	**2693.41**	0.00	17	2712.54	0.71
LC1_2_5	**20**	**2702.05**	**20**	**2702.05**	0.00	**20**	**2702.05**	0.00	**20**	**2702.05**	0.00
LC1_2_6	**20**	**2701.04**	**20**	**2701.04**	0.00	**20**	**2701.04**	0.00	**20**	**2701.04**	0.00
LC1_2_7	**20**	**2701.04**	**20**	**2701.04**	0.00	**20**	**2701.04**	0.00	20	2703.81	0.10
LC1_2_8	**19**	**3354.27**	20	2689.83		20	2689.83		20	2691.98	
LC1_2_9	**18**	**2724.24**	18	2724.24	0.00	18	2724.24	0.00	18	2967.68	8.94
LC1_2_10	**17**	**2942.13**	17	2943.49	0.05	17	2943.49	0.05	18	3108.14	5.64
LC2_2_1	**6**	**1931.44**	**6**	**1931.44**	0.00	**6**	**1931.44**	0.00	**6**	**1931.44**	0.00
LC2_2_2	**6**	**1881.40**	**6**	**1881.40**	0.00	**6**	**1881.40**	0.00	8	2481.41	
LC2_2_3	**6**	**1844.33**	6	1844.33	0.00	**6**	**1844.33**	0.00	7	2229.97	
LC2_2_4	**6**	**1767.12**	**6**	**1767.12**	0.00	**6**	**1767.12**	0.00	7	2631.99	
LC2_2_5	**6**	**1891.21**	**6**	**1891.21**	0.00	**6**	**1891.21**	0.00	**6**	**1892.00**	0.04
LC2_2_6	**6**	**1857.78**	**6**	**1857.78**	0.00	**6**	**1857.78**	0.00	6	1858.88	0.06
LC2_2_7	**6**	**1850.13**	**6**	**1850.13**	0.00	**6**	**1850.13**	0.00	6	1863.21	0.71
LC2_2_8	**6**	**1824.34**	**6**	**1824.34**	0.00	**6**	**1824.34**	0.00	6	1846.91	1.24
LC2_2_9	**6**	**1854.21**	**6**	**1854.21**	0.00	**6**	**1854.21**	0.00	7	1878.84	
LC2_2_10	**6**	**1817.45**	**6**	**1817.45**	0.00	**6**	**1817.45**	0.00	6	1869.08	2.84
LR1_2_1	**20**	**4819.12**	**20**	**4819.12**	0.00	**20**	**4819.12**	0.00	**20**	**4819.12**	0.00
LR1_2_2	**17**	**4621.21**	**17**	**4621.21**	0.00	**17**	**4621.21**	0.00	18	4849.96	4.95
LR1_2_3	**14**	**4402.38**	15	3612.64		15	3612.64		16	3938.21	
LR1_2_4	**10**	**3027.06**	10	3037.38	0.34	10	3037.38	0.34	12	3141.59	3.78
LR1_2_5	**16**	**4760.18**	**16**	**4760.18**	0.00	**16**	**4760.18**	0.00	17	5336.58	12.11
LR1_2_6	**13**	**4800.94**	14	4175.16		14	4178.24		16	4270.12	
LR1_2_7	**12**	**3543.36**	12	3550.61	0.20	12	3550.61	0.20	14	3556.77	0.38
LR1_2_8	**9**	**2759.32**	9	2784.53	0.91	9	2784.53	0.91	9	2904.12	5.25
LR1_2_9	**13**	**5050.75**	14	4354.66		14	4354.66		16	4405.44	
LR1_2_10	**11**	**3664.08**	11	3714.16	1.37	11	3714.16	1.37	13	3788.53	3.40

Table 6.5 Continues

#ins	BKN		Männel & Bortfeldt (2016)			Ropke and Pisinger (2006)			SWO		
	nv	tdist	nv	tdist	%gap	nv	tdist	%gap	nv	tdist	%gap
LR2_2_1	**5**	**4073.10**	**5**	**4073.10**	0.00	**5**	**4073.10**	0.00	5	5894.15	
LR2_2_2	**4**	**3796.00**	**4**	**3796.00**	0.00	**4**	**3796.00**	**0.00**	5	5752.74	
LR2_2_3	**4**	**3098.36**	**4**	**3098.36**	0.00	**4**	**3098.36**	**0.00**	5	4068.86	
LR2_2_4	**3**	**2486.00**	**3**	**2486.14**	0.01	**3**	**2486.14**	**0.00**	4	3172.02	
LR2_2_5	**4**	**3438.39**	**4**	**3438.39**	0.00	**4**	**3438.39**	**0.00**	4	4180.55	21.58
LR2_2_6	**3**	**4457.95**	4	3201.54		4	3201.54		5	4481.08	0.52
LR2_2_7	**3**	**3098.35**	**3**	3135.05	1.18	3	3135.05	1.18	3	3222.09	3.99
LR2_2_8	**2**	**2449.36**	2	2555.40	4.33	2	2555.40	4.33	2	2461.63	0.50
LR2_2_9	**3**	**3922.11**	3	3930.49	0.21	3	3930.49	0.21	5	4050.77	3.28
LR2_2_10	**3**	**3254.83**	3	3323.37	2.11	3	3344.08	2.74	4	3221.91	
LRC1_2_1	**19**	**3606.06**	**19**	**3606.06**	0.00	19	3606.06	0.00	19	3847.97	6.71
LRC1_2_2	**15**	**3671.02**	**15**	**3671.02**	0.00	15	3674.80	0.10	17	3622.22	
LRC1_2_3	**13**	**3154.92**	13	3161.75	0.22	13	3178.17	0.74	15	3666.89	
LRC1_2_4	**10**	**2631.82**	**10**	**2631.82**	0.00	**10**	**2631.82**	0.00	11	3140.67	
LRC1_2_5	**16**	**3715.81**	**16**	**3715.81**	0.00	**16**	**3715.81**	0.00	16	4040.14	8.73
LRC1_2_6	**16**	**3572.16**	**16**	**3572.16**	0.00	**17**	**3368.66**		17	3562.95	
LRC1_2_7	**14**	**3666.34**	14	3668.39	0.06	14	3668.39	0.06	17	3602.42	
LRC1_2_8	**13**	**3145.74**	13	3146.70	0.03	13	3174.55	0.92	14	3218.65	2.32
LRC1_2_9	**13**	**3157.34**	**13**	**3157.34**	0.00	13	3226.72	2.20	15	3182.61	0.80
LRC1_2_10	**12**	**2928.90**	12	2951.29	0.76	12	2951.29	0.76	13	3060.25	
LRC2_2_1	**6**	**3595.18**	**6**	**3595.18**	0.00	6	3605.40	0.28	7	3722.51	
LRC2_2_2	**5**	**3158.25**	5	3327.18	5.35	5	3327.18	5.35	6	2793.19	
LRC2_2_3	**4**	**2881.99**	4	2938.28	1.95	4	2938.28	1.95	5	2972.56	3.14
LRC2_2_4	**3**	**2835.40**	3	2887.97	1.85	3	2887.97	1.85	4	2574.68	
LRC2_2_5	**5**	**2776.93**	5	2776.93	0.00	5	2776.93	0.00	5	3029.15	9.08
LRC2_2_6	**5**	**2707.96**	**5**	**2707.96**	0.00	**5**	**2707.96**	0.00	5	2879.44	6.33
LRC2_2_7	**4**	**3010.68**	4	3044.40	1.12	4	3056.09	1.51	4	3065.55	1.82
LRC2_2_8	**4**	**2399.89**	**4**	**2399.95**	0.00	**4**	**2399.95**	0.00	5	2667.93	
LRC2_2_9	**4**	**2208.49**	**4**	**2208.49**	0.00	**4**	**2208.49**	0.00	4	2292.50	3.80
LRC2_2_10	**3**	**2437.88**	3	2550.56	4.62	3	2550.56	4.62	34	2134.16	
Average	10.00	3062.00	10.08	3007.65	0.49	10.10	3006.98	0.59	11.35	3215.55	3.51

Table 6.6 summarizes the mean results of benchmark instances with 50 and 100 requests, as presented by Li & Lim (2001). The proposed SWO algorithm has gap values of 0.44% and 3.51%, while Ropke and Pisinger (2006) obtained identical results for the former and a gap value of 0.59% for the latter. Li and Lim (2001) obtained a gap value of 0.87%, and for large benchmark instances, Männel & Bortfeldt (2016) achieved a gap value of 0.49%. The SWO algorithm proposed yields outcomes almost identical to those of BKN across all cases, incorporating those obtained using highly competitive approaches in the literature. Based on the findings, the SWO mechanism renders the same quality solution as the most superior methods available for PDPTW.

Table 6.6 Summary results of the Li and Lim (2001) instances

	Li and Lim (50 requests) instances			Li and Lim (100 requests) instances		
	a_nv	**a_tdist**	**a_gap**	**a_nv**	**a_tdist**	**a_gap**
BKN	7.18	1036.78		10.00	3062.00	
Li & Lim (2001)	7.23	1039.08	0.87			
Ropke and Pisinger(2006)	7.18	1036.78	0.00	10.10	3006.98	0.59
Männel & Bortfeldt (2016)				10.08	3007.65	0.49
SWO	7.21	1042.99	0.44	11.35	3215.55	3.51

a_nv: average number of vehicle used, a_tdist: average total distance, a_gap: average gap

6.5.2 Männel & Bortfeldt (2016) benchmark instances

Männel & Bortfeldt (2016) generated the following instances to provide a sufficiently large set of the 3L-PDP benchmarks, including 54 instances with different characteristics. Concerning the distribution of claims and delivery locations, they created three variants: "Random," "Mixed set," and "Pure set" (see Table 6.7).

Table 6.7 Characteristics of the Männel & Bortfeldt (2016) benchmark instances

Number of requests	2 boxes per request on average			3 boxes per request on average			**Total**
	Random (RAND)	**Mixed cluster (CPCD)**	**Pure cluster (CLUS)**	**Random (RAND)**	**Mixed cluster (CPCD)**	**Pure cluster (CLUS)**	
50	5	5	5	5	5	5	30
75	3	3	3	3	3	3	18
100	1	1	1	1	1	1	6

In the "Random" variant, customers are neatly dispersed in a rectangular section of the field, while in other variants, they are clustered. In the "Mixed cluster" variant, individual clusters contain pick-up and delivery points, while in a separate cluster of the "Pure clusters" variant, only one type of customer group is formed.

Loading spaces and boxes are generated in a manner similar to Gendreau et al. (2006). The dimensions of the vehicles' uniform loading spaces have been selected as $L = 60, W = 25, and\ H = 30$ length units. The dimensions of the boxes, including length, width, and height, are selected randomly from the intervals $[0.2L, 0.6L], [0.2W, 0.6W], and\ [0.2H, 0.6H]$.

The loading space weight capacity has been set to 45,000 weight units, and this is also the volume capacity. Box requirements for one-third of the instances are chosen with a weight and volume proportion of three to one, while identical values are chosen for weights and volumes for the remaining requests.

The maximum number of accepted vehicles per instance is determined in such a way that the algorithm can find a feasible solution to the problem relatively quickly. However, in some cases, reducing the number of vehicles may be necessary to minimize the total distance traveled.

Since the reloading ban was not considered in this study, two of the five variants and one-dimensional (1D) version, presented in Männel & Bortfeldt (2016), are used for comparison in this section (Table 6.8).

Table 6.8 The five variants of the 3L-PDP presented in Männel & Bortfeldt (2016)

variants	RS loading	RS unloading	Reloading ban	Independent partial routes	Considered in this chapter
1	y	n	n	n	
2	y	y	n	n	√
3	y	n	y	n	
4	y	y	y	n	
5	y	a	a	y	√

y: yes, n: no, a: automatically

The summarized results for the 3L-PDP instances regarding total distance traveled ($tdist$) are presented in Table 6.9. The names for different variants of instances are listed in the leftmost column. The first comparison column shows the total distances for the one-dimensional test where only the weight and routing constraints are considered. The next pair of columns shows the total distances for tests where classical packing constraints are considered, except for the RS constraint. Finally, the last columns show the results of tests with all constraints considered.

Table 6.9 Summarized results of Männel & Bortfeldt (2016) benchmark instances

Instances		NumOfIns	1d		Variant 2		Variant 5	
			Männel & Bortfeldt (2016)	SWO	Männel & Bortfeldt (2016)	SWO	Männel & Bortfeldt (2016)	SWO
50	CLUS	10	**966.08**	1018.1	1049.46	**1026.54**	1188.64	**1132.24**
50	CPCD	10	**1102.27**	1154.32	1256.14	**1163.45**	1327.12	1341.96
50	RAND	10	**1264.45**	1354.44	1370.71	**1366.71**	1628.35	**1574.12**
75	CLUS	6	**1162.01**	1324.71	**1307.85**	1334.26	1459.69	1517.48
75	CPCD	6	**1831.07**	2082.52	2088.54	**2049.78**	2242.18	2340.2
75	RAND	6	**1638.77**	1889.89	**1812.67**	1896.91	2135.22	**2116.56**
100	CLUS	2	**3191.82**	3741.93	**3700.82**	3877.27	4174.28	4534.96
100	CPCD	2	**3560.57**	3985.69	4177.89	**3927.7**	4289.37	4709.22
100	RAND	2	**3098.21**	3771.92	**3533.16**	3811.67	4055.33	4405.07
		Average	1979.47	2258.17	2255.25	2272.70	2500.02	2630.20

The results on the total travel distance (tdist), as reported by Männel & Bortfeldt (2016), can be found in Table 6.9 and on the following website *"https://data.mendeley.com/datasets/svgff97bn4/1*". Request numbers and instance names are provided in the leftmost column. The subsequent column presents the tdist for the 1D problem with consideration of the Weight Constraint (WC) and the PDP constraints exclusively. The tdist and gaps (Eq. 5.2) for the 3L-PDP 1D, variants 2 and 5, can be found in the following columns.

For the one-dimensional version of the problem and variants 2 and 5, the gap values of SWO are 9.86%, -1.04%, and 0.51%, respectively. Variant 2 showed better outcomes in 33 test instances, according to the SWO results, and the average result was superior to its competitor. For variant 5, the proposed algorithm obtained 21 superior solutions. The study suggests that the proposed algorithm performs exceptionally well compared to previous research in the area of Pickup and Delivery with loading constraint issues. However, it is important to note that the quality of the solutions obtained using SWO decreases as the number of constraints increases, consistent with the findings of Männel and Bortfeldt (2016). Additionally, the results demonstrate that pickup and delivery problems involving three-dimensional loads result in a considerable rise in travel distances. Hence, this idea is commonly more realistic than one-dimensional and less restricted ones.

Table 6.10 Detailed results of Männel & Bortfeldt (2016) benchmark instances

Instances		1D			Variant 2			Variant 5		
		Männel & Bortfeldt (2016)	**Proposed**	**%gap**	**Männel & Bortfeldt (2016)**	**Proposed**	**%gap**	**Männel & Bortfeldt (2016)**	**Proposed**	**%gap**
50	CLUS_2_1	**888.14**	941.71	6.03	979.58	**954.65**	-2.54	1121.46	**1034.64**	-7.74
50	CLUS_2_2	**852.82**	889.59	4.31	927.90	**911.61**	-1.76	1109.68	**986.72**	-11.08
50	CLUS_2_3	**921.30**	965.15	4.76	1011.51	**965.15**	-4.58	1151.12	**1087.74**	-5.51
50	CLUS_2_4	**1031.28**	1094.77	6.16	1117.70	**1087.99**	-2.66	1275.24	**1186.91**	-6.93
50	CLUS_2_5	**1132.02**	1196.79	5.72	1231.75	**1196.79**	-2.84	1379.22	**1321.34**	-4.20
50	CLUS_3_1	**889.57**	901.49	1.34	983.86	**954.25**	-3.01	**1050.41**	1052.16	0.17
50	CLUS_3_2	**854.41**	910.01	6.51	924.67	**903.94**	-2.24	1099.70	**998.55**	-9.20
50	CLUS_3_3	**926.75**	984.82	6.27	997.03	**984.82**	-1.22	1123.87	**1068.74**	-4.91
50	CLUS_3_4	**1026.55**	1095.86	6.75	**1091.72**	1105.39	1.25	1252.35	**1229.49**	-1.83
50	CLUS_3_5	**1137.96**	1200.80	5.52	1228.89	**1200.80**	-2.29	**1323.39**	1356.11	2.47
50	CPCD_2_1	**1102.82**	1142.78	3.62	1259.81	**1142.78**	-9.29	**1378.32**	1386.23	0.57
50	CPCD_2_2	**1039.63**	1126.67	8.37	1168.05	**1089.13**	-6.76	1260.34	**1240.01**	-1.61
50	CPCD_2_3	**996.64**	1023.34	2.68	1140.37	**1029.50**	-9.72	1234.57	1243.69	0.74
50	CPCD_2_4	**1128.00**	1202.68	6.62	1254.35	**1218.20**	-2.88	**1332.65**	1340.17	0.56

Table 6.10 Contınues

	Instances	1D			Variant 2			Variant 5		
		Männel & Bortfeldt (2016)	Proposed	%gap	Männel & Bortfeldt (2016)	Proposed	%gap	Männel & Bortfeldt (2016)	Proposed	%gap
50	CPCD_2_5	**1237.93**	1304.03	5.34	1391.59	**1313.28**	-5.63	**1460.59**	1526.91	4.54
50	CPCD_3_1	**1097.01**	1126.34	2.67	1298.45	**1134.75**	-12.61	1342.60	**1325.14**	-1.30
50	CPCD_3_2	**1048.17**	1078.25	2.87	1222.38	**1112.96**	-8.95	1245.59	**1237.89**	-0.62
50	CPCD_3_3	**998.97**	1046.46	4.75	1140.22	**1044.51**	-8.39	**1252.47**	1254.43	0.16
50	CPCD_3_4	**1135.69**	1208.75	6.43	1283.16	**1242.37**	-3.18	**1313.58**	1367.16	4.08
50	CPCD_3_5	**1237.80**	1283.86	3.72	1403.03	**1307.00**	-6.84	**1450.47**	1497.96	3.27
50	RAND_2_1	**1362.26**	1441.71	5.83	1454.24	**1441.71**	-0.86	1739.10	**1604.67**	-7.73
50	RAND_2_2	**1181.42**	1260.06	6.66	1318.04	**1303.79**	-1.08	**1586.44**	1614.06	1.74
50	RAND_2_3	**1234.98**	1354.89	9.71	**1334.59**	1369.43	2.61	**1666.28**	1707.38	2.47
50	RAND_2_4	**1246.39**	1379.72	10.70	**1367.16**	1380.97	1.01	1585.81	**1498.68**	-5.49
50	RAND_2_5	**1276.39**	1352.03	5.93	**1354.64**	1384.75	2.22	1594.02	**1489.53**	-6.56
50	RAND_3_1	**1359.03**	1429.22	5.16	1467.18	**1446.11**	-1.44	**1715.25**	1759.14	2.56
50	RAND_3_2	**1184.75**	1302.59	9.95	1326.86	**1305.41**	-1.62	1581.58	**1404.37**	-11.20
50	RAND_3_3	**1251.30**	1300.89	3.96	1345.13	**1329.63**	-1.15	1661.71	**1481.54**	-10.84
50	RAND_3_4	**1266.16**	1397.56	10.38	1381.39	**1355.40**	-1.88	1564.35	**1497.72**	-4.26
50	RAND_3_5	**1281.81**	1325.75	3.43	1357.90	**1349.86**	-0.59	**1588.92**	1684.12	5.99
75	CLUS_2_1	**1180.71**	1358.68	15.07	**1315.95**	1332.67	1.27	**1469.57**	1528.56	4.01
75	CLUS_2_2	**1136.71**	1262.03	11.02	**1252.09**	1298.43	3.70	**1425.73**	1525.78	7.02
75	CLUS_2_3	**1163.14**	1333.39	14.64	1342.37	**1340.85**	-0.11	**1495.71**	1492.75	-0.20
75	CLUS_3_1	**1178.62**	1330.76	12.91	**1323.61**	1365.08	3.13	**1455.55**	1556.37	6.93
75	CLUS_3_2	**1138.91**	1315.85	15.54	**1288.53**	1315.49	2.09	**1436.75**	1490.71	3.76
75	CLUS_3_3	**1173.96**	1347.54	14.79	**1324.54**	1353.05	2.15	**1474.84**	1510.72	2.43
75	CPCD_2_1	**1803.87**	2021.88	12.09	**2012.83**	2021.88	0.45	**2237.26**	2312.59	3.37
75	CPCD_2_2	**1792.68**	1954.22	9.01	**2033.39**	2020.07	-0.66	**2226.24**	2239.95	0.62
75	CPCD_2_3	**1879.48**	2072.58	10.27	2141.70	**2068.23**	-3.43	**2283.54**	2472.70	8.28
75	CPCD_3_1	**1815.60**	2378.45	31.00	2114.13	**2067.57**	-2.20	**2253.30**	2313.58	2.68
75	CPCD_3_2	**1810.81**	2016.38	11.35	2070.63	**2018.90**	-2.50	**2195.44**	2299.15	4.72
75	CPCD_3_3	**1883.97**	2051.61	8.90	2158.56	**2102.05**	-2.62	**2257.27**	2403.22	6.47
75	RAND_2_1	**1701.32**	1926.99	13.26	**1876.01**	1955.73	4.25	2118.13	**2050.48**	-3.19
75	RAND_2_2	**1562.31**	1801.65	15.32	**1762.22**	1869.15	6.07	**2130.80**	2321.17	8.93
75	RAND_2_3	**1653.16**	1878.54	13.63	**1839.14**	1860.65	1.17	**2188.07**	2046.64	-6.46
75	RAND_3_1	**1698.67**	1946.84	14.61	**1867.50**	1954.61	4.66	**2146.59**	2229.70	3.87
75	RAND_3_2	**1562.97**	1799.26	15.12	**1745.46**	1823.60	4.48	2081.42	**2007.85**	-3.53
75	RAND_3_3	**1654.16**	1986.07	20.07	**1785.70**	1917.72	7.39	2146.31	**2043.50**	-4.79
100	CLUS_2_1	**3176.76**	3739.54	17.72	**3687.00**	3894.07	5.62	**4193.50**	4653.34	10.97
100	CLUS_3_1	**3206.88**	3744.31	16.76	**3714.63**	3860.48	3.93	**4155.06**	4416.59	6.29
100	CPCD_2_1	**3541.89**	3965.57	11.96	4212.94	**3965.57**	-5.87	**4320.29**	4772.38	10.46
100	CPCD_3_1	**3579.25**	4005.82	11.92	4142.83	**3889.83**	-6.11	**4258.44**	4646.06	9.10
100	RAND_2_1	**3086.28**	3794.82	22.96	**3530.66**	3829.53	8.46	**4070.19**	4349.62	6.87
100	RAND_3_1	**3110.14**	3749.01	20.54	**3535.65**	3793.81	7.30	**4040.47**	4460.53	10.40
Average		1496.67	1667.60	9.86	1682.24	1675.67	-1.04	1879.66	1919.02	0.51

6.5.3 New benchmark instances

In the instances of Männel & Bortfeldt (2016) presented in section 6.5.2, the low number of loads, the lack of a time window, and the weight distribution constraints were considered shortcomings. Since it is declared that the 3L-PDP problem we study considers TW and WD (Table 6.1) constraints in addition to previous work, a new instance set should be prepared that includes customer service start and end times. In addition, the weight of each load must also be determined in order to ensure stability during the loading of a large number of loads per request.

New instances consist of 56 instances derived from the PDP instances by Li & Lim (2001) and the Container Loading Problem instances by Bischoff & Ratcliff (1995). Each instance has 100 customers and up to 5000 boxes. The geographic data in problem sets ky_lr1 and ky_lr 2 are generated randomly, while those in problem sets ky_lc1 and ky_lc2 are clustered. Problem sets ky_lrc1 and ky_lrc2 have a combination of randomly generated and clustered structures. Sets ky_lr1, ky_lc1, and ky_lrc1 have a short scheduling period, while sets ky_lr2, ky_lc2, and ky_lrc2 have a wider scheduling period, enabling multiple customers to be served by the same vehicle. The instances were created concerning the loading volume provided by a 20 ft ISO container defined as dimensions $L = 587, W = 233, and\ H = 220$. All instances have five box types, and various numbers of these boxes are randomly distributed to customers. The weight capacity of the vehicles is determined as 10786 weight units, and this limit is independent of the volume capacity. In addition, fragility and stack direction limits are randomly defined for each box type. The new benchmark instances are provided in the following website and a representation of an instance is given in Table 6.11.

https://data.mendeley.com/datasets/z29ncb2khx/2

Table 6.11 A representation of an instance

Number of requests	Maximum number of routes	Loading space's length (L)	width (W)	height (H)	max. cargo weight	Number of box types						
50	25	587	245	300	10786	5						
#	p_x	p_y	p_tw_s	p_tw_e	p_st	d_x	d_y	d_tw_s	d_tw_e	d_st	n_r	no. of box types
0	40	50	0	1236	0	40	50	0	1236	0	0	0
1	42	66	0	155	90	45	65	749	1068	90	1732	2 4 15 5 13
#	max_d	h_f	med_d	h_f	min_d	h_f	frag	w_b				
1	113	0	92	0	33	1	0	60				

#:Number of Request, p_x:X-coordinate of pickup node, p_y:Y-coordinate of pickup node, p_tw_s:Time Window of pickup node (Start), p_tw_e:Time Window of pickup node (End), p_st:Service time at pickup node, d_x:X-coordinate of delivery node, d_y:Y-coordinate of delivery node, d_tw_s:Time Window of delivery node (Start), d_tw_e:Time Window of delivery node (End), d_st:Service time at delivery node, n_r:Weight of the request, no. of box types :Number of the box types (2 3 22 5 26 : request has two box types, namely type 3 with 22 copies and type 5 with 26 copies)
#types:Number of Types, max_d:Maximum dimension, h_f:Height flag of previous dimension, med_d:Medium dimension, h_f:Height flag of previous dimension, min_d:Minimum dimension, h_f:Height flag of previous dimension, frag:Fragility flag, w_b:Weight of each box

In order to compare the performance of the developed algorithm when different constraints are taken into account, five different variants were defined. As can be seen in Table 6.12, the entry for each variant and each restriction is "√" if the restriction is necessary " " otherwise. Var_5 has the largest number of constraints, and this is the first time this problem has been solved so comprehensively.

Table 6.12 The variants of the proposed algorithm for solving the 3L-PDP with new instances

Variants	**TW**	**WL**	**CPC**	**OR**	**ST**	**FR**	**LIFO/RS**	**WD**
Var_1	√	√						
Var_2	√	√	√					
Var_3	√	√	√	√	√	√		
Var_4	√	√	√	√	√	√	√	
Var_5	√	√	√	√	√	√	√	√

The comprehensive outcomes for the new 3L-PDP instances on total travel distance (tdist) are demonstrated in Table 6.13. The instance names are listed in the leftmost column. Subsequent columns reveal the number of vehicles used (nv_best) and the total travel distances (tdist_best) for var_1 test's best solution. The next column shows the mean total travel distance values obtained from ten runs (a_tdist). In the subsequent columns, details are indicated for the other variants listed in Table 6.2.

Table 6.13 Detailed results of the new KY_3L-PDP benchmark instances

	var_1			var_2			var_3			var_4			var_5		
	nv_best	**tdist_best**	**a_dist**	**nv_best**	**tdist_best**	**a_dist**	**nv_best**	**tdist_best**	**a_dist**	**nv_best**	**tdist_best**	**a_dist**	**nv_best**	**tdist_best**	**a_dist**
ky_lc101_	10	824.71	825.22	10	824.71	826.42	10	847.31	851.27	10	1048.51	1050.90	12	1265.22	1275.36
ky_lc102_	10	837.75	838.84	10	837.75	843.68	10	892.38	904.80	10	931.39	949.95	11	1197.89	1204.12
ky_lc103_	9	842.65	845.77	9	904.11	909.41	9	994.99	1002.17	10	1052.97	1071.24	9	1017.34	1030.29
ky_lc104_	8	836.05	838.82	9	934.89	936.11	9	901.58	904.45	8	899.61	902.53	9	907.34	929.57
ky_lc105_	11	967.83	971.00	11	1030.62	1033.33	11	1097.48	1116.41	11	1125.40	1133.22	12	1184.34	1185.82
ky_lc106_	10	910.72	918.23	10	928.18	929.46	10	931.96	944.28	10	993.05	1012.61	10	1147.86	1151.91
ky_lc107_	10	965.11	971.16	10	1046.29	1056.10	10	1037.12	1038.74	10	1044.07	1049.04	11	1207.06	1238.51
ky_lc108_	10	837.93	842.42	10	869.73	875.29	10	952.27	968.64	10	930.74	941.87	10	1026.27	1055.10
ky_lc109_	9	882.83	890.50	9	956.90	959.07	9	973.96	984.33	9	903.21	918.34	10	919.36	941.80
ky_lc201_	7	863.74	872.14	7	1691.35	1703.50	7	1216.94	1225.46	7	1354.39	1355.48	9	1360.55	1395.94
ky_lc202_	5	883.97	890.19	5	1044.02	1052.43	5	1001.24	1017.81	6	1265.52	1286.62	6	1524.50	1534.28
ky_lc203_	4	771.92	779.53	4	1168.02	1177.49	4	1183.89	1193.40	5	1333.74	1359.94	4	1772.03	1789.71
ky_lc204_	3	765.85	771.33	3	786.46	793.39	3	943.63	943.73	3	1055.14	1061.77	4	1183.86	1203.73
ky_lc205_	5	689.49	692.11	5	1094.22	1098.97	5	900.41	914.47	6	1014.43	1020.37	5	1193.14	1226.85
ky_lc206_	5	723.07	725.91	5	933.23	933.90	5	859.09	867.85	5	1145.21	1148.62	5	1117.56	1122.28
ky_lc207_	4	806.36	813.23	4	1214.09	1220.29	4	1037.40	1055.67	4	1126.35	1131.58	5	968.61	981.22
ky_lc208_	4	722.99	725.54	4	1131.38	1132.46	4	1042.80	1044.55	4	1101.74	1107.06	5	1011.59	1025.90
ky_lr101_	15	1471.97	1475.15	15	1497.87	1504.85	15	1478.58	1490.44	15	1644.61	1674.86	17	1912.62	1927.72
ky_lr102_	14	1412.79	1414.50	14	1456.71	1468.54	14	1509.21	1516.48	14	1571.42	1596.62	15	1686.68	1735.41
ky_lr103_	12	1271.55	1279.61	12	1283.28	1285.41	12	1322.33	1343.76	12	1399.11	1417.97	13	1561.35	1564.40
ky_lr104_	11	1089.74	1099.56	11	1090.54	1097.88	11	1149.01	1163.26	11	1196.60	1203.03	11	1274.71	1296.98
ky_lr105_	16	1376.25	1376.41	16	1465.89	1466.10	16	1436.42	1445.80	16	1513.67	1539.87	15	1461.32	1479.93
ky_lr106_	13	1250.74	1262.21	13	1302.42	1304.83	13	1315.88	1332.08	13	1457.98	1468.69	15	1462.07	1479.14
ky_lr107_	11	1147.39	1149.40	11	1142.99	1148.09	11	1166.64	1172.26	11	1305.77	1325.50	12	1470.95	1514.24
ky_lr108_	11	1036.55	1043.21	11	1103.28	1108.24	11	1141.96	1164.41	11	1231.81	1245.12	11	1272.84	1293.03
ky_lr109_	14	1207.98	1213.27	14	1546.62	1555.64	14	1536.75	1547.33	14	1598.41	1618.29	13	1535.68	1547.70
ky_lr110_	11	1158.38	1161.36	11	1161.75	1162.08	11	1273.21	1288.62	11	1209.53	1219.15	12	1283.21	1318.96
ky_lr111_	11	1061.94	1063.40	11	1226.05	1235.72	11	1169.12	1186.41	11	1258.30	1276.19	12	1362.88	1395.56
ky_lr112_	10	1063.01	1065.49	10	1077.08	1078.73	10	1052.87	1060.55	10	1193.11	1205.93	11	1279.68	1289.38

Table 6.13 Continues

	var_1			var_2			var_3			var_4			var_5		
	nv_best	**tdist_best**	**a_dist**	**nv_best**	**tdist_best**	**a_dist**	**nv_best**	**tdist_best**	**a_dist**	**nv_best**	**tdist_best**	**a_dist**	**nv_best**	**tdist_best**	**a_dist**
ky_lr201_	7	1266.52	1277.71	7	1263.95	1272.28	7	1309.82	1322.20	7	1428.69	1436.68	7	1546.57	1579.76
ky_lr202_	10	1564.22	1578.70	10	1846.88	1848.51	10	1906.13	1941.65	10	1953.71	1989.05	9	1733.24	1743.74
ky_lr203_	6	1400.48	1408.87	6	1536.93	1552.03	6	1571.70	1587.71	6	1682.69	1702.09	5	1615.00	1632.72
ky_lr204_	5	1251.95	1258.77	5	1469.10	1474.01	5	1432.53	1451.47	5	1548.09	1569.39	4	1851.20	1880.81
ky_lr205_	5	1174.10	1185.48	5	1381.01	1383.15	5	1378.36	1380.66	5	1470.48	1490.67	5	1469.26	1496.76
ky_lr206_	6	1233.43	1240.61	6	1279.50	1288.54	6	1360.98	1379.00	6	1485.21	1491.35	5	1500.59	1526.07
ky_lr207_	6	1282.14	1289.32	6	1658.98	1673.30	6	1659.09	1671.39	6	1697.58	1699.96	5	1554.94	1568.06
ky_lr208_	4	1313.45	1318.74	4	1771.88	1771.89	4	1683.32	1696.51	4	1856.62	1880.98	4	1769.56	1821.47
ky_lr209_	6	1188.31	1189.73	6	1332.25	1333.96	6	1418.34	1438.19	6	1501.96	1506.90	5	1368.44	1385.10
ky_lr210_	8	1322.11	1332.02	8	1576.97	1578.40	8	1544.39	1557.35	8	1670.54	1692.17	6	1726.92	1772.98
ky_lr211_	4	1061.56	1067.30	4	1115.88	1125.66	4	1142.14	1159.93	4	1281.74	1295.56	4	1369.43	1408.06
ky_lrc101_	16	1589.83	1603.04	16	1641.21	1642.98	16	1654.98	1676.18	16	1761.50	1764.43	16	1890.97	1934.92
ky_lrc102_	16	1779.86	1786.15	16	1802.18	1817.82	16	1776.65	1795.06	16	1776.53	1777.58	15	1756.37	1808.50
ky_lrc103_	12	1300.04	1302.51	12	1406.68	1414.78	12	1385.03	1406.42	12	1474.40	1488.43	12	1375.01	1383.22
ky_lrc104_	9	1015.42	1016.79	9	1030.10	1034.76	9	1044.67	1057.96	10	1191.77	1203.58	10	1325.48	1340.70
ky_lrc105_	13	1473.76	1482.06	13	1479.39	1491.95	13	1488.98	1495.73	13	1653.07	1685.04	13	1750.85	1783.23
ky_lrc106_	14	1444.19	1457.59	14	1630.40	1644.76	14	1642.15	1663.94	14	1821.74	1855.46	14	1525.52	1536.48
ky_lrc107_	11	1163.01	1170.05	11	1165.67	1169.72	11	1238.49	1254.84	11	1328.35	1349.33	11	1478.85	1490.08
ky_lrc108_	12	1237.27	1243.96	12	1412.62	1425.09	12	1366.26	1372.53	12	1660.72	1684.07	12	1486.64	1507.79
ky_lrc201_	9	1582.18	1589.55	9	1817.91	1826.03	9	1842.05	1862.24	9	2113.49	2145.65	9	2818.25	2855.62
ky_lrc202_	7	1623.43	1633.94	7	1702.86	1703.12	7	1656.96	1688.24	7	2002.28	2019.13	7	2389.80	2441.78
ky_lrc203_	4	1263.19	1265.87	4	1401.59	1407.00	5	1344.03	1349.04	4	1667.29	1672.52	4	1684.72	1709.84
ky_lrc204_	4	856.35	859.96	4	1279.70	1279.76	4	1327.83	1338.67	4	1312.98	1331.29	4	1535.54	1540.68
ky_lrc205_	7	1312.09	1322.50	7	1644.23	1647.43	7	1466.16	1480.01	7	1972.93	1986.87	6	2519.59	2550.36
ky_lrc206_	7	1355.02	1358.35	7	1708.91	1723.79	7	1903.85	1917.33	7	1905.89	1960.96	7	2226.07	2229.43
ky_lrc207_	6	1400.17	1407.68	8	2056.68	2070.27	9	1997.58	2018.50	8	2195.84	2240.48	6	2334.30	2343.95
ky_lrc208_	5	1165.07	1166.43	6	1436.71	1445.56	7	1467.86	1495.80	7	1564.07	1579.49	5	1757.99	1772.72
Averages	**8.79**	**1148.22**	**1154.09**	**8.86**	**1314.30**	**1320.43**	**8.91**	**1310.30**	**1324.07**	**8.95**	**1426.53**	**1442.70**	**9.00**	**1516.60**	**1538.92**

Summarized results for the average best total travel distance $(tdist)$ and gap values over the instance groups are presented in Table 6.14. Instances group names are listed in the leftmost column. The subsequent column displays the total travel distances for the one-dimensional PDP (Var_1) test, where exclusively the weight constraint (WL) and routing constraints have been taken into account. The ensuing columns enumerate the total travel distances for the different variants. All presented total travel distances are mean values over ten runs. The following equation calculates the corresponding gaps;

$$gap(\%) = \frac{tdist\ -\ tdist_{var\_1}}{tdist_{var\_1}} 100 \tag{6.3}$$

Table 6.14 Summarized results of the new KY_3L-PDP benchmark instances

	Var_1	Var_2	gap(%)	Var_3	gap(%)	Var_4	gap(%)	Var_5	gap(%)
ky_lc1	878.40	925.91	5.41	947.67	7.89	992.11	12.94	1096.96	24.88
ky_lc2	778.42	1132.85	45.53	1023.17	31.44	1174.57	50.89	1266.48	62.70
ky_lr1	1212.36	1279.54	5.54	1296.00	6.90	1381.69	13.97	1463.66	20.73
ky_lr2	1278.02	1475.76	15.47	1491.53	16.71	1597.94	25.03	1591.38	24.52
ky_lrc1	1375.42	1446.03	5.13	1449.65	5.40	1583.51	15.13	1573.71	14.42
ky_lrc2	1319.69	1631.07	23.60	1613.29	22.25	1929.18	46.18	2070.95	56.93
Average	1140.39	1315.19	16.78	1303.55	15.10	1426.53	27.36	1516.60	34.03

In summary, solving the 3L-PDP instances instead of the corresponding one-dimensional PDP instances leads to a significant increase in total travel distances. On average, across all problem variants, the total travel distances increase by 16.78% and 15.10%, respectively, compared to the one-dimensional variant. The increase is even more pronounced in problem variants 4 and 5, with mean increases of 27.36% and 34.03%, respectively.

Additionally, only option 1, the classical elitism approach stated in section 5.2, was used to solve the new instance set. The SWO approach, which was newly developed, was not employed. Furthermore, the tests were executed without the memory capability of retaining previous solutions, a feature created to decrease the number of insignificant operations and elaborated in section 5.4.

Table 6.15 The further analysis of the proposed solution method (SWO)

Instances	SWO		Option 1				without memory			
	tdist	CPU time	tdist	%gap_ tdist	CPU time	%gap_ time	tdist	%gap_ tdist	CPU time	%gap_ time
ky_lc101_	1265.22	1193.24	1517.51	19.94	1083.74	10.10	1438.21	13.67	1297.10	-8.01
ky_lc102_	1197.89	1012.50	1313.69	9.67	880.43	15.00	1201.40	0.29	983.17	2.98
ky_lc103_	1017.34	951.59	1108.79	8.99	1015.30	-6.28	1199.49	17.90	1003.38	-5.16
ky_lc104_	907.34	930.00	978.20	7.81	1139.99	-18.42	922.63	1.69	1399.90	-33.57
ky_lc105_	1184.34	1165.14	1389.74	17.34	1294.64	-10.00	1406.55	18.76	1177.08	-1.01
ky_lc106_	1147.86	1236.88	1424.90	24.14	1402.39	-11.80	1207.88	5.23	1334.71	-7.33
ky_lc107_	1207.06	1222.58	1239.40	2.68	980.67	24.67	1386.23	14.84	1538.54	-20.54
ky_lc108_	1026.27	1124.33	1143.40	11.41	1135.18	-0.96	1185.37	15.50	1172.30	-4.09
ky_lc109_	919.36	1050.58	979.99	6.60	1147.67	-8.46	1105.37	20.23	1446.13	-27.35
ky_lc201_	1360.55	1131.97	1402.97	3.12	1333.43	-15.11	1480.40	8.81	1447.43	-21.79
ky_lc202_	1524.50	1254.69	1887.21	23.79	1189.97	5.44	1725.71	13.20	1430.11	-12.27
ky_lc203_	1772.03	1266.28	2114.23	19.31	1342.16	-5.65	1776.26	0.24	1101.84	14.92
ky_lc204_	1183.86	1198.75	1286.28	8.65	1097.09	9.27	1232.99	4.15	1274.44	-5.94
ky_lc205_	1193.14	1276.76	1325.42	11.09	1482.43	-13.87	1473.16	23.47	1282.35	-0.44
ky_lc206_	1117.56	1274.17	1323.72	18.45	1259.64	1.15	1212.04	8.45	1377.39	-7.49
ky_lc207_	968.61	1240.93	1083.72	11.88	1178.72	5.28	1166.59	20.44	1356.07	-8.49
ky_lc208_	1011.59	1274.48	1260.24	24.58	1510.47	-15.62	1119.20	10.64	1285.41	-0.85
ky_lr101_	1912.62	1066.51	1965.70	2.78	1102.60	-3.27	1955.68	2.25	1682.77	-36.62
ky_lr102_	1686.68	1336.15	1955.47	15.94	1124.33	18.84	1720.67	2.02	1068.18	25.09
ky_lr103_	1561.35	1078.35	1594.51	2.12	1209.44	-10.84	1902.29	21.84	1138.08	-5.25
ky_lr104_	1274.71	1312.18	1342.37	5.31	1072.62	22.33	1300.49	2.02	1222.82	7.31
ky_lr105_	1461.32	1313.23	1739.76	19.05	1099.71	19.42	1462.73	0.10	1108.39	18.48
ky_lr106_	1462.07	1311.88	1535.13	5.00	1389.64	-5.60	1710.91	17.02	1106.50	18.56
ky_lr107_	1470.95	1233.58	1472.46	0.10	1461.86	-15.62	1779.00	20.94	1419.99	-13.13
ky_lr108_	1272.84	1230.48	1517.84	19.25	1112.01	10.65	1508.18	18.49	1490.92	-17.47
ky_lr109_	1535.68	1384.83	1700.17	10.71	1408.65	-1.69	1886.38	22.84	1315.90	5.24
ky_lr110_	1283.21	1210.72	1536.37	19.73	1592.76	-23.99	1529.58	19.20	1581.73	-23.46
ky_lr111_	1362.88	1440.88	1643.20	20.57	1365.01	5.56	1465.02	7.49	1609.44	-10.47
ky_lr112_	1279.68	1136.94	1334.00	4.24	1195.75	-4.92	1386.47	8.35	1708.53	-33.46
ky_lr201_	1546.57	1250.53	1753.42	13.37	1321.07	-5.34	1817.15	17.50	1377.97	-9.25
ky_lr202_	1733.24	1231.48	2055.58	18.60	1448.34	-14.97	1906.06	9.97	1215.56	1.31
ky_lr203_	1615.00	1222.44	1716.16	6.26	1243.83	-1.72	2009.08	24.40	1258.78	-2.89
ky_lr204_	1851.20	1334.00	2133.90	15.27	1451.63	-8.10	1957.69	5.75	1684.16	-20.79
ky_lr205_	1469.26	1235.50	1690.17	15.04	1416.28	-12.76	1757.53	19.62	1348.04	-8.35
ky_lr206_	1500.59	1500.67	1595.60	6.33	1168.63	28.41	1682.92	12.15	1643.06	-8.67
ky_lr207_	1554.94	1508.61	1886.97	21.35	1404.27	7.43	1614.55	3.83	1789.83	-15.71
ky_lr208_	1769.56	1224.35	1796.97	1.55	1377.94	-11.15	1950.67	10.23	1272.89	-3.81
ky_lr209_	1368.44	1353.19	1420.72	3.82	1215.55	11.32	1548.45	13.15	1254.61	7.86
ky_lr210_	1726.92	1317.73	2046.13	18.48	1560.66	-15.57	1853.93	7.35	1395.30	-5.56
ky_lr211_	1369.43	1244.00	1372.10	0.19	1650.40	-24.62	1418.36	3.57	1734.69	-28.29
ky_lrc101_	1890.97	1433.37	1937.07	2.44	1651.95	-13.23	1933.71	2.26	1388.46	3.23
ky_lrc102_	1756.37	1477.82	2028.93	15.52	1454.74	1.59	1767.65	0.64	1282.86	15.20
ky_lrc103_	1375.01	1293.16	1532.53	11.46	1475.36	-12.35	1651.32	20.10	1383.39	-6.52
ky_lrc104_	1325.48	1278.79	1569.12	18.38	1746.88	-26.80	1568.25	18.32	1255.22	1.88
ky_lrc105_	1750.85	1437.41	2188.22	24.98	1296.92	10.83	2114.77	20.79	1615.05	-11.00
ky_lrc106_	1525.52	1475.04	1810.39	18.67	1298.66	13.58	1759.60	15.34	1325.30	11.30
ky_lrc107_	1478.85	1362.91	1742.09	17.80	1451.66	-6.11	1811.52	22.49	1548.45	-11.98
ky_lrc108_	1486.64	1588.03	1775.91	19.46	1698.87	-6.52	1797.98	20.94	1324.01	19.94
ky_lrc201_	2818.25	1497.80	3515.47	24.74	1746.41	-14.24	2828.82	0.38	1875.56	-20.14
ky_lrc202_	2389.80	1346.56	2433.43	1.83	1613.81	-16.56	2541.00	6.33	1752.64	-23.17
ky_lrc203_	1667.29	1453.01	1769.75	6.15	1723.17	-15.68	1993.10	19.54	1565.55	-7.19
ky_lrc204_	1312.98	1429.59	1471.96	12.11	1509.64	-5.30	1404.01	6.93	1890.21	-24.37
ky_lrc205_	2519.59	1474.84	2705.23	7.37	1759.85	-16.19	3005.54	19.29	1428.59	3.24
ky_lrc206_	1905.89	1585.01	2105.59	10.48	1524.35	3.98	1956.74	2.67	1881.03	-15.74
ky_lrc207_	2195.84	1500.69	2520.12	14.77	1762.82	-14.87	2472.99	12.62	1721.19	-12.81
ky_lrc208_	1757.99	1348.86	2005.54	14.08	1487.39	-9.31	1846.83	5.05	1872.00	-27.95
Averages	**1504.12**	**1290.46**	**1690.99**	**12.41**	**1358.38**	**-3.55**	**1675.31**	**11.81**	**1418.66**	**-7.35**

$$gap\_tdist(\%) = \frac{tdist\ -\ tdist_{SWO}}{tdist_{SWO}} 100 \tag{6.4}$$

$$gap\_time(\%) = \frac{time\ -\ time_{vSWO}}{time_{SWO}} 100 \tag{6.5}$$

All values are averaged across ten runs with 10,000 iterations and across all instances of the same type. In Table 6.15, gap_dist values are presented as percentages of the average total distance traveled per instance type shown in the first column (Eq.(6.4)). In terms of distance traveled, using type 1 elitism and disabling memory caused solutions to be inferior by 12.67% and 11.81%, respectively, in comparison to the SWO approach was not used. The gap-time values are presented as percentages of the CPU time allocated to each instance type, as presented in the first column (Eq.(6.5)). It should be noted that the option 1 type elitism approach, coupled with the deactivation of memory, resulted in -3.55% and -7.35%, respectively, in terms of computation time. It is crucial to observe that disabling memory and non-using SWO decreases the quality of the solution and significantly increases the computation time.

6.6 Chapter Summary

In this chapter, we explained the 3L-PDP. Initially, the MIP formulation has been given for the 3L-PDP based on the previous formulations. Then, the proposed search method, SWO, was implemented on the 1D-PDP benchmark instances of Li & Lim (2001), and it has been observed that it can generate feasible and Best-Known Solutions (BKNs) to one-dimensional PDPs. It achieved the BKN in 46 out of 56 tests and outperformed the heuristic solution in the same instances.

The proposed SWO approach was then tested with benchmark instances from Männel & Bortfeldt (2016). The results show that the developed search algorithm can give high-quality solutions to the instances in reasonable computation times. In order to take into account the TW and WD constraints, the SWO has been implemented for the new benchmark instances derived from the literature. The new benchmark instances provide an excellent opportunity to analyze the algorithm with commonly used constraints, including WD and TW. The computational study shows that the developed algorithm effectively solves the most generalized 3L-PDP versions.

CHAPTER SEVEN
CONCLUSIONS AND OUTLOOK

7.1 Summary of the Thesis

This thesis focused on an integrated approach to vehicle routing and multi-dimensional packing problems. In particular, we have studied various methods to solve integrated versions of routing and three-dimensional loading problems. This integrated problem constitutes a major study area for cost reduction in many transport sectors. It dictates the route determination according to the loading constraints while minimizing the cost and reflects the reality of the load to the computations in the highest way with its three-dimensionality and realistic constraints. In this context, the contributions of this thesis are expected to have both scientific and practical relevance.

Before we explain the integrated routing and loading problems, we have examined two main problems that constitute this problem. For this reason, Chapter 2 provides first both the formulation and literature information about the vehicle routing problem and its extensions. It then provides detailed formulation and literature information on the three-dimensional bin packing problem/ the container loading problem. Furthermore, the significant constraints in loading problems are grouped and explained. The chapter concludes with a brief introduction to routing problems with two- and three-dimensional loading constraints.

Chapter 3 reviews the literature on the Three-Dimensional Loading Capacitated Vehicle Routing Problem (3L-CVRP). In the literature, the inadequacy of CP studies carried out in the context of this problem has been emphasized. The 3L-CVRP combines two distinct problems and stands out for its potential for decomposed and hybrid methods. Additionally, it should be noted that in some studies, despite resolving pickup and delivery constraints, "Time Window" and "Weight Distribution" constraints were not considered. In the context of all these shortcomings and opportunities in the literature, the contributions of this thesis to the literature have been expressed.

The main focus of Chapter 4 is on solution approaches that incorporate exact methods, decomposed CP models, and a hybrid CP solution technique with a metaheuristic approach for the 3L-CVRP. CP-based decomposed models could only generate optimum solutions for small-sized problems. A hybrid model consisting of CP and EA was developed for large-sized problems, and many benchmark instances in the literature were solved, leading to competitive results.

Chapter 5 presents a new solution method that uses an Evolutionary Strategy (ES) based on elitism controlled by the sine function for solving the 3L-CVRP. The elitism approach in this method changes over time depending on a sine function, which is why we named it Sine Wave Optimization (SWO). In the classical elitism approach, elitists are selected from only the best individuals, while in our proposed method, any segment of the population can be determined as elite. In this manner, the search process is balanced between exploration and exploitation. Various variants were created for problem-specific parameter determination and applied to the VRP and CLP benchmark instances, which are frequently solved in the literature. The obtained results were analyzed, and the parameter values giving the best performance were used in the solution of 3L-CVRP with SWO.

Chapter 6 presents the Pickup and Delivery Problem with Three-Dimensional Loading Constraints (3L-PDP). SWO, the solution method mentioned in the previous chapter, was compared to the existing studies in the literature to solve the 3L-PDP. Unlike other routing problems, the 3L-PDP emphasizes the requirement for request sequencing and mandates the LIFO constraint to decrease reloading effort during the loading process. The five variants of the developed model were created by enabling various constraints, and these variants were tested using benchmark instances. Accordingly, the quality of the solution decreased as the number of constraints increased. The findings unambiguously demonstrate that when it comes to pickup and delivery issues, utilizing a three-dimensional model leads to a considerable rise in travel distances, which renders it more realistic overall.

7.2 Future Research Directions

While preparing this thesis, many ideas emerged that can be analyzed in the near future. The loading constraints examined are only some of the constraints that should be considered in practice. Therefore, future studies can enhance compliance with actual conditions by incorporating the difficulties experienced in the loading process into the problem. For example, one issue may be preparing the loading plan when the loading zone and the loaded objects are not rectangular. In order to guarantee the feasibility of solutions for practically all relevant constraints, the integration and application of further constraints may also be of interest to future research.

In most studies, LIFO constraints are very strict and significantly impact the loading plans. Better solutions, higher utilization rates, and more flexibility can be achieved if reloading is permitted during the development of routes. In addition, integrating situations such as drivers' social and human needs and physical conditions of roads into the problem with additional constraints while solving routing problems will ensure the reality and applicability of the solution. Also, the execution of a route can be affected by stochastic processes. Due to factors such as breakdown and traffic density, required driving times cannot always be reliably estimated. In this context, fuzzy planning aims to minimize the possibility of violating time windows.

In commercial constraint programming solvers, providing users with increased command over "interval" and "sequence" variables will enhance the effectiveness of solving scheduling, routing, and loading problems.

By making the integration of evolutionary algorithms and elitism more sophisticated, the parameters in the proposed sinewave approach can be redesigned to be functions of time. Thus, the balance of intensification and diversification can be achieved by searching the entire solution space. In addition, learning methods and parallel computing techniques can be used to reduce the computation time. To this end, some of the approaches developed in this thesis can be extended.

REFERENCES

Ahmeti, A., & Musliu, N. (2021). Hybridizing Constraint Programming and Meta-Heuristics for Multi-Mode Resource-Constrained Multiple Projects Scheduling Problem. In *Proceedings of the 13th International Conference on the Practice and Theory of Automated Timetabling-PATAT* (Vol. 1).

Akpinar, S. (2016). Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem. *Expert Systems with Applications*, 61, 28-38.

Apt, K. (2003). Principles of constraint programming. Cambridge university press.

Ayough, A., Khorshidvand, B., Massomnedjad, N., & Motameni, A. (2020). An integrated approach for three-dimensional capacitated vehicle routing problem considering time windows. *Journal of Modelling in Management*, 15(3),995-1015.

Baldi, M. M., Perboli, G., & Tadei, R. (2012). The three-dimensional knapsack problem with balancing constraints. *Applied Mathematics and Computation*, 218(19), 9802-9818.

Bartók, T., & Imreh, C. (2011). Pickup and delivery vehicle routing with multidimensional loading constraints. *Acta Cybernetica (Szeged), 20*(1), 17-33.

Battarra, M., Cordeau, J. F., & Iori, M. (2014). Chapter 6: pickup-and-delivery problems for goods transportation. In Vehicle Routing: *Problems, Methods, and Applications, Second Edition* (pp. 161-191). Society for Industrial and Applied Mathematics.

Benders, J. F. (2005). Partitioning procedures for solving mixed-variables programming problems. *Computational Management Science,* 2(1), 3-19.

Bent, R., & Van Hentenryck, P. (2006). A two-stage hybrid algorithm for pickup and delivery vehicle routing problems with time windows. *Computers & Operations Research, 33(4)*, 875-893.

Berbeglia, G., Cordeau, J. F., Gribkovskaia, I., & Laporte, G. (2007). Rejoinder on: Static pickup and delivery problems: a classification scheme and survey. *Top,* 15(1), 45-47.

Beyer, H. G., & Schwefel, H. P. (2002). Evolution strategies–a comprehensive introduction. *Natural computing*, 1(1), 3-52.

Bischoff, E. E. (2006). Three-dimensional packing of items with limited load bearing strength. *European Journal of Operational Research*, 168(3), 952-966.

Bischoff, E. E., & Ratcliff, M. S. W. (1995). Issues in the development of approaches to container loading. *Omega*, 23(4), 377-390.

Bortfeldt, A. (2012). A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, 39(9), 2248-2257.

Bortfeldt, A., & Homberger, J. (2013). Packing first, routing second—a heuristic for the vehicle routing and loading problem. *Computers & Operations Research*, 40(3), 873-885.

Bortfeldt, A., & Wäscher, G. (2013). Constraints in container loading–A state-of-the-art review. *European Journal of Operational Research*, *229*(1), 1-20.

Bortfeldt, A., & Yi, J. (2020). The split delivery vehicle routing problem with three-dimensional loading constraints. *European Journal of Operational Research*, *282*(2), 545-558.

Bortfeldt, A., Hahn, T., Männel, D., & Mönch, L. (2015). Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3D loading constraints. *European Journal of Operational Research*, *243*(1), 82-96.

Braekers, K., Ramaekers, K., & Van Nieuwenhuyse, I. (2016). The vehicle routing problem: State of the art classification and review. *Computers & Industrial Engineering*, 99, 300-313.

Cacchiani, V., Hemmelmayr, V. C., & Tricoire, F. (2014). A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics*, 163, 53-64.

Calvet, L., de Armas, J., Masip, D., & Juan, A. A. (2017). Learnheuristics: hybridizing metaheuristics with machine learning for optimization with dynamic inputs. *Open Mathematics*, *15*(1), 261-280.

Cappart, Q., Thomas, C., Schaus, P., & Rousseau, L. M. (2018, August). A constraint programming approach for solving patient transportation problems. In *International conference on principles and practice of constraint programming* (pp. 490-506). Springer, Cham.

Çavdar, B., & Sokol, J. (2019). A computation-implementation parallelization approach to the vehicle loading and routing problem. *Networks*, *73*(1), 119-134.

Ceschia, S., Schaerf, A., & Stützle, T. (2013). Local search techniques for a routing-packing problem. *Computers & Industrial Engineering*, *66*(4), 1138-1149.

Chen, C. S., Lee, S. M., & Shen, Q. S. (1995). An analytical model for the container loading problem. *European Journal of operational research*, 80(1), 68-76.

Chi, J., & He, S. (2023). Pickup capacitated vehicle routing problem with three-dimensional loading constraints: *Model and algorithms. Transportation Research Part E: Logistics and Transportation Review*, 176, 103208

Cordeau, J. F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3), 573-586.

Cordeau, J. F., & Groupe d'études et de recherche en analyse des décisions (Montréal, Québec). (2000). *The VRP with time windows*. Montréal: Groupe d'études et de recherche en analyse des décisions.

Côté, J. F., Guastaroba, G., & Speranza, M. G. (2017). The value of integrating loading and routing. *European Journal of Operational Research*, *257*(1), 89-105.

Curtois, T., Landa-Silva, D., Qu, Y., & Laesanklang, W. (2018). Large neighbourhood search with adaptive guided ejection search for the pickup and delivery problem with time windows. *EURO Journal on Transportation and Logistics*, 7(2), 151-192.

Dantzig, G. B., & Wolfe, P. (1960). Decomposition principle for linear programs. *Operations Research*, *8*(1), 101-111.

Darwin, C. (1859). *On the origin of species by means of natural selection, or the preservation of favoured races in the struggle for life*. London: John Murray.

De Backer, B., Furnon, V., Shaw, P., Kilby, P., & Prosser, P. (2000). Solving vehicle routing problems using constraint programming and metaheuristics. *Journal of Heuristics*, *6*(4), 501-523.

De Jong, K. A. (1975). *An analysis of the behavior of a class of genetic adaptive systems*. University of Michigan.

Desrochers, M., Lenstra, J. K., Savelsbergh, M. W., & Soumis, F. (1987). *Vehicle routing with time windows: optimization and approximation* (No. OS-R8715). CWI. Department of Operations Research and System Theory [BS].

Desrosiers, J., & Dumas, Y. (1988). The shortest path problem for the construction of vehicle routes with pick-up, delivery and time constraints. *In Advances in Optimization and Control: Proceedings of the Conference "Optimization Days 86" Held at Montreal, Canada, April 30–May 2,* 1986 (pp. 144-157). Springer Berlin Heidelberg.

Dumas, Y., Desrosiers, J., & Soumis, F. (1991). The pickup and delivery problem with time windows. *European journal of operational research*, 54(1), 7-22.

Dupas, R., Grebennik, I., Lytvynenko, O., & Baranov, O. (2017). An heuristic approach to solving the one-to-one pickup and delivery problem with three-dimensional loading constraints. *International Journal of Information Technology and Computer Science*, 9(10), 1-12.

Dyckhoff, H. (1990). A typology of cutting and packing problems. *European Journal of Operational Research*, 44(2), 145-159.

Edis, E. B., & Oguz, C. (2012). Parallel machine scheduling with flexible resources. *Computers & Industrial Engineering*, *63*(2), 433-447.

Eiben, A. E., & Smith, J. E. (2015). *Introduction to evolutionary computing*. Springer-Verlag Berlin Heidelberg.

Eksioglu, B., Vural, A. V., & Reisman, A. (2009). The vehicle routing problem: A taxonomic review. *Computers & Industrial Engineering*, 57(4), 1472-1483.

El Hachemi, N., Gendreau, M., & Rousseau, L. M. (2011). A hybrid constraint programming approach to the log-truck scheduling problem. *Annals of Operations Research*, *184*(1), 163-178.

Emmerich, M., Shir, O. M., & Wang, H. (2018). Evolution strategies. In *Handbook of Heuristics* (pp. 89-119). Springer, Cham.

Escobar-Falcón, L. M., Álvarez-Martínez, D., Granada-Echeverri, M., Escobar, J. W., & Romero-Lázaro, R. A. (2016). A matheuristic algorithm for the three-dimensional loading capacitated vehicle routing problem (3L-CVRP). *Revista Facultad de Ingeniería Universidad de Antioquia,* (78), 09-20.

Fekete, S. P., Schepers, J., & Van der Veen, J. C. (2007). An exact algorithm for higher-dimensional orthogonal packing. *Operations Research*, 55(3), 569-587.

Focacci, F., Lodi, A., & Milano, M. (2002). Mathematical programming techniques in constraint programming: A short overview. *Journal of Heuristics*, *8*(1), 7-17.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1965). Intelligent decision-making through a simulation of evolution. *Simulation*, 5(4), 267-279.

Fourier, J. B. J., & Darboux, G. (1822). *Théorie analytique de la chaleur* (Vol. 504). Paris: Didot.

Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2009). Ant colony optimization for the two-dimensional loading vehicle routing problem. *Computers & Operations Research*, 36(3), 655-673.

Fuellerer, G., Doerner, K. F., Hartl, R. F., & Iori, M. (2010). Metaheuristics for vehicle routing problems with three-dimensional loading constraints. *European Journal of Operational Research*, *201*(3), 751-759.

Gedik, R., Kirac, E., Milburn, A. B., & Rainwater, C. (2017). A constraint programming approach for the team orienteering problem with time windows. *Computers & Industrial Engineering*, *107*, 178-195.

Gedik, R., Rainwater, C., Nachtmann, H., & Pohl, E. A. (2016). Analysis of a parallel machine scheduling problem with sequence dependent setup times and job availability intervals. *European Journal of Operational Research*, *251*(2), 640-650.

Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. *simulation*, *76*(2), 60-68.

Gehring, H. (1997). A genetic algorithm for solving the container loading problem. *International transactions in operational research*, 4(5-6), 401-418.

Gehring, H., Menschner, K., & Meyer, M. (1990). A computer-based heuristic for packing pooled shipment containers. *European Journal of Operational Research*, 44(2), 277-288.

Gendreau, M., Iori, M., Laporte, G., & Martello, S. (2006). A tabu search algorithm for a routing and container loading problem. *Transportation Science*, *40*(3), 342-350.

George, J. A., & Robinson, D. F. (1980). A heuristic for packing boxes into a container. *Computers & Operations Research*, 7(3), 147-156.

Golden, B. L., Raghavan, S., & Wasil, E. A. (Eds.). (2008). *The vehicle routing problem: latest advances and new challenges* (Vol. 43). New York: Springer.

Gribkovskaia, I., & Laporte, G. (2008). One-to-many-to-one single vehicle pickup and delivery problems. In *The Vehicle Routing Problem: Latest Advances and New Challenges* (pp. 359-377). Springer, Boston, MA.

Gribkovskaia, I., Laporte, G., & Shyshou, A. (2008). The single vehicle routing problem with deliveries and selective pickups. *Computers & Operations Research,* 35(9), 2908-2924.

Guimarans, D., Herrero, R., Ramos, J. J., & Padrón, S. (2011). Solving vehicle routing problems using constraint programming and Lagrangean relaxation in a metaheuristics framework. *International Journal of Information Systems and Supply Chain Management (IJISSCM)*, *4*(2), 61-81.

He, F., & Qu, R. (2012). A constraint programming-based column generation approach to nurse rostering problems. *Computers & Operations Research*, *39*(12), 3331-3343.

Ho, S. C., & Szeto, W. Y. (2016). GRASP with path relinking for the selective pickup and delivery problem. *Expert Systems with Applications*, 51, 14-25.

Hokama, P., Miyazawa, F. K., & Xavier, E. C. (2016). A branch-and-cut approach for the vehicle routing problem with loading constraints. *Expert Systems with Applications*, *47*, 1-13.

Holland, J. (1975). Adaptation in natural and artificial systems, univ. of mich. press. *Ann Arbor*, *7*, 390-401.

Homberger, J., & Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 37(3), 297-318.

Hooker, J.N. (2007). *"Integrated Methods for Optimization,"* International Series in Operations Research and Management Science, Springer, number 978-0-387-38274-6, December.

Huang, Y. H., Hwang, F. J., & Lu, H. C. (2016). An effective placement method for the single container loading problem. *Computers & Industrial Engineering*, *97*, 212-221.

Husbands, P., Copley, P., Eldridge, A., & Mandelis, J. (2007). An introduction to evolutionary computing for musicians. In *Evolutionary computer music* (pp. 1-27). Springer, London.

IBM. 2019. "IBM ILOG CPLEX Optimization Studio V12. 9.0 documentation." *https://www.ibm.com/docs/en/icos/12.9.0*

Iori, M. (2016). An annotated bibliography of combined routing and loading problems. *Yugoslav Journal of Operations Research*, *23*(3), 311-326

Iori, M., & Martello, S. (2010). Routing problems with loading constraints. *Top*, *18*(1), 4-27.

Iori, M., Salazar-González, J. J., & Vigo, D. (2007). An exact approach for the vehicle routing problem with two-dimensional loading constraints. *Transportation science*, 41(2), 253-264.

Irnich, S., Toth, P., & Vigo, D. (2014). Chapter 1: The family of vehicle routing problems. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition* (pp. 1-33). Society for Industrial and Applied Mathematics.

Jain, V., & Grossmann, I. E. (2001). Algorithms for hybrid MILP/CP models for a class of optimization problems. *INFORMS Journal on Computing*, *13*(4), 258-276.

Jamil, M., & Yang, X. S. (2013). A literature survey of benchmark functions for global optimization problems. *arXiv preprint arXiv*:1308.4008.

Jamrus, T., & Chien, C. F. (2016). Extended priority-based hybrid genetic algorithm for the less-than-container loading problem. *Computers & Industrial Engineering*, 96, 227-236.

Józefowska, J., Pawlak, G., Pesch, E., Morze, M., & Kowalski, D. (2018). Fast truck-packing of 3D boxes. *Engineering Management in Production and Services*, *10*(2).

Jun, S., & Lee, S. (2022). Evolutionary neural network for learning of scalable heuristics for pickup and delivery problems with time windows. *Computers & Industrial Engineering*, 169, 108282.

Junqueira, L., & Morabito, R. (2015). Heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem in a carrier. *Computers & Industrial Engineering*, *88*, 110-130.

Junqueira, L., Oliveira, J. F., Carravilla, M. A., & Morabito, R. (2013). An optimization model for the vehicle routing problem with practical three-dimensional loading constraints. *International Transactions in Operational Research*, *20*(5), 645-666.

Kallehauge, B., Larsen, J., & Madsen, O. B. (2006). Lagrangian duality applied to the vehicle routing problem with time windows. Computers & Operations Research, 33(5), 1464-1487.

Karabulut, K., & İnceoğlu, M. M. (2004, October). A hybrid genetic algorithm for packing in 3D with deepest bottom left with fill method. In *International Conference on Advances in Information Systems* (pp. 441-450). Springer, Berlin, Heidelberg.

Kek, A. G., Cheu, R. L., & Meng, Q. (2008). Distance-constrained capacitated vehicle routing problems with flexible assignment of start and end depots. *Mathematical and Computer Modelling*, 47(1-2), 140-152.

Koch, H. (2018). *Vehicle routing problems with three-dimensional loading constraints and backhauls.* vorgelegt und angenommen an der Fakultat fur Wirtschaftswissenschaft Otto-von-Guericke-Universit¨at Magdeburg

Kok, A. L., Meyer, C. M., Kopfer, H., & Schutten, J. M. J. (2010). A dynamic programming heuristic for the vehicle routing problem with time windows and European Community social legislation. *Transportation Science*, 44(4), 442-454.

Kolen, A. W., Rinnooy Kan, A. H. G., & Trienekens, H. W. (1987). Vehicle routing with time windows. *Operations Research*, 35(2), 266-273.

Koza, J. R.. *Genetic Programming*. MIT Press, Cambridge, MA, 1992.

Krebs, C., & Ehmke, J. F. (2021). Axle weights in combined vehicle routing and container loading problems. *EURO Journal on Transportation and Logistics*, 10, 100043.

Krebs, C., Ehmke, J. F., & Koch, H. (2023). Effective loading in combined vehicle routing and container loading problems. *Computers & Operations Research*, 149, 105988.

Kucuk, M., & Ermis, M. (2010, October). A new hybrid evolutionary algorithm for three-dimensional packing problems. In *2010 IEEE International Conference on Systems, Man and Cybernetics* (pp. 4029-4034). IEEE.

Küçük, M., & Yıldız, Ş. T. (2019). A constraint programming approach for the pickup and delivery problem with time windows. *Pamukkale Üniversitesi Mühendislik Bilimleri Dergisi*, *25*(9), 1041-1049.

Küçük, M., & Yildiz, S. T. (2022). Constraint programming-based solution approaches for three-dimensional loading capacitated vehicle routing problems. *Computers & Industrial Engineering*, 171, 108505.

Laborie, P. (2009, May). IBM ILOG CP Optimizer for detailed scheduling illustrated on three problems. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems* (pp. 148-162). Springer, Berlin, Heidelberg.

Lacomme, P., Toussaint, H., & Duhamel, C. (2013). A GRASP× ELS for the vehicle routing problem with basic three-dimensional loading constraints. *Engineering Applications of Artificial Intelligence*, *26*(8), 1795-1810.

Laporte, G. (2009). Fifty years of vehicle routing. *Transportation science*, 43(4), 408-416.

Laporte, G., Desrochers, M., & Nobert, Y. (1984). Two exact algorithms for the distance-constrained vehicle routing problem. *Networks*, 14(1), 161-172.

Li, H., & Lim, A. (2001, November). A metaheuristic for the pickup and delivery problem with time windows. *In Proceedings 13th IEEE International Conference on Tools with Artificial Intelligence. ICTAI 2001* (pp. 160-167). IEEE.

Li, M. P., & Kuhl, M. E. (2017, December). Design and simulation analysis of PDER: A multiple-load automated guided vehicle dispatching algorithm. *In 2017 winter simulation conference (wsc)* (pp. 3311-3322). IEEE.

Mahvash, B., Awasthi, A., & Chauhan, S. (2017). A column generation based heuristic for the capacitated vehicle routing problem with three-dimensional loading constraints. *International Journal of Production Research*, *55*(6), 1730-1747.

Mak-Hau, V., Moser, I., & Aleti, A. (2018). An exact algorithm for the heterogeneous fleet vehicle routing problem with time windows and three-dimensional loading constraints. *In Data and Decision Sciences in Action* (pp. 91-101). Springer, Cham.

Männel, D., & Bortfeldt, A. (2016). A hybrid algorithm for the vehicle routing problem with pickup and delivery and three-dimensional loading constraints. *European Journal of Operational Research*, *254*(3), 840-858.

Männel, D., & Bortfeldt, A. (2018). Solving the pickup and delivery problem with three-dimensional loading constraints and reloading ban. *European Journal of Operational Research*, *264*(1), 119-137.

Martello, S., Pisinger, D., & Vigo, D. (2000). The three-dimensional bin packing problem. *Operations research*, 48(2), 256-267.

Martin, M., Hokama, P. H., Morabito, R., & Munari, P. (2020). The constrained two-dimensional guillotine cutting problem with defects: an ILP formulation, a Benders decomposition and a CP-based algorithm. *International Journal of Production Research*, *58*(9), 2712-2729.

Massomnedjad, N., & Motameni. A. 2020. An integrated approach for three-dimensional capacitated vehicle routing problem considering time windows. *Journal of Modelling in Management*, 15(3), 995-1015.

Miao, L., Ruan, Q., Woghiren, K., & Ruo, Q. (2012). A hybrid genetic algorithm for the vehicle routing problem with three-dimensional loading constraints. *RAIRO-Operations Research-Recherche Opérationnelle*, *46*(1), 63-82.

Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. *Knowledge-based systems*, *96*, 120-133.

Moura, A. (2019). A model-based heuristic to the vehicle routing and loading problem. *International Transactions in Operational Research*, *26*(3), 888-907.

Moura, A., & Oliveira, J. F. (2009). An integrated approach to the vehicle routing and container loading problems. *OR Spectrum*, *31*(4), 775-800.

Mourgaya, M., & Vanderbeck, F. (2006). The periodic vehicle routing problem: classification and heuristic. *RAIRO-Operations Research*, 40(2), 169-194.

Nasiri, A., Keedwell, E., Dorne, R., Kern, M., & Owusu, G. (2022, July). A hyper-heuristic approach for the PDPTW. *In Proceedings of the Genetic and Evolutionary Computation Conference Companion* (pp. 196-199).

Ngoenriang, N., Sawadsitang, S., Leangsuksun, C., Niyato, D., & Tan, P. S. (2019, April). Joint Vehicle Routing and Loading in Delivery Planning: A Stochastic Programming Approach. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)* (pp. 1-5). IEEE.

North, A. C., & Helliwell, J. R. (2013). X-Ray Crystallography of Macromolecules, *Theory, and Methods*.

Pace, S., Turky, A., Moser, I., & Aleti, A. (2015). Distributing fibre boards: a practical application of the heterogeneous fleet vehicle routing problem with time windows and three-dimensional loading constraints. *Procedia Computer Science*, *51*, 2257-2266.

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2006). A survey on pickup and delivery models Part II: Transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, *58*, 81-117

Parragh, S. N., Doerner, K. F., & Hartl, R. F. (2008). A survey on pickup and delivery problems. Journal für Betriebswirtschaft, 58(1), 21-51.

Pesant, G., Gendreau, M., Potvin, J. Y., & Rousseau, J. M. (1998). An exact constraint logic programming algorithm for the traveling salesman problem with time windows. *Transportation Science*, *32*(1), 12-29.

Pinto, T. (2016). *Models and advanced optimization algorithms for the integrated management of logistics operations* (Doctoral dissertation, Universidade do Minho (Portugal)).

Pinto, T., Alves, C., & Valério de Carvalho, J. (2018). Models and advanced optimization algorithms for the integrated management of logistics operations. In *Operational Research:* IO2017, Valença, Portugal, June 28-30 XVIII (pp. 313-324). Springer International Publishing.

Pollaris, H., Braekers, K., Caris, A., Janssens, G. K., & Limbourg, S. (2015). Vehicle routing problems with loading constraints: state-of-the-art and future directions. *OR Spectrum*, *37*(2), 297-330.

Pullen, H. G. M., & Webb, M. H. J. (1967). A computer application to a transport scheduling problem. *The computer journal*, 10(1), 10-13.

Qu, R., & He, F. (2008, December). A hybrid constraint programming approach for nurse rostering problems. In *International Conference on Innovative Techniques and Applications of Artificial Intelligence* (pp. 211-224). Springer, London.

Quoc, N. H. B., & Anh, D. T. (2010, November). Combining constraint programming and Tabu search to solve vehicle routing problem with time windows. In *2010 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)* (pp. 1-4). IEEE.

Rajaei, M., Moslehi, G., & Reisi-Nafchi, M. (2022). The split heterogeneous vehicle routing problem with three-dimensional loading constraints on a large scale. *European Journal of Operational Research*, 299(2), 706-721.

Rechenberg, H. (1971). *Aspects of the Teaching of Quantum Mechanics* (Vol. 105). University of Texas at Austin, Center for Particle Theory.

Reil, S., Bortfeldt, A., & Mönch, L. (2018). Heuristics for vehicle routing problems with backhauls, time windows, and 3D loading constraints. *European Journal of Operational Research*, *266*(3), 877-894.

Ren, J., Tian, Y., & Sawaragi, T. (2011). A tree search method for the container loading problem with shipment priority. *European Journal of Operational Research*, 214(3), 526-535.

Renaud, J., Laporte, G., & Boctor, F. F. (1996). A tabu search heuristic for the multi-depot vehicle routing problem. *Computers & Operations Research*, 23(3), 229-235.

Režnar, T., Martinovič, J., Slaninová, K., Grakova, E., & Vondrák, V. (2017). Probabilistic time-dependent vehicle routing problem. *Central European Journal of Operations Research*, 25(3), 545-560.

Rodrígueza, D. A. A., Martíneza, D. Á., & Escobarb, J. W. (2022). A hybrid matheuristic approach for the vehicle routing problem with three-dimensional loading constraints. *International Journal of Industrial Engineering Computations*, 13, 421-434.

Ropke, S., & Cordeau, J. F. (2009). Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, 43(3), 267-286.

Ropke, S., & Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4), 455-472.

Rousseau, L. M., Gendreau, M., & Pesant, G. (2002). Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of Heuristics*, *8*(1), 43-58.

Rousseau, L. M., Gendreau, M., Pesant, G., & Focacci, F. (2004). Solving VRPTWs with constraint programming-based column generation. *Annals of Operations Research*, *130*(1-4), 199-216.

Ruan, Q., Zhang, Z., Miao, L., & Shen, H. (2013). A hybrid approach for the vehicle routing problem with three-dimensional loading constraints. *Computers & Operations Research*, *40*(6), 1579-1589.

Ruland, K. S., & Rodin, E. Y. (1997). The pickup and delivery problem: Faces and branch-and-cut algorithm. *Computers & mathematics with applications*, 33(12), 1-13.

Schwefel, H. P. (1977). Evolutionsstrategien für die numerische optimierung. In Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie (pp. 123-176). Birkhäuser, Basel.

Shaw, P. (1998, October). Using constraint programming and local search methods to solve vehicle routing problems. In *International Conference on Principles and Practice of Constraint Programming* (pp. 417-431). Springer, Berlin, Heidelberg.

Simon, D. (2013). *Evolutionary optimization algorithms.* John Wiley & Sons.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research,* 35(2), 254-265.

Solomon, M. M., & Desrosiers, J. (1988). Survey paper—time window constrained routing and scheduling problems. *Transportation science*, 22(1), 1-13.

Song, X., Jones, D., Asgari, N., & Pigden, T. (2020). Multi-objective vehicle routing and loading with time window constraints: a real-life application. *Annals of Operations Research,* 291(1), 799-825.

Sörensen, K., Sevaux, M., & Glover, F. (2018). A history of metaheuristics. In *Handbook of heuristics* (pp. 791-808). Springer, Cham.

Surjanovic, S. & Bingham, D. (2013). Virtual Library of Simulation Experiments: Test Functions and Datasets. Retrieved October 11, 2022, from *http://www.sfu.ca/~ssurjano*.

Talatahari, S., & Azizi, M. (2021). Chaos Game Optimization a novel metaheuristic algorithm. *Artificial Intelligence Review*, 54(2), 917-1004.

Talbi, E. G. (2009). *Metaheuristics: from design to implementation* (Vol. 74). John Wiley & Sons.

Talbi, E. G. (2016). Combining metaheuristics with mathematical programming, constraint programming and machine learning. *Annals of Operations Research*, *240*(1), 171-215.

Talbi. E. G. (Ed.). (2013). *Hybrid metaheuristics* (Vol. 166). Berlin Heidelberg: Springer.

Tan, K. C., Chew, Y. H., & Lee, L. H. (2006). A hybrid multiobjective evolutionary algorithm for solving vehicle routing problem with time windows. *Computational Optimization and Applications*, 34(1), 115-151.

Tan, S. Y., & Yeh, W. C. (2021). The vehicle routing problem: state-of-the-art classification and review. *Applied Sciences*, 11(21), 10295.

Tao, Y., & Wang, F. (2015). An effective tabu search approach with improved loading algorithms for the 3L-CVRP. *Computers & Operations Research*, *55*, 127-140.

Tarantilis, C. D., Zachariadis, E. E., & Kiranoudis, C. T. (2009). A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container-loading problem. *IEEE Transactions on Intelligent Transportation Systems*, *10*(2), 255-271.

Terno, J., Scheithauer, G., Sommerweiß, U., & Riehme, J. (2000). An efficient approach for the multi-pallet loading problem. *European Journal of Operational Research*, 123(2), 372-381.

Toth, P., & Vigo, D. (Eds.). (2002). "*The vehicle routing problem."* Society for Industrial and Applied Mathematics.

Toth, P., & Vigo, D. (Eds.). (2014). *Vehicle routing: problems, methods, and applications*. Society for Industrial and Applied Mathematics.

Varun Kumar, S. G., & Panneerselvam, R. (2017). A study of crossover operators for genetic algorithms to solve VRP and its variants and new sinusoidal motion crossover operator. *International Journal of Computational Intelligence Research*, 13(7), 1717-1733.

Vega-Mejía, C. A., Montoya-Torres, J. R., & Islam, S. M. (2019). A nonlinear optimization model for the balanced vehicle routing problem with loading constraints. *International Transactions in Operational Research*, *26*(3), 794-835.

Vega-Mejía, C., González-Neira, E., Montoya-Torres, J., & Islam, S. (2020). Using a hybrid heuristic to solve the balanced vehicle routing problem with loading constraints. *International Journal of Industrial Engineering Computations*, *11*(2), 255-280.

Vidal, T., Crainic, T. G., Gendreau, M., & Prins, C. (2013). Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. *European Journal of Operational Research*, 231(1), 1-21.

Vidal, T., Laporte, G., & Matl, P. (2020). A concise guide to existing and emerging vehicle routing problem variants. *European Journal of Operational Research*, 286(2), 401-416.

Wallace, M. (2006, June). Hybrid algorithms in constraint programming. In *International Workshop on Constraint Solving and Constraint Logic Programming* (pp. 1-32). Springer, Berlin, Heidelberg.

Wang, F., Tao, Y., & Shi, N. (2009, April). A survey on vehicle routing problem with loading constraints. In *2009 International Joint Conference on Computational Sciences and Optimization* (Vol. 2, pp. 602-606). IEEE.

Wäscher, G., Haußner, H., & Schumann, H. (2007). An improved typology of cutting and packing problems. *European journal of operational research*, 183(3), 1109-1130.

Wei, L., Zhang, Z., & Lim, A. (2014). An adaptive variable neighborhood search for a heterogeneous fleet vehicle routing problem with three-dimensional loading constraints. *IEEE Computational Intelligence Magazine*, 9(4), 18-30.

Wei, L., Zhang, Z., Zhang, D., & Leung, S. C. (2018). A simulated annealing algorithm for the capacitated vehicle routing problem with two-dimensional loading constraints. *European Journal of Operational Research*, *265*(3), 843-859.

Weisstein, Eric W. "Fourier Series--Square Wave." From MathWorld--A Wolfram Web Resource. https://mathworld.wolfram.com/FourierSeriesSquareWave.html

Wolpert, D. H., & Macready, W. G. (1995). No free lunch theorems for search (Vol. 10). *Technical Report* SFI-TR-95-02-010, Santa Fe Institute.

Wu, J., Zheng, L., Huang, C., Cai, S., Feng, S., & Zhang, D. (2019, November). An improved hybrid heuristic algorithm for pickup and delivery problem with three-dimensional loading constraints. *In 2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 1607-1612). IEEE.

Yang, X. S. (2009). *Harmony search as a metaheuristic algorithm. In Music-inspired harmony search algorithm* (pp. 1-14). Springer, Berlin, Heidelberg.

Yu, B., Yang, Z. Z., & Yao, B. (2009). An improved ant colony optimization for vehicle routing problem. *European journal of operational research,* 196(1), 171-176.

Zachariadis, E. E., Tarantilis, C. D., & Kiranoudis, C. T. (2013). Designing vehicle routes for a mix of different request types, under time windows and loading constraints. *European Journal of Operational Research*, *229*(2), 303-317.

Zhang, J., Lam, W. H., & Chen, B. Y. (2013). A stochastic vehicle routing problem with travel time uncertainty: trade-off between cost and customer service. *Networks and Spatial Economics*, 13(4), 471-496.

Zhang, Z., Wei, L., & Lim, A. (2015). An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. *Transportation Research Part B: Methodological*, *82*, 20-35.

Zhao, X., Bennell, J. A., Bektaş, T., & Dowsland, K. (2016). A comparative review of 3D container loading algorithms. *International Transactions in Operational Research*, 23(1-2), 287-320.

Zheng, Y. J. (2015). Water wave optimization: a new nature-inspired metaheuristic. *Computers & Operations Research*, *55*, 1-11.

Zhu, W., Qin, H., Lim, A., & Wang, L. (2012). A two-stage tabu search algorithm with enhanced packing heuristics for the 3L-CVRP and M3L-CVRP. *Computers & Operations Research*, *39*(9), 2178-2195.

Zhu, Z., Xiao, J., He, S., Ji, Z., & Sun, Y. (2016). A multi-objective memetic algorithm based on locality-sensitive hashing for one-to-many-to-one dynamic pickup-and-delivery problem. *Information Sciences*, 329, 73-89.