LAST MILE DELIVERY ROUTING PROBLEM USING AUTONOMOUS ELECTRIC VEHICLES

by
NIMA MORADI

Submitted to the Graduate School of Engineering and Natural Sciences
in partial fulfilment of
the requirements for the degree of Master of Science

Sabancı University
July 2022

ABSTRACT

LAST MILE DELIVERY ROUTING PROBLEM USING AUTONOMOUS ELECTRIC VEHICLES

NIMA MORADI

INDUSTRIAL ENGINEERING M.Sc. THESIS, JULY 2022

Thesis Supervisor: Prof. Bülent Çatay
Thesis Co-advisor: Dr. Mir Ehsan Hesam Sadati

Keywords: Vehicle routing problem, autonomous electric vehicle, covering-routing problem, two-phase heuristic, variable neighbourhood search, simulated annealing

Distribution management is one of the important elements of the supply chain or logistic system due to its large contribution to the total cost of the system. One of the growing industries in distribution management is using autonomous electric vehicles for last mile delivery. Applying autonomous delivery vehicles (ADV) to deliver the products has many applications in grocery shopping, logistics, food delivery, etc. In the real-world situations for ADV, a subset of delivery sites (customers) is visited directly; however, the remaining customers must be covered by (assigned to) the delivery sites en-route with which their distance is within the maximum walking distance. Accordingly, the present thesis studies a last-mile delivery routing using ADV which is a covering-routing problem (or median-routing problem) satisfying the load capacity, route distance/duration, and customer's walking distance constraints. The addressed problem is called Covering Electric Vehicle Routing problem (CE-VRP). Two mathematical models are proposed for CE-VRP: one with the assignment cost as the objective function, and the other with the assignment distance as a constraint. The proposed models are developed according to efficient mathematical models proposed for handling the constraints of the maximum route distance/duration in the literature with the polynomial number of constraints and decision variables. Due to the NP-hardness of the CE-VRP, a new two-phase heuristic consisting of selecting the delivery sites and customers assignment (first phase), and routing the vehicles visiting the delivery sites (second phase) is proposed to solve the large-sized instances. Also, several efficient *repair* and *improvement* operators are proposed in the first phase, and a hybrid Variable Neighbourhood search with Simulated Annealing (VNS-SA) metaheuristic is designed to find the high-quality routes by diversifying and intensifying the solution space in the second phase. The computational results show the efficiency of the proposed method in solving the various-sized instances of CE-VRP and other covering-routing problems. Finally, concluding remarks and suggestions for future studies are stated.

ÖZET

OTONOM ELEKTRİKLİ ARAÇLAR İLE SON KİLOMETRE DAĞITIM ROTALAMASI PROBLEMİ

NIMA MORADI

ENDÜSTRİ MÜHENDİSLİĞİ YÜKSEK LİSANS TEZİ, TEMMUZ 2022

Tez Danışmanı: Prof. Bülent Çatay
Tez Yardımcı Danışmanı: Dr. Mir Ehsan Hesam Sadati

Anahtar Kelimeler: Araç rotalama problemi, otonom elektrikli araç, kapsama-rotalama problemi, iki aşamalı sezgisel, değişken komşuluk arama, benzetimli Tavlama

Dağıtım yönetimi, sistemin toplam maliyetine yaptığı büyük katkı nedeniyle tedarik zinciri veya lojistik sistemin önemli unsurlarından biridir. Elektrikli otonom araçların son kilometre teslimatı için kullanımı dağıtım yönetiminde hızla gelişen endüstrilerden biridir. Ürünlerin teslimatında otonom dağıtım araçlarının (ODA) kullanımının, market alışverişi, lojistik, yiyecek dağıtımı vb. alanlarda birçok uygulaması bulunmaktadır. Gerçek hayatta ODA kullanılırken, teslimat yerlerinin bir alt kümesi (müşteriler) doğrudan ziyaret edilir; ancak, kalan müşteriler, mesafelerinin maksimum yürüme mesafesi içerisinde olduğu yol üzerindeki teslimat yerleri (atanmış) tarafından karşılanmalıdır. Bu tez çalışmasında, ODA kullanılarak son kilometre teslimatının rotalandırılması, yük kapasitesi, rota mesafesi/süresi ve müşterinin yürüme mesafesi kısıtlarını sağlayan kapsama rotalama (medyan rotalama) problemi incelenmektedir. Bu problemi Elektrikli Araç Kapsama-Rotalama Problemi (EA-KRP) olarak adlandırdık. EA-KRP için önerilen matematiksel modellerin ilkinde: atama maliyeti amaç fonksiyonu olarak diğerinde ise atama mesafesi kısıt olarak formüle edilmiştir. EA-KRP'nin NP zorluğu nedeniyle, büyük ölçekli problemleri çözmek için teslimat yerlerinin seçilmesini ve müşterilerin atanmasını (birinci aşama) ve teslimat yerlerini ziyaret eden araçların rotalandırılmasını (ikinci aşama) içeren iki aşamalı bir sezgisel yöntem önerilmiştir. Birinci aşamada onarma ve iyileştirme operatörlerinden faydalanılırken. ikinci aşamada çözüm uzayını çeşitlendirerek ve

yoğunlaştırarak yüksek kaliteli rotaları bulabilmek için hibrit bir Değişken Komşuluk Araması Benzetilmiş Tavlama (DKA-BT) metasezgisel yaklaşım tasarlanmıştır. Yapılan deneysel çalışmaların sonuçları, önerilen yöntemin çeşitli büyüklükteki EA-KRP örnekleri ile Kapasiteli Araç Rotalama ve Kapsama Rotalama problemlerini çözmedeki etkinliğini göstermektedir. Son olarak, elde edilen sonuçlar değerlendirilerek ve gelecek çalışmalar için öneriler sunulmuştur.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my primary supervisor, Prof. Bülent Çatay, and my co-advisor, Dr. Mir Ehsan Hesam Sadati, who guided me throughout this project. I would also like to thank my friends, Nozir, Neslihan, and Arghavan, who offered deep insight into the study and helped me in editing the in Turkish abstract.

I also especially appreciate my family who supported me throughout this thesis.

To my dear parent
For their constant support and kindness

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ABC Artificial Bee Colony Algorithm

ACA Ant Colony Algorithm

ADV Autonomous Delivery (Electric) Vehicle

BCTP Bi-objective Maximal Covering Minimal Tour Problem

BVNS Basic Variable Neighbourhood search

CE-VRP Covering Electric-Vehicle Routing Problem

CG Column Generation

CluTSP clustered Travelling Salesman Problem

CluVRP clustered Vehicle Routing Problem

Cm-RSP Capacitated Multi-vehicle Ring Star problem

CoVRP Covering Vehicle Routing Problem

CoVRPwIT Covering Vehicle Routing Problem with Integrated Tours

CPTP Capacitated Profitable Tour Problem

CRTP Capacitated Ring Tree Problem

CSP Covering Salesman Problem

CTP Covering Tour Problem

CTSPP Covering Traveling Salesman Problem with Profit

CVRP Capacitated Vehicle Routing problem

DCVRP Distanced-constrained Capacitated Vehicle Routing Problem

DGVRP Distance-constrained Generalized Vehicle Routing Problem

EA Evolutionary Algorithm

E-VRP Electric Vehicle Routing Problem

EV Electric Vehicle

GA Genetic Algorithm

GCSP Generalized Covering Salesman Problem

GRASP Greedy Randomized Adaptive Search Procedure

GTSP Generalized Travelling Salesman Problem

G-VRP Green Vehicle Routing Problem

GVRP Generalized Vehicle Routing Problem

GVRPTW Generalized Vehicle Routing Problem with Time Windows

IP Integer Linear Programming

LMCP Location Maximal Covering Problem

LNS Large Neighbourhood Search

LRP Location-Routing Problem

LSCP Location Set Covering Problem

MCPTP Multi-vehicle Capacitated Profitable Tour Problem

MCTP Maximal Covering Tour Problem

mm-CTP Multi-vehicle Multi-Covering Tour Problem

m-RSP Multi-Ring Star problem

m-CTP Multi-vehicle Covering Tour Problem

MCCP Maximum Covering Cycle Problem

MCP Median Cycle Problem

MDCTVRP Multi-Depot Covering Tour Vehicle Routing Problem

MIP Mixed-integer Linear Programming

MTP Median Tour Problem

PCCTP Prize-collecting Covering Tour Problem

PCTSP Prize-collecting Travelling Salesman Problem

PCVRP Prize-collecting Vehicle Routing Problem

RSP Ring Star Problem

RTFLP Ring Tree Facility Location Problem

SA Simulated Annealing

SRSP Steiner Ring Star Problem

SS Scatter Search

STP Steiner Tree Problem

TCCVRP Time-constrained Capacitated Covering Vehicle Routing Problem

TCMCRP Time-constrained Maximal Covering Routing Problem

TCMCSP Time-constrained Maximal Covering Salesman Problem

TCVRP Time-constrained Capacitated Vehicle Routing Problem

TCVRPTW Time-constrained Vehicle Routing Problem with Time Windows

TOP Team Orientation Problem

TS Tabu Search

TSP Travelling Salesman Problem

TSPP Travelling Salesman Problem with Profits

VND Variable Neighborhood Descent

VNS Variable Neighbourhood Search

VNS-SA Variable Neighbourhood Search with Simulated Annealing

VRAP Vehicle Routing-Allocation Problem

VRDAP Vehicle Routing with Demand Allocation Problem

VRP Vehicle Routing Problem

VRPPC Vehicle Routing Problem with Private Fleet and Common Carriers

1. INTRODUCTION

One of the important elements of a logistic or supply chain system is distribution management. Distribution management is defined as the activities and processes which lead to distributing the products from their origin to different destinations or customers and delivering the products to the final consumer[1]. Distributing any product or goods plays a critical role in the economical aspects of a service provider company due to its direct impact on the company's costs and customer satisfaction (Farham-Nia & Ghaffari-Hadigheh, 2021). An important activity inside the distribution system is the last-mile delivery, which is defined as the final step of the product journey from the last delivery hub to the hand of the customer[2]. Also, last-mile delivery activities can account for 53% of the total shipping costs and 41% of total supply chain costs and even this number can be increased due to inefficient policies in the supply chain or routing sectors[3].

Nowadays, one approach to last-mile delivery is to use *autonomous delivery vehicle* (ADV) to deliver the products in a cheap and fast way both of which are critical criteria for any company providing delivery services. ADV's are electric vehicles (EV) in their nature, which are self-driving vehicles without any human role[4]. Due to the large number of customers who prefer to order and buy a product via the online platform, and fast delivery, ADV's are becoming a major part of any urban logistic and product delivery in the next decade (Gao, Kaas, Mohr & Wee, 2016). It is expected that the ADV's market "will rise at a Compound annual growth rate (CAGR) of 11% from 2019 to reach the 200 billion USD in 2029" (see the website https://aiworldschool.com/ for more information on ADV). Udelv[5] is a start-up company that applies the ADV's for last-mile or middle-mile delivery with

[1] https://www.shipbob.com/blog/distribution-management/

[2] https://en.wikipedia.org/wiki/Last_mile_(transportation)

[3] https://onfleet.com/blog/what-is-last-mile-delivery/

[4] https://aiworldschool.com/

[5] https://www.udelv.com/

the technology of the autonomous vehicles provided by Mobileye (a system software company that is active in developing the autonomous driving technologies). Fig. 1.1 shows a sample of the ADV's suggested by Udelv company to its customers. This type of ADV is a multi-stop EV with a "hot-swappable modular cargo pod called the uPod". Also, this multi-stop ADV has a load capacity at maximum 2,000 pounds of goods, is able to stop 80 times on each trip, travels between 160 and 300 miles per tour (trip) considering the battery level and is connected to "Udelv's mobile apps" to monitor its routing, delivery and returning activities[6].

Figure 1.1 The first cab-less autonomous electric delivery vehicle designed for multi-stop delivery provided by Udelv company (Source of the image: https://mma.prnewswire.com/)

The multi-stop ADV will be one of the popular EV's due to its efficiency and capability to deliver the products in an economical and fast way as a part of the last-mile delivery processes[7]. One of the biggest challenges in the last-mile delivery of ADV's is the "routing" decisions which need a fast decision-maker algorithm to route its path from origin to the final destination while delivering the products to the customers on time and returning to the origin without violating the load capacity, maximum route distance/duration constraints. At first, after receiving the orders of the customers, each ADV is filled with the ordered products considering the load capacity and coverage area of the ADV (see Fig. 1.2). Next, the route planning for an ADV with respect to the locations of the customers is found by a route optimization algorithm connected to the central server system. Then, the customers walk a distance to reach the stopping location of the ADV (delivery site or "hot spots") to pick up their products or goods from the parcel lockers by providing

[6] https://www.prnewswire.com/

[7] https://www.udelv.com/about

their electronic personal information (see Fig. 1.3). Finally, after delivering the all products, the ADV moves to another stopping location or returns to the origin. A successful use of the autonomous electric vehicles is the Mobile Locker for grocery delivery (Draeger's Market, a San Mateo, CA grocery store[8]). In this grocery delivery example, each autonomous vehicle is filled with the products and then after traveling for 30 minutes, it arrives at the stopping location and then the customers have 1 hour to pick up their orders.

Figure 1.2 The cargo section of the multi-stop autonomous electric delivery vehicle provided by Udelv company (Source of the image: https://mma.prnewswire.com/)

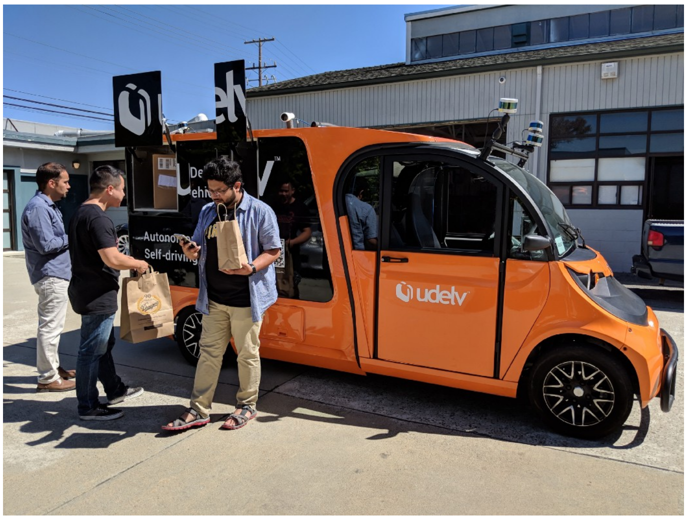


Figure 1.3 Customers delivery process from the parcel lockers within the autonomous electric delivery van provided by Udelv company (Source of the image: https://medium.com/)

In last-mile delivery by the ADV's, route optimization plays a critical role in the productivity of usage of ADV's and justifying their usage as a delivery vehicle in the transportation or supply chain systems. The multi-stop ADV, as mentioned

[8] https://medium.com/@udelv/mobile-lockers-new-delivery-methods-by-self-driving-cars-de04e50d4cea

above, have limited traveling distance/duration and cargo capacity which limits its capability in visiting more customers. Despite these limitations in using ADV, efficient routing of the vehicles and assignment of the customers to a selected delivery site (stopping location of ADV) will encounter limitations while minimizing the routing and usage costs. Although the decision of routing the EV's, known as the Electric Vehicle Routing Problem (E-VRP) (Schneider, Stenger & Goeke, 2014), has been studied by many researchers in recent years, there is no work on routing the multi-stop ADV and assigning the customers to each stopping location of ADV while considering the constraints of route duration (time) limit and distance (length) limit, load capacity, and maximum walking distance for the customers to reach a delivery site. Also, it is worthy to say that, in this thesis, the literature of E-VRP or Green VRP (G-VRP) are not reviewed since in the E-VRP and its variants, the location of the charging stations are found, which is not in the scope of this thesis. For interested readers, some review works of E-VRP (Erdelić & Carić, 2019; Kucukoglu, Dewil & Cattrysse, 2021; Lin, Zhou & Wolfson, 2016), G-VRP (Erdoğan & Miller-Hooks, 2012; Lin, Choy, Ho, Chung & Lam, 2014; Moghdani, Salimifard, Demir & Benyettou, 2021; Sharafi & Bashiri, 2016), Autonomous Vehicle Logistic System (AVLS) (James & Lam, 2017), and Autonomous Vehicle Routing Problem Solution (VRPS) considering pilot satisfaction (Yan, 2018) are suggested.

To find the optimal or near-optimal solution of routing the ADV considering the several constraints, both routing and assigning (covering) decisions must be taken into account. In the real-world situations for ADV, a subset of customers are visited in their location; however, the remaining customers must be covered by or assigned to a delivery site with which their distance is within the maximum walking distance. So, the addressed problem in this thesis is different from the classic Vehicle Routing Problem (VRP) (originally introduced in Dantzig & Ramser (1959)) in which all customers must be visited directly exactly one time by one vehicle, and all vehicles must start their trip from and return to the origin (depot). Accordingly, the last-mile delivery routing using the autonomous electric vehicle, or ADV, is a covering-routing problem (or median-routing problem) while considering the load capacity, route distance/duration, and customer's walking distance constraints. Considering an upper bound on the walking distance of a customer is an important parameter that indicates customer accessibility. This problem, the last-mile delivery routing using the autonomous electric vehicles, is called Covering Electric-Vehicle Routing Problem (CE-VRP) in the present thesis and it is the main addressed problem, which has not been studied previously, to our best knowledge, in the literature.

CE-VRP not only has practical importance in the last-mile delivery of electric-vehicles routing and urban logistics, but it has also theoretical attractiveness because of its complexity and extending the sub-problems that lie in the class of the VRP and covering-routing problems. This thesis aims to address the CE-VRP (a covering-routing problem finding the best routes for the vehicle to visit a subset of customers directly and the optimal assignment of the unvisited customers to the customers on the main routes (called delivery sites) satisfying the load capacity, route length/time, and walking distance constraints) and propose a mathematical programming model for it and design an efficient optimization solver based on heuristic/metaheuristic methods. The main contributions of the present thesis can be listed as follows:

- A new covering-routing problem is introduced as Covering Electric-Vehicle Routing Problem (CE-VRP) with respect to a real last-mile delivery routing problem;
- A new classification scheme is introduced for the covering-routing problems in which isolating the customers is not allowed and all customers must be either visited or covered;
- The studied CE-VRP extends the literature on the covering-routing problems by considering the load capacity, route distance & duration, and maximum walking distance constraints;
- A new mixed-integer programming model is developed for the addressed problem;
- An efficient two-phase heuristic, consisting of a new greedy construction algorithm, a hybrid metaheuristic to find the near-optimal routes, and several problem-specific repair and improvement operators to optimize the covering (assignment) decisions, is proposed to solve the large-sized instances of CE-VRP;
- The performance of the proposed two-phase heuristic and repair & improvement operators is validated by solving the various variants of CE-VRP and comparing the results with the existing solvers in the literature;

The present thesis is structured as follows: in the second chapter, the literature on the related problems from routing problem to covering-routing are reviewed while introducing a new classification scheme for the most-related problem to the addressed problem in this thesis, CE-VRP. In the third chapter, the main problem is stated and a mathematical model, which is polynomial in the number of decision variables and constraints, is proposed in two versions. In the fourth chapter, the solution methodology is explained briefly and a two-phase heuristic with efficient repair and improvement operators is described. In the fifth chapter, the computational results to validate the performance of the proposed solution method and several improvement operators are presented beside the comparison study for evaluating the performance of the proposed heuristic in solving the main problem and its variants. Finally, in the sixth chapter, the concluding remarks and suggestions for future studies are stated.

2. LITERATURE REVIEW

In this chapter, the related works to the main problem of the present thesis are reviewed. The review is started with the general routing problems and then continued by introducing the various variants of it, and finally, the most-related problems to the addressed problem in terms of features and assumptions are summarized by introducing a new classification scheme for such problems.

2.1 Routing problems

Routing problems are among the well-known and well-studied Combinatorial Optimization Problems (COP) in the context of Operations Research (OR) which find a (closed) path starting from an initial point and finishing at a final point while optimizing an objective and satisfying some constraints. From the graph[1] perspective, routing problems are classified into two sub-problems: incomplete tour, say path, finding and complete tour, say cycle, construction problems. Path finding problems consist of Single-Source Shortest Path (SSSP) (Medak & Gogoi, 2018) and All-Pairs Shortest Path (APSP) (Katz & Kider, 2008). Cycle construction problems aim to find a cycle (a complete tour) that goes through the nodes of the graph by traversing the edges; it is called a cycle because after starting the tour from a specific node, called as origin or depot, it finishes the tour at the same node (point).

The simple form of the cycle construction problem is known as Travelling Salesman Problem (TSP) (Flood, 1956) which tries to find a minimum-length cycle inside the graph visiting all nodes exactly one time. TSPs can be classified in two ways: based on constraints, or Hamiltonian property (Laporte & Martín, 2007). If they are classified based on the constraints, two general classes including unconstrained,

[1] A visual representation in discrete mathematics which works with a set of nodes (vertices) connected by a set of arc (edges) (West, 2001)

and constrained sub-problems are introduced. Also, if they are classified based on Hamiltonian property, then Hamiltonian (section 2.2) and non-Hamiltonian (section 2.3) sub-problems appears. The present thesis focuses on the non-Hamiltonian cycle construction problem and its extensions.

2.2 Hamiltonian cycle construction problems

In Hamiltonian cycle construction problems, all nodes (customers) must be visited directly by the salesmen (vehicles); for example, in Capacitated VRP (CVRP), the aim is to find the routes for more than one vehicle while satisfying the load capacity of each vehicle and visiting each customer exactly one time. One of the Hamiltonian cycle construction problems, which is close to the addressed problem of the thesis, is Clustered VRP (CluVRP), first introduced by Sevaux & Sörensen (2008). The CluVRP is the generalized version of Clustered TSP (CluTSP), in which there are multiple vehicles (Snyder & Daskin, 2006). As a simple definition of CluTSP, it tries to find the main cycle by constructing the secondary cycles on some of the nodes assigned to the main cycle. In CluVRP, the clusters and the nodes within each cluster are determined in advance, and two decisions include finding the routes visiting the centers of each cluster and finding the routes serving all customers within each cluster. For Hamiltonian cycle construction problems and their variants, interested readers are referred to the review work of Toth & Vigo (2014).

2.3 Non-Hamiltonian cycle construction problems

In non-Hamiltonian cycle construction problems, the assumption of visiting all customers is relaxed, so a subset of customers may not be visited. There are three approaches to handling the non-visited customers: one way is to leave some of the customers isolated (called non-Hamiltonian problems with profits); the second way is to *assign* them to (*cover* them by) a customer or node which is on the vehicle's tour (called set covering non-Hamiltonian problems); and the third way which is a combination of two previous approaches, in which customers are visited or covered or isolated (called maximal covering non-Hamiltonian problems). These three approaches with their sub-problems are explained in the following.

2.3.1 Non-Hamiltonian problems with profits

In the first approach, non-Hamiltonian problems with profits, or isolating some customers, each customer associates with a prize, or reward or profit, to be visited or not. For example, in non-Hamiltonian TSP with profits, some customers may not be chosen to be on the tour, which leads to TSP with Profits (TSPP) (Feillet, Dejax & Gendreau, 2005). Consequently, TSPP tries to find a solution with minimum tour cost and maximum total profit collected from the customers while satisfying some constraints (usually capacity or tour duration constraints). Both tour cost and total profit can be considered either as an objective function or as a constraint. If the objective function is the maximization of the collected total profit with an upper bound on the total traveled distance, it is called selective TSP (Laporte & Martello, 1990) or Orienteering Problem (OP) (Golden, Levy & Vohra, 1987). If the objective function is to minimize the total traveling (tour) cost with a lower bound on the total profit collected, it is called Prize-collecting TSP (PCTSP) (Balas, 1989). Finally, if the objective function is to optimize a combination of both tour cost and profit collected without any restriction on them, it is known as (Capacitated) Profitable Tour Problem (CPTP) (Jepsen, Petersen, Spoorendonk & Pisinger, 2014).

Moreover, in selective TSP, PCTSP, and CPTP sub-problems, if more than one vehicle (salesman) is used, then three new variants of selective VRP, or Team Orientation Problem (TOP)[2] (Sabo, Pop & Horvat-Marc, 2020), Prize-collecting VRP (PCVRP) (Tang & Wang, 2006), and Multi-vehicle CPTP (MCPTP) (Handoko, Chuin, Gupta, Soon, Kim & Siew, 2015) are developed, respectively. Since this thesis does not focus on the prize-related TSP'ss or VRP's, the interested readers are referred to the seminal works such as (Archetti, Speranza & Vigo, 2014; Balas, 2007; Dell'Amico, Maffioli & Värbrand, 1995; Gendreau, Laporte & Semet, 1998; Laporte & Martello, 1990; Vidal, Maculan, Ochi & Vaz Penna, 2016). Furthermore, there are several other problems which are very close to TSPP or VRP with Profits (VRPP), namely Arc Routing Problem with Profits (ARPP) (Archetti, Feillet, Hertz & Speranza, 2010) in which arcs associate with a prize or profit, Tourist Trip Design Problem (TTDP) (Gavalas, Konstantopoulos, Mastakas & Pantziou, 2014), Mixed TOP (MTOP) (Gavalas, Konstantopoulos, Mastakas, Pantziou & Vathis,

[2]TOP has been also studied under the name Multiple Tour Maximum Collection Problem (MTMCP) in the literature (Butt & Cavalier, 1994).

2016) which considers prize or profit for both nodes and arcs, Tour Suggestion Problem for Leisure and Tourism (TSPLT) (Maervoet, Brackman, Verbeeck, Causmaecker & Berghe, 2013), Bus Touring Problem (BTP) (Deitch & Ladany, 2000), VRP with Private Fleet and Common Carriers (VRPPC) (Bolduc, Renaud, Boctor & Laporte, 2008), in which each customer must be visited by a private vehicle or outsourced to a common carrier; VRPPC is very close to PCVRP if outsourcing a customer is considered equal to not visiting a customer (minimizing the outsourcing cost is equal to the minimizing the penalty cost of not visiting).

2.3.2 Set covering non-Hamiltonian problems

The second approach, set covering non-Hamiltonian problems, or covering a subset of customers, leads to introducing a new class of tour construction problems known as Median Cycle Problem (MCP), "cycle" for constructing a cycle, or ring or tour, in the solution representation, and "median" for being a simple median problem at some nodes to cover the remaining nodes; MCP is also known as Ring Star Problem (RSP) in the literature (Labbé, Laporte, Martín & González, 2004) (see section 2.3.2.3). In other words, in MCP all customers must be either visited or covered by another customer, or node, which is on the tour. Throughout this thesis, *covering, assigning*, and *allocating* are referring to the same concept in the cycle construction problem. In the following, MCP and its variants are more elaborated due to their closeness to the present thesis' scope.

In this thesis, a new classification framework for set covering non-Hamiltonian problems is introduced which is adapted from the classification scheme for maximal covering non-Hamiltonian problems, under the name routing and path planning with spatial coverage, originally introduced by Glock & Meyer (2022).

2.3.2.1 Classification scheme

In this section, a new classification scheme for set covering non-Hamiltonian problems is presented. In this thesis, all set covering non-Hamiltonian problems are defined by ten characteristics, including "depot type", "customer type", "Steiner point type", "depot kind", "customer kind", "Steiner point kind", "constraint type", "objective function", "the number of tours (vehicles)", and "the number of depots" as given in Table 2.1. Also, in this classification, we do not consider the set covering non-Hamiltonian problems with partial or probabilistic coverage and continuous topological space since the main focus of this thesis is not reviewing and classifying all problems but related ones. First of all, three first characteristics, including "depot type", "customer type", and "Steiner point type" are explained in the following.

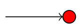
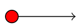
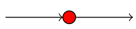
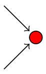
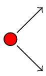
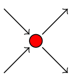
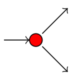
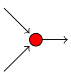
Table 2.1 Classification scheme for set covering non-Hamiltonian problems

Characteristic		**Notation**
Depot type		$\mathcal{D}$
Customer type		$\mathcal{C}$
Steiner point type		$\mathcal{S}$
Depot kind	Optional	$\mathcal{D}_o$
	Mandatory	$\mathcal{D}_m$
Customer kind	Optional	$\mathcal{C}_o$
	Mandatory	$\mathcal{C}_m$
Steiner point kind	Optional	$\mathcal{S}_o$
	Mandatory	$\mathcal{S}_m$
Constraint type	Load capacity (LC)	$\mathcal{LC}$
	Distance-constrained vehicles (DV)	$\mathcal{DV}$
	Time-constrained vehicles (TV)	$\mathcal{TV}$
	Time windows (TW)	$\mathcal{TW}$
	Backhaul customers (BC)	$\mathcal{BC}$
	Pickup and delivery (PD)	$\mathcal{PD}$
	Precedence relationship (PR)	$\mathcal{PR}$
	Maximum walking distance	$\mathcal{MC}$
Objective function	Min. route cost	$\mathcal{O}_1$
	Min. assignment cost	$\mathcal{O}_2$
	Min. delivery site opening cost	$\mathcal{O}_3$
	Min. depot opening cost	$\mathcal{O}_4$
The number of routes (vehicles)	Single	$\mathcal{K}_1$
	Multiple	$\mathcal{K}_2$
The number of depots (delivery site)	Single	$\mathcal{W}_1$
	Multiple	$\mathcal{W}_2$

Suppose there is only a node, or a point, with several directed arcs. Now let's assume that the direction of each arc represents the flow of the products or goods. For example, $\longrightarrow$ means that some products are transported from the left-hand side to the right-hand side. According to the different combinations of a node and single (multiple) in-degree or out-degree of the node, a total of nine various combinations could be generated as given in Table 2.2.

Now we define the "depot" as a node which can be represented by any types of "a" to "i" (given in Table 2.2). The "customer" is defined as a special kind of "depot" that cannot generate any product, or the out-degree of the "customer" node must be zero (note that a "customer" node can have an out-degree, which leads to the routing problems with pickup and backhaul customers; so based on the definition here, the "customer" node can be represented by any types of "b", "d", "e", "g", "h", or "i" (note that a customer with the type of "a" is considered in the non-Hamiltonian problems with profits or maximal covering non-Hamiltonian problems in which the customer can be left isolated). Also, another node type is defined which must have at least one out-degree (except in the case in which there is no in-degree, then it can have no out-degree), and also it cannot be an initial or origin node. This new node is known as the "Steiner point", or transit or transition point, in the literature (Baldacci, Dell'Amico & González, 2007), which is just for helping the products to flow between the nodes; in other words, the "Steiner point" cannot be a node having no out-degree (cannot be a terminal or sink node). Therefore, "Steiner point" can only be represented by the types of "a", "d", "g", "h", or "i". Finally, a "delivery site point" (Ceselli, Righini & Tresoldi, 2014) or "parcel locker" (Jiang, Zang, Dong & Liang, 2022) is equivalent to the "Steiner point", so throughout this thesis, "Steiner point", "delivery site point", and "parcel locker" refer to the same type of node. Accordingly, the provided above definitions for "depot", "customer", and "Steiner point" are true for the depots, customers, and delivery sites in the remaining part of the thesis.

Table 2.2 Different combinations of a node and single (multiple) in-degree or out-degree of the node

Combination of a node and arcs	**In-degree**	**Out-degree**	**Name**	**notation**
	0	0	Isolated node	a
	1	0	Terminal node	b
	0	1	Initial node	c
	1	1	On-route node	d
	> 1	0	Sink node	e
	0	> 1	Origin node	f
	> 1	> 1	Hub node type 1	g
	1	> 1	Hub node type 2	h
	> 1	1	Hub node type 3	i

As a result, each depot, customer and Steiner point can be represented by the node types given in Table 2.2. For instance, depot type is defined by a non-empty subset of the set $\{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}, \mathsf{e}, \mathsf{f}, \mathsf{g}, \mathsf{h}, \mathsf{i}\}$, customer type is defined by a non-empty subset of the set $\{\mathsf{b}, \mathsf{d}, \mathsf{e}, \mathsf{g}, \mathsf{h}, \mathsf{i}\}$, and Steiner (delivery site) type is defined by a non-empty subset of the set $\{\mathsf{a}, \mathsf{d}, \mathsf{g}, \mathsf{h}, \mathsf{i}\}$ in any set covering non-Hamiltonian problem.

In the proposed classification scheme, kind of each depot, customer or Steiner point can be in two modes of *optional* or *mandatory* (Glock & Meyer, 2022). For example, a depot node with kind of optional is free to be on the tours or not; however, a depot node with kind of mandatory, which must be on at least one tour, has no freedom to be an isolated node. Also, suppose a customer with optional kind to be visited or covered, then the type of this customer can be chosen from the set $\{\mathsf{b}, \mathsf{d}, \mathsf{e}, \mathsf{g}, \mathsf{h}, \mathsf{i}\}$; however, if a customer is with mandatory kind, which must be visited or be on a tour, then this customer cannot be covered, so its type is reduced to the set $\{\mathsf{d}, \mathsf{g}, \mathsf{h}, \mathsf{i}\}$. Therefore, considering the kinds of optional or mandatory for each node is important and it has to be taken into account in classifying the set covering non-Hamiltonian problems.

Moreover, the characteristic of constraint type in the classification scheme consists of seven constraints defined for tour construction problems plus the maximum coverage distance for the nodes with notation $\mathcal{MC}$. The characteristic of the objective function indicates the type of the objective function of a problem which can be one or more of the following objectives: minimizing the route cost, minimizing the assignment (covering) cost, minimizing the opening cost of a delivery (Steiner) site, and minimizing the opening cost of a depot (in multi-depot problems).

In the following, various set covering non-Hamiltonian problems from the simplest to more complex, introduced in the literature, are explained briefly and then these problems are classified according to the proposed classification scheme at the end to determine their similarities and differences with the addressed problem in the present thesis.

2.3.2.2 Location Set Covering Problem

The covering problems try to locate the facilities such that they are able to "cover" or serve a set of customers. In this context, a customer is *coverable* by a facility if and only if the customer is located within a predefined coverage distance of d_{max} from the facility. There are two types of the objective functions in covering models as follows: trying to cover all customers with the smallest number of facilities; or covering as many customers as possible with a given number of facilities. The former objective type is referred to as Location Set Covering Problem (LSCP) (first introduced by Toregas, Swain, ReVelle & Bergman (1971)), and the latter one is known as Location Maximal Covering Problem (LMCP).

2.3.2.3 Median Cycle Problem (MCP) or Ring Star Problem (RSP)

MCP was first introduced by Labbé, Laporte, Martín & González (1999) in 1999. MCP is also known as RSP in the literature; both MCP and RSP have the same mathematical model (Labbé et al., 2004); however, mostly, RSP refers to a problem with the assignment cost in the objective function, and MCP refers to a problem with an upper bound on the assignment cost (as a constraint) (Laporte & Martín, 2007). RSP aims to find a cycle containing a subset of nodes of a graph while minimizing two kinds of costs: the routing cost of the cycle, and the cost of the nodes' assignment (allocation) to the nodes which are on the cycle.

According to the literature, two main solution procedures have been designed for MCP and RSP: exact approaches, and heuristic-based solvers. Several exact solution techniques have been proposed for MCP/RSP as follows: branch-and-cut algorithm (Kedad-Sidhoum & Nguyen, 2010; Labbé et al., 1999), Separation procedures with branch-and-cut algorithm (Labbé, Laporte, Martın & González, 2005), epsilon-constraint method (Püskül, Aslan, Onay, Erdogan & Taşgetiren, 2022). Moreover, heuristic solvers for MCP include Variable Neighborhood Tabu Search (VNTS) (Martín, 2000; Pérez, Moreno-Vega & Martın, 2003), greedy Evolutionary Algorithm (EA) (Renaud, Boctor & Laporte, 2003,0), elitist EA and a population-based local search (Liefooghe, Jourdan & Talbi, 2010), differential EA (Tasgetiren, Bulut, Pan & Suganthan, 2011), EA (Calvete, Galé & Iranzo, 2013a), approximation and heuristic algorithms (Chen, Hu, Tang, Wang & Zhang, 2017), Ant Colony System Algorithm (Zang, Jiang, Ding & Fang, 2021), Multi-objective EA with Local Search (MEALS) (Calvete, Galé & Iranzo, 2016).

Moreover, if in MCP/RSP there exists an upper bound on the number of the nodes (or stops) on the cycle (tour), then MCP/RSP becomes Median Tour Problem (MTP) (Current & Schilling, 1994). There are several works on the MTP and its variants in the literature, such as heuristics on MTP (Current & Schilling, 1996), and recently introduced Generalized MTP (GMTP) (Obreque, Paredes-Belmar, Miranda, Campuzano & Gutiérrez-Jarpa, 2020). Also, if in MCP/RSP only some specific nodes (Steiner points) must be on the cycle (tour) and the customers must be assigned to these Steiner points, then MCP/RSP becomes Steiner RSP (SRSP), originally introduced by Lee, Chiu & Sanchez (1998). SRSP has been studied by proposing mathematical models and solution techniques, including MIP modeling (Yuh, Lee & Park, 2014), and First Level Scatter Search (SS) algorithm (Xu, Chiu & Glover, 2001),

2.3.2.4 Covering Salesman Problem (CSP)

In TSP, if the assumption to visit directly all customers at their node is relaxed, then a new variant of TSP, called Covering Salesman Problem (CSP) (Current & Schilling, 1989), appeared. In CSP, the customers must be visited directly by a vehicle or covered by another node (customer, delivery, or distribution points). In other words, CSP tries to find a route to visit some customers, not all of them and then assigns the non-visited customers to the nodes on the route (covers them) while minimizing the route and assignment (covering) costs. In CSP, in addition to visiting or covering a customer, the covered customer must be within a pre-specified covering distance from the covering customer to be assignable to (coverable by) it. Also, the only difference between CSP and RSP is that in CSP the assignment part is considered as a constraint with an upper bound on the traveled (covering) distance for each customer; however, RSP minimizes the assignment cost as the objective function without considering a maximum on covering distance.

Moreover, a recent extension of CSP is Generalized CSP (GCSP), in which each customer i needs to be assigned (covered) at least k_i times, and also visiting each node associates with a cost (Golden, Naji-Azimi, Raghavan, Salari & Toth, 2012; Naji Azimi, Salari, Golden, Raghavan & Toth, 2009). Also, if in CSP it is assumed that some customers must be visited and the remaining ones must be covered, then CSP becomes Covering Tour Problem (CTP) (Gendreau, Laporte & Semet, 1997). In addition, it can be assumed that some arcs (edges) in the graph of CSP are removed, known as a non-complete graph. Also, it can be defined that a node i is coverable by node j if and only if both nodes would be adjacent to each other (there exists an arc from i to j or vice versa). If in CSP the coverability is defined by adjacent nodes in a non-complete graph, and it is allowed the cycle not to visit some of the nodes, then CSP becomes a very new problem, Maximum Covering Cycle Problem (MCCP), first introduced by Grosso, Salassa & Vancroonenburg (2016) (for some mathematical proofs on MCCP see (Briskorn, 2019)). MCCP has been also solved by branch-and-cut (Álvarez-Miranda & Sinnl, 2020).

CSP and its variants have been investigated by several models and solution techniques, such as Approximation algorithms (Arkin & Hassin, 1994), integer programming (IP)-based local search (Salari & Naji-Azimi, 2012), branch-and-bound (Maziero, Usberti & Cavellucci, 2021), SS (Baldacci, Boschetti, Maniezzo & Zamboni, 2005), Variable Neighbourhood Search (VNS) (Shaelaiea, Naji-Azimib & Salaria, 2014), Artificial Bee Colony (ABC) with Genetic algorithm (GA) (Pandiri, Singh & Rossi, 2020), A multi-start Iterated Local Search (ILS) (Venkatesh, Srivastava & Singh, 2019), GA (Tripathy, Tulshyan, Kar & Pal, 2017), parallel VNS (Zang, Jiang, Ratli & Ding, 2020), hybrid EA (Lu, Benlic & Wu, 2021), Deep Reinforcement Learning (Li, Zhang, Wang, Wang, Han & Wang, 2021), heuristic algorithms (Alinaghian & Goli, 2015), heuristic for CSP with Nodes and Segments (CSPNS) (Matsuura & Kimura, 2017) in which both nodes and segments on the tour can cover the nodes not on the tour, and Non-dominated Sorting GA (NSGA)-II for multi-objective CSP (Tripathy, Biswas & Pal, 2021).

Furthermore, several methods have been proposed for CTP, including reduction rules (Motta, Nogueira & Ochi, 2010; Motta, Ochi & Martinhon, 2001b), combined cooperative strategy (EA with branch-and-cut algorithm) (Jozefowiez, Semet & Talbi, 2007), ABC (Pandiri & Singh, 2019), Greedy Randomized Adaptive Search Procedure (GRASP) (Motta, Ochi & Martinhon, 2001a), A selector operator-based adaptive Large Neighborhood Search (LNS) (Leticia Vargas, Jozefowiez & Ngueveu, 2015; Vargas, Jozefowiez & Ngueveu, 2015), dynamic programming (Vargas, Jozefowiez & Ngueveu, 2017), heuristic algorithm using local search (Murakami, 2018a). Also, according to the literature, the other variants and extensions of CTP are multi-objective CTP (Nolz, Doerner, Gutjahr & Hartl, 2010), Multi-vehicle CTP (m-CTP) (Ha, Bostel, Langevin & Rousseau, 2013; Hachicha, Hodgson, Laporte & Semet, 2000; Jozefowiez, 2014; Kammoun, Derbel, Ratli & Jarboui, 2015,1; Lopes, Souza & da Cunha, 2013; Margolis, Song & Mason, 2022; Murakami, 2014,1; Oliveira, Moretti & Reis, 2015; Ziegler, 2013) in which m routes are found to visit a part of the customers and cover the remaining ones while satisfying a maximum tour length, multi-objective m-CTP (Glize, Roberti, Jozefowiez & Ngueveu, 2020), m-CTP with split delivery (Naji-Azimi, Renaud, Ruiz & Salari, 2012), Multi-vehicle multi-CTP (mm-CTP) (Kammoun, Derbel & Jarboui, 2018,2; Pham, Hà & Nguyen, 2017) (for dataset description on mm-CTP see (Pham, Hà & Nguyen, 2018)), Multi-Vehicle Probabilistic CTP (MVPCTP) (Karaoğlan, Erdoğan & Koç, 2018), Prize-collecting CTP (PCCTP) (Clímaco, Rosseti, Silva & Guerine, 2021), stochastic CTP (Tricoire, Graf & Gutjahr, 2012; Zehetner & Gutjahr, 2018), geometric CTP (Arkin & Hassin, 1991),online CTP (Zhang & Xu, 2018), distance-constrained generalized m-CTP (Singh, Kamthane & Tanksale, 2021), Multi-vehicle Cumulative CTP (m-CCTP) (Flores-Garza, Salazar-Aguilar, Ngueveu & Laporte, 2017), Multi-Depot Covering Tour Vehicle Routing Problem (MDCTVRP), first introduced by Allahyari, Salari & Vigo (2015), in which CTP is generalized by considering more than one vehicle and more than one depot.

2.3.2.5 Capacitated Multiple-Ring Star Problem (Cm-RSP)

Capacitated Multi-Ring Star Problem (Cm-RSP) was first introduced by Baldacci et al. (2007). Cm-RSP aims to find the set of cycles (rings), so that, each cycle includes a central depot, a subset of customers, and other possible points called transition (transit or *Steiner*) points that can be used to reduce routing costs. Each customer must be either on a cycle or assigned to a customer, which is on a cycle, or a transit point. Also, each transit point can be visited at most once and it cannot be assigned to a customer or other transit point. The number of customers on a cycle plus the number of customers assigned to at least one of the customers or transit points on that cycle must not exceed the capacity of the cycle (route). Therefore, Cm-RSP extends the MCP/RSP by considering the capacitated m ring stars.

Due to the theoretical and practical importance, Cm-RSP have been studied in terms of both modeling and solution procedures. For example, in addition to two-index formulation (Baldacci et al., 2007), Cm-RSP has been also modeled by two-commodity flow formulation (Baldacci et al., 2007), and set covering formulation (Hoshino & De Souza, 2012) in the literature. Moreover, there are two main solution approaches for Cm-RSP, including exact and metaheuristics. The works proposing exact solvers for Cm-RSP are as follows: branch-and-cut (Baldacci et al., 2007), Column Generation (Hoshino & Souza, 2008), branch-and-cut-and-price (Fouilhoux & Questel, 2014a; Hoshino & de Souza, 2009; Hoshino & De Souza, 2012), branch-and-cut (Berinsky & Zabala, 2011), Dynamic programming (Baldacci, Hill, Hoshino & Lim, 2017). Also, heuristic/metaheuristic works on Cm-RSP consist of Heuristic algorithms (Naji Azimi, Salari & Toth, 2009; Naji-Azimi, Salari & Toth, 2010), IP-based Local Search (Toth, Naji Azimi & Salari, 2011), VNS with an IP-based improvement (Naji-Azimi, Salari & Toth, 2012; Salari, Naji Azimi & Toth, 2011), Memetic Algorithm (MA) (Zhang, Qin & Lim, 2014), combined GRASP and Tabu Search (TS) (Mauttone, Nesmachnow, Olivera & Amoza, 2008), VNS (Franco, López-Santana & Mendez-Giraldo, 2016; Salari, Naji-Azimi & Toth, 2010). Furthermore, several variants for Cm-RSP have been introduced in the literature, such as Cm-RSP under Diameter Constrained Reliability (Cm-RSP-DCR) (Bayá, Mauttone, Robledo, Romero & Rubino, 2016), multi-objective Cm-RSP (Barma, Dutta, Mukherjee & Kar, 2021; Calvete, Galé & Iranzo, 2013b), Capacitated m Two-Node Survivable Star Problem (Cm-TNSSP) (Baya, Mauttone & Robledo, 2017; Bayá, Mauttone, Robledo & Romero, 2016a,1; Bayá Mantani, Mauttone Vidales & Robledo Amoza, 2015), Non-Disjoint Multi-Ring-Star Problem (NDRSP) (Fouilhoux & Questel, 2012,1).

2.3.2.6 Generalized Vehicle Routing Problem (GVRP)

In the Generalized VRP (GVRP) the customers are clustered into several groups, but visiting a customer in each group suffices to visiting all remaining customers in the cluster. The GVRP finds the optimal routes for the vehicles considering the constraints of load capacity, visiting exactly one of the customers within each cluster to minimize the route costs. Therefore, GVRP is composed of three decisions: assignment of the vehicles to each cluster, routing the vehicles, and selection of a specific node in each cluster as a stopping point (delivery site) for the vehicle visiting that cluster. Also, GVRP is the extended version of Generalized TSP (GTSP) in which there is only one route (vehicle).

GVRP was first introduced by Ghiani & Improta (2000) as an extension of VRP. After that, GVRP has been studied by several researchers due to its both theoretical and practical consequences (Baldacci, Bartolini & Laporte, 2010; Shimizu & Sakaguchi, 2013). In GVRP, the set of the nodes is partitioned into $c+1$ *mutually exclusive* nonempty subsets, called clusters, $R_0, R_1, ..., R_c$. The cluster R_0 has only one node 0, which represents the depot, and the remaining n nodes are distributed among the other clusters. Each customer has a certain amount of demand and the total demand of each cluster can be satisfied by any of its nodes. Also, there exist m vehicles with a capacity of Q. Also, here, for GVRP, the demand of each customer is considered equal to the total demand of that customer's cluster. The decision variables of GVRP are defined as follows: x_{ijm}: Binary variable equals to 1 if vehicle m travels from $i \in V$ to $j \in V$; 0 otherwise. z_i: Binary variable equals to 1 if the customer i is selected in the tour; 0 otherwise.

The GVRP and its variants have been solved by several exact and approximate solvers: constructive and local search algorithms for GVRP (Pop, Zelina, Lupşe, Sitar & Chira, 2011), TS for GVRP with Time Windows (GVRPTW) (Moccia, Cordeau & Laporte, 2012), branch-and-cut for GVRP (Bektaş, Erdoğan & Røpke, 2011), new IP for GVRP (Pop, Kara & Marc, 2012), GA with a local–global approach for GVRP (Pop, Matei & Sitar, 2013), Mixed-integer Linear Programming (MIP) and a heuristic for GVRP with Backhaul and Linehaul Customers (GVRPB) Mitra (2005), Ant Colony Algorithm (ACA) for GVRP (Pop, Pintea, Zelina & Dumitrescu, 2009), Column Generation (CG) and two metaheuristics for GVRP with Flexible Fleet Size (GVRP-flex) (Afsar, Prins & Santos, 2014), branch-and-cut and GRASP for GVRP-flex (Hà, Bostel, Langevin & Rousseau, 2014), branch-and-cut for GVRP with Stochastic Demands (GVRPSD) (Biesinger, Hu & Raidl, 2016), VNS for GVRP (Pop, Fuksz & Marc, 2014), CG for GVRPTW (Yuan, Cattaruzza, Ogier, Semet & Vigo, 2021), GA for GVRPSD (Biesinger, Hu & Raidl, 2018), GA for GVRP (Pop, Matei & Valean, 2011), VNS for GVRPSD (Biesinger, Hu & Raidl, 2015), ACA for Dynamic GVRP (DGVRP) (Pop, Pintea & Dumitrescu, 2009), branch-cut-and-price algorithm for GVRP (Reihaneh & Ghoniem, 2017a), LNS for GVRPTW (Dumez, Tilk, Irnich, Lehuédé & Péton, 2021), branch-and-cut for GTSP with Time Windows (GTSPTW) (Yuan, Cattaruzza, Ogier & Semet, 2020), LNS for GVRP (Mattila, 2018), parallel universes' algorithms with TS for GVRP (Navidadham, Arbabsadeghi, Bayat & Didehvar, 2015), comparison of heuristics on GVRP (Pop, Matei, Burzu & Gyorodi, 2011), heuristic algorithm for Distance-Constrained GVRP (DGVRP) (Mattila, 2016), MIP for GVRP with synchronization and precedence side constraints (Quttineh, Larsson, Lundberg & Holmberg, 2013), Dantzig–Wolfe decomposition and CG for time-indexed GVRP (Van den Bergh, Quttineh, Larsson & Beliën, 2016).

2.3.2.7 Location-Routing Problem (LRP)

The Location-Routing Problem (LRP) (Laporte, 1987) is composed of a set of customers with known demand and a set of potential depot sites. LRP finds the location of the depots and the vehicle routes from the depots to the customers by minimizing the locating depots and routing costs. Also, there is a fixed depot opening cost at each potential site and the route costs. Each customer is assigned to exactly one depot where the available vehicles are located. A vehicle route must start from and end at the same depot. Some limitations may be considered in LRP, such as the maximum travel time of a route, the maximum distance traveled by a vehicle, the maximum capacity of a vehicle, or the maximum capacity of a depot.

2.3.2.8 Vehicle Routing with Demand Allocation Problem (VRDAP)

Vehicle Routing with Demand Allocation Problem (VRDAP) (Ghoniem, Scherrer & Solak, 2013) can be considered in the context of LRP, and VRP with Intermediate Facilities (VRP-IF) (Angelelli & Speranza, 2002; Schneider, Stenger & Hof, 2015). VRDAP can be represented by a graph $G = (N, E)$, where $N = \{0\} \cup K \cup S$, and K is the set of customers, S is the set of potential delivery sites, and $\{0\}$ is the central depot. Also, V is the set of vehicles with the capacity Q. Each vehicle starts from the central depot and sequentially visits a subset of delivery sites to supply goods to customers, and returns the central depot. Each delivery site is visited at most one time by a vehicle. Each customer $i \in K$ has a demand d_i. Also, each edge (i,j) has the route cost c_{ij} and f_{ij} as the cost of assigning customer $j \in K$ to the delivery site $i \in S$. Decision variables of VRDAP are as follows: $s_{ij} = 1$ *iff* customer $i \in K$ is assigned to the delivery site $j \in S$, $e_{ijk} = 1$ *iff* the edge (i,j) is traversed by the vehicle k, $\theta_{ijk} = 1$ *iff* the delivery site $i \in S$, to which the customer $j \in K$ is assigned, is visited by the vehicle k, and q_{iv} is the total cumulative deliveries made upon serving site $i \in S$ by the vehicle v.

2.3.2.9 Multi-depot Covering Tour Vehicle Routing Problem (MDCTVRP)

As mentioned in the previous section, MDCTVRP was first introduced by Allahyari et al. (2015) (To our best knowledge this is the only work on MDCTVRP), in which CSP is generalized by considering more than one vehicle and more than one depot. In the proposed MDCTVRP by Allahyari et al. (2015), not only is there more than one depot, but there is also more than one heterogeneous vehicle. The demand of each customer is satisfied by either visiting or covering it by the customers on the tours (if it is placed within a coverage distance from the customers on the routes). In other words, MDCTVRP is a combination of the Multi-depot VRP (MDVRP) and CSP (Allahyari et al., 2015). MDCTVRP aims to find the minimum-cost tours for a fleet of heterogeneous vehicles departing from a selected depot of a set of potential depots and return to it while covering the non-visited customers with a minimum covering (assignment) cost.

2.3.2.10 Covering Vehicle Routing Problem (CoVRP)

Covering Vehicle Routing Problem (CoVRP)[3], recently introduced by Buluc, Peker, Kara & Dora (2021), aims to find the routes to visit a subset of customers and cover the remaining customers if they are located within a predetermined coverage distance from the customers on the routes. In other words, CoVRP generalizes the

[3] In this thesis, we choose the abbreviation "CoVRP" for Covering Vehicle Routing Problem to avoid any conflict with the well-known CVRP (capacitated version)

CSP with considering multiple vehicles. Also, CoVRP is very similar to m-CTP, but in m-CTP unlike the CoVRP, the set of the nodes that can be visited are different from the nodes that can be covered e.g., Steiner points are not considered in CoVRP. Moreover, CoVRP has a distance constraint on the tour duration for each vehicle, which is not considered in m-CTP. Also, CoVRP has not assumed any capacity load for each vehicle unlike the m-CTP and most of VRP variants. In the same work, Buluc et al. (2021) generalized the CoVRP by finding the tours for the covered nodes instead of enforcing the covered customers to visit the covering (stopping) customer. This new variant was called CoVRP with integrated tours (CoVRPwIT) which finds two kinds of routes: the main routes which visit the part of the customers, and the secondary routes which are designed for the covered customers by the customers on the main routes. CoVRPwIT can be helpful and applicable when it is not possible or profitable for customers to come to the covering nodes, instead, the company finds profitable secondary routes to visit the covered customers.

2.3.2.11 Capacitated Ring Tree Problem (CRTP)

Capacitated Ring Tree Problem (CRTP) has appeared for the first time in the work of Hill & Voß (2016), which provided a general representation of the variants of routing-allocation and ring star problems. CRTP is the combination between the cycle (ring)-based models like TSP with tree-based models like Steiner Tree Problem (STP) (Imase & Waxman, 1991) considering capacity constraint. In network design models, *ring tree* is defined as a graph consisting of a cycle $\mathscr{C}$ and a set of "node disjoint tress" $\mathscr{T}$, each of them intersecting with $\mathscr{C}$ in exactly one node. If the number of cycles (rings) equals 1 i.e., $|\mathscr{C}| = 1$, then the ring tree is called a pure tree, and also if $\mathscr{T} = 0$, then the ring tree becomes a cycle. So, if in m-RSP some customers are visited directly or covered, then Cm-RSP becomes CRTP. Therefore, in CRTP each customer (U) must be either on a cycle (ring) or connected to a cycle using the assignment. Also, Steiner points are just for helping the network design to reduce the routing and assignment costs, so they can be on a cycle or not. There must be at most m ring trees, each of them contains at most q customers (both visited and allocated), ring tree capacity. CRTP reduces to TSP if $U_1 = \emptyset$, $W = \emptyset$, $m = 1$, and $q = |U_2|$; as a result, CRTP is NP-hard (Hill & Voß, 2016).

The facility location version of CRTP is known as the Ring Tree Facility Location Problem (RTFLP) and it was first introduced by Abe, Hoshino & Hill (2015). RT-FLP combines VRP and STP. Two layers of networks are designed in RTFLP to connect two different types of customers to a central depot. "The first (inner) layer consists of cycles that intersect in the depot and collect all type 2 customers, and some of the type 1 customers. The second (outer) layer consists of a forest that contains the remaining type 1 customers such that each tree shares exactly one vertex with the inner layer" (Abe et al., 2015). For each ring tree and tree of the forest, there is an upper bound on the number of customers in each tree and tree capacity.

2.3.3 Maximal covering non-Hamiltonian problems

In the third approach, maximal covering non-Hamiltonian problems, customers have three destinies: they are visited, or covered, or not visited (not covered). Nice recent work on classifying the maximal covering non-Hamiltonian problems is the work by Glock & Meyer (2022), in which these kinds of problems are called "routing and path planning problems with spatial coverage". In the following, some maximal covering non-Hamiltonian problems, which are close to this thesis' problem, are described.

2.3.3.1 Vehicle Routing-Allocation problem (VRAP)

In the Vehicle Routing-Allocation Problem (VRAP) (Beasley & Nascimento, 1996) there are three alternatives to serve the customers (nodes): visiting directly by a vehicle, assigning a customer to a customer visited directly by a vehicle, or not visiting a customer. In other words, there is no need to visit all customers by the vehicles. But, nonvisited customers have to be either allocated (assigned) to a customer on a tour or left isolated. VRAP aims to minimize the routing costs, allocation (assignment) costs for nonvisited customers, and costs related to isolated customers (neither visited nor allocated). In addition to VRAP, several sub-problems for the class of maximal covering non-Hamiltonian problems have been introduced in the literature as follows:

- Maximal Covering Tour Problem (MCTP) (Current & Schilling, 1994), a special version of MTP, in which the assignment objective is to maximize the total demand (coverage as much as possible) while satisfying the pre-specified maximum travel distance from a tour stop.
- Time-constrained Maximal Covering Salesman Problem (TCMCSP) (Dast-

mardi, Mohammadi & Naderi, 2020; Naji-Azimi & Salari, 2014; Ozbaygin, Yaman & Karasan, 2016), which aims to find a cycle visiting a subset of customers with an upper bound on the traveled time while maximizing the total satisfied demands.

- Time-constrained Maximal Covering Routing Problem (TCMCRP) (Amiri & Salari, 2019; Sinnl, 2021), which Generalizes TCMCSP to find the K cycles (routes). In both TCMCSP and TCMCRP, customers are either visited, or covered, or left isolated.

- Budgeted Prize-collecting Covering Sub-graph Problem (BPCCSP), introduced in the unpublished work of Morandi, Leus & Yaman (2021), in which each customer can be visited, covered, or not covered, and the objective is to find a connected sub-graph while maximizing the total collected profit and covered customers and satisfying the constraints on the length of the sub-graph and coverage capacity of each customer.

- Covering Traveling Salesman Problem with Profit (CTSPP) (Jiang et al., 2022), which aims to find a Hamiltonian cycle visiting the most profitable customers or covering the customers as much as possible by parcel lockers while not violating a maximum budget on the tour length.

- Distance-constrained Generalized Covering Traveling Salesman Problem (DGCTSP) (Singh et al., 2021), which extends the GCSP by finding minimum-cost m covering tours and considering an upper bound on the distance of each tour.

- Routing Mobile Medical Facilities (RMMF) (Yücel, Salman, Bozkaya & Gökalp, 2020), which is a variant of TOP where prizes (profits) come from covered customers' locations. Also, it minimizes the total route cost while fully or partially covering the customers according to their closeness to a mobile facility.

- Bi-objective Maximal Covering Minimal Tour Problem (BCTP) (Goldani, 2020), which is very close to CTP and aims at finding a cycle while minimizing the route cost and maximizing the covered demands. In BCTP, there is a subset of customers who must be on the tour and a subset of customers from which the covered customers are chosen.

- Close-enough Vehicle Routing Problem (CEVRP) (Mennell, 2009), Close-enough Arc Routing Problem (CEARP) (Ha, Bostel, Langevin & Rousseau, 2014), Set Orienteering Problem (SOP) (Archetti, Carrabs & Cerulli, 2018), Close-enough Orienteering Problem (CEOP) (Štefaníková, Váňa & Faigl, 2020), Informative Path Planning (IPP) (Binney & Sukhatme, 2012), etc (for full explanation on these problems or related ones see the review work of Glock & Meyer (2022)).

2.4 Comparison of the problem of thesis with related works

To clarify the differences between the proposed problem with the existing problems in the literature, Table 2.3 is presented (according to the ten characteristics provided in section 2.3.2.1). Note that in Table 2.3, the maximal covering non-Hamiltonian problems are not considered in the comparison with the present work. Moreover, Table 2.4 represents the comparison between the non-Hamiltonian problems with profits plus maximal covering non-Hamiltonian problems with the present work. Note that in Table 2.4, the objective function ($\mathcal{O}_5$) is the maximization of the collected prize (profit) which is just considered in the non-Hamiltonian problems with profits and maximal covering non-Hamiltonian problems. Also, the constraint $\mathcal{TW}$ is removed since it has not been considered in the related problems. Moreover, Table 2.5 compares the present work with the most-related problems in the literature with their references and solution approaches. In Table 2.5, to prevent the Table from lengthening, the references for the problems with sign * are not provided completely, so the reader is referred to their related description in previous sections of this chapter. Moreover, in Table 2.5, if in a problem there exist some customers that must be visited or must be covered, they are shown with words "M_v" or "M_c", respectively; otherwise, if it is not mandatory to visit or cover a subset of customers, then it is shown with word "O" in the column of service approach. Also, in the column of delivery point, the word "C" means that the customer nodes are used as the delivery points, and the word "S" means that the Steiner nodes (a kind of customer with 0 demand) or potential delivery sites are used as delivery points, and finally, the word "C/S" means that the delivery points (stopping locations) are chosen among both customer and Steiner points. Also, to illustrate the differences of the addressed problem, CE-VRP (see chapter 3), Fig. 2.1 is provided which shows how the main problem of this thesis is reduced or converted to the other related problems in the literature. In this figure, the notations are defined as follows: m is the number of vehicles, Q is the load capacity, T is the maximum route time, D is the maximum route distance, d_{max} is the maximum walking/coverage distance for the customers/delivery sites, and s is the service time on nodes.

According to Fig. 2.1, CE-VRP can be reduced to three sub-problems including CoVRP (by removing load capacity and maximum route time constraints), Cm-RSP (by removing maximum route length and time and considering some customers as the Steiner points (nodes)), and DGVRP (by removing maximum route time and assuming that the clusters are known in advance with no maximum walking distance). Also, CoVRP reduces to m-CTP if some customers are considered as the Steiner points and a subset of customers must be visited or covered. Moreover, Cm-RSP reduces to three sub-problems including MTP (by not considering the Steiner points, considering an upper bound on the number of stops en-route, removing the load capacity constraint while routing with only one vehicle), RSP (by removing load capacity and Steiner points and routing with a single vehicle), and VRDAP (by assuming that it is mandatory to visit/cover a subset of the customers). Moreover, RSP reduces to CSP if the Steiner points are included among the nodes, and RSP also becomes another version, MCP, if the maximum walking distance is included as a constraint in the model. In addition, DGVRP reduces to Distance-constrained CVRP (DCVRP) if the clusters are removed i.e., the maximum walking distance is zero. And DCVRP reduces to the well-known CVRP after removing the maximum route distance.

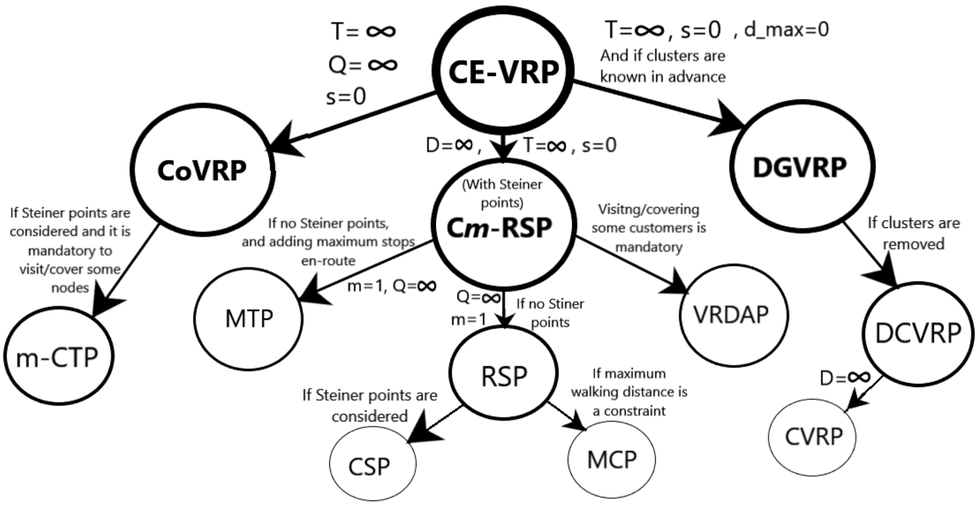


Figure 2.1 The relationship between the addressed problem in the thesis, CE-VRP, with other most-related problems in the literature

Table 2.3 Comparison of the most-related Hamiltonian and set covering non-Hamiltonian problems with the present work

Problem	$\mathcal{D}$	$\mathcal{C}$	$\mathcal{S}$	$\mathcal{D}_o$	$\mathcal{D}_m$	$\mathcal{C}_o$	$\mathcal{C}_m$	$\mathcal{S}_o$	$\mathcal{S}_m$	$\mathcal{LC}$	$\mathcal{DV}$	$\mathcal{TV}$	$\mathcal{TW}$	$\mathcal{MC}$	$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3$	$\mathcal{O}_4$	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
CVRP	g	d	a		✓		✓		✓	✓					✓					✓	✓	
DCVRP	g	d	a		✓		✓		✓	✓	✓				✓					✓	✓	
TCVRPTW	g	d	a		✓		✓		✓			✓	✓		✓					✓	✓	
CluVRP	g	d, g	a		✓	✓			✓	✓				✓	✓					✓	✓	
MCP (2.3.2.3)	d	b, d, g, h	a		✓	✓			✓					✓	✓				✓		✓	
RSP (2.3.2.3)	d	b, d, g, h	a		✓	✓			✓						✓	✓			✓		✓	
LSCP (2.3.2.2)	a	b	a, f		✓		✓	✓			✓			✓			✓					✓
MTP (2.3.2.3)	d	b, d, g, h	a		✓	✓			✓						✓	✓			✓		✓	
SRSP (2.3.2.3)	d	b	a, g, h		✓		✓	✓							✓	✓			✓		✓	
CSP (2.3.2.4)	d	b, d, g, h	a, g, h		✓	✓		✓						✓	✓				✓		✓	
CTP (2.3.2.4)	d	b, d, g, h	a, g, h		✓	✓	✓	✓						✓	✓				✓		✓	
m-CTP (2.3.2.4)	g	b, d, g, h	a, g, h		✓	✓	✓	✓			✓			✓	✓					✓	✓	
mm-CTP (2.3.2.4)	a, d, g	b, d, g, h	a, g, h	✓		✓	✓	✓			✓			✓	✓			✓		✓		✓
Cm-RSP (2.3.2.5)	g	b, d, g, h	a, g, h		✓	✓		✓		✓					✓	✓				✓	✓	
GVRP (2.3.2.6)	g	b, d, g, h	a		✓	✓			✓	✓				✓	✓					✓	✓	
DGVRP (2.3.2.6)	g	b, d, g, h	a		✓	✓			✓	✓	✓			✓	✓					✓	✓	
LRP (2.3.2.7)	a, d, g	d	a	✓			✓		✓	✓					✓			✓		✓		✓
VRDAP (2.3.2.8)	g	b	a, g, h		✓		✓	✓		✓					✓	✓				✓		✓
MDCTVRP (2.3.2.9)	a, d, g	b, d, g, h	a	✓		✓			✓	✓				✓	✓	✓				✓		✓
CoVRP (2.3.2.10)	g	b, d, g, h	a		✓	✓			✓		✓			✓	✓					✓	✓	
CoVRPwIT (2.3.2.10)	g	d, g	a		✓	✓			✓		✓			✓	✓					✓	✓	
Present work	g	b, d, g, h	a		✓	✓			✓	✓	✓	✓		✓	✓	✓				✓	✓	

Table 2.4 Comparison of non-Hamiltonian problems with profits and maximal covering non-Hamiltonian problems with the present work

Problem	$\mathcal{D}$	$\mathcal{C}$	$\mathcal{S}$	$\mathcal{D}_o$	$\mathcal{D}_m$	$\mathcal{C}_o$	$\mathcal{C}_m$	$\mathcal{S}_o$	$\mathcal{S}_m$	$\mathcal{LC}$	$\mathcal{DV}$	$\mathcal{TV}$	$\mathcal{MC}$	$\mathcal{O}_1$	$\mathcal{O}_2$	$\mathcal{O}_3$	$\mathcal{O}_4$	$\mathcal{O}_5$	$\mathcal{K}_1$	$\mathcal{K}_2$	$\mathcal{W}_1$	$\mathcal{W}_2$
TOP (2.3.1)	d	a, d	a		✓	✓			✓		✓							✓		✓	✓	
PCVRP (2.3.1)	d	a, d	a		✓	✓			✓		✓			✓						✓	✓	
MCPTP (2.3.1)	d	a, d	a		✓	✓			✓	✓				✓				✓		✓	✓	
LMCP (2.3.2.2)	a	a, b	f		✓	✓			✓				✓					✓				✓
MCCP (2.3.2.4)	d	a, b, d, g, h	a, g, h		✓	✓		✓					✓	✓				✓	✓		✓	
PCCTP (2.3.2.4)	d	a, b, d, g, h	a, g, h		✓	✓	✓	✓			✓		✓	✓					✓		✓	
VRAP (2.3.3.1)	g	a, b, d, g, h	a		✓	✓			✓					✓	✓			✓		✓	✓	
MCTP (2.3.3)	d	a, b, d, g, h	a		✓	✓			✓	✓	✓		✓	✓				✓	✓		✓	
TCMCSP (2.3.3)	d	a, b, d, g, h	a		✓	✓			✓			✓	✓					✓	✓		✓	
TCMCRP (2.3.3)	g	a, b	a, g, h		✓	✓		✓				✓	✓					✓		✓	✓	
CTSPP (2.3.3)	d	a, b, d, g, h	a, g, h		✓	✓		✓				✓	✓			✓		✓	✓			✓
BCTP (2.3.3)	d	a, b, d, g, h	a, g, h		✓	✓	✓	✓					✓	✓				✓	✓		✓	
Present work	g	b, d, g, h	a		✓	✓			✓	✓	✓	✓	✓	✓	✓					✓	✓	

Table 2.5 Comparison of the most-related problems in the literature with the present work

Reference	**Problem**	**Vehicle**	**Service approach**	**Delivery point**	$\mathcal{LC}$	$\mathcal{DV}$	$\mathcal{TV}$	$\mathcal{MC}$	$\mathcal{O}_1$	$\mathcal{O}_2$	**Solution approach**
(Current & Schilling, 1989)	CSP*	Single	O	C/S				✓	✓		Heuristic
(Current & Schilling, 1994)	MTP*	Single	O	C					✓	✓	Heuristic
(Labbé et al., 1999)	MCP*	Single	O	C				✓	✓		Branch-and-cut
(Labbé et al., 1999)	RSP*	Single	O	C					✓	✓	Branch-and-cut
(Hachicha et al., 2000)	m-CTP*	Multi	$O/M_c/M_v$	C/S		✓		✓	✓		Heuristic
(Baldacci et al., 2007)	Cm-RSP*	Multi	O	C/S	✓				✓	✓	Branch-and-cut
(Ghoniem et al., 2013)	VRDAP	Multi	M_c	S	✓				✓	✓	CG
Solak et al. (2014)	VRDAP	Multi	M_c	S	✓				✓	✓	Benders decomposition
(Mattila, 2016)	DGVRP	Multi	O	C	✓	✓		✓	✓		Heuristic
(Reihaneh & Ghoniem, 2017b)	VRDAP	Multi	M_c	S	✓				✓	✓	Multi-start heuristic
(Reihaneh & Ghoniem, 2019)	VRDAP	Multi	M_c	S	✓				✓	✓	Branch-and-price
(Buluc et al., 2021)	CoVRP	Multi	O	C		✓		✓	✓		Heuristic
Present work	CE-VRP	Multi	O	C	✓	✓	✓	✓	✓	✓	Enhanced heuristic

3. PROBLEM DESCRIPTION AND FORMULATION

In this section, a new set covering non-Hamiltonian problem, called as Covering Electric Vehicle Routing Problem (CE-VRP), which tries to find the optimal routes and location for delivery sites while visiting or covering all customers and satisfying the capacity, traveled time and distance constraints, is presented. In CE-VRP each customer must be either visited directly by a vehicle (be on a tour or a cycle) or assigned to a customer, which is on the tour (cycle or route). In other words, a vehicle delivers the products to a customer either by visiting him/her directly in his/her place or by waiting for him/her to come to the allocated (assigned) delivery site. Furthermore, CE-VRP can be seen as a routing-clustering problem in which the customers are first clustered and then the centroid of each cluster (delivery site) is found with respect to the minimum routing cost between the centroids and the minimum assignment cost within that cluster (the closest node, according to Euclidean distance, to the remaining nodes of the cluster). Also, each vehicle has a maximum capacity Q, maximum travelling length D, and maximum travelling duration T, which limits the number of served customers by that cycle (vehicle).

Hence, CE-VRP is a variant of CVRP in which vehicles choose a location to stop and wait for customers to deliver their products besides clustering the customers to make a trade-off between vehicle travel cost and customer service accessibility (assignment cost). Moreover, CE-VRP extends the well-known GVRP (Ghiani & Improta, 2000) by assuming that the clusters are unknown and there is no a priori knowledge of the nodes within the clusters. Also, CE-VRP extends the Cm-RSP (Baldacci et al., 2007) by assuming a maximum tour length and time duration on each cycle (tour or ring). In addition, in CE-VRP, there is no Steiner point ($|W| = \emptyset$) unlike the Cm-RSP. Furthermore, CE-VRP is different from VRDAP (Reihaneh & Ghoniem, 2019) in the sense that the potential delivery sites are chosen from the customers' place in CE-VRP and the mathematical models of the proposed CE-VRP and VRDAP have different decision variables and constraints. To clarify the explained problem, a numerical example is given as follows: suppose that in an example of CE-VRP, there are 8 customers and 1 depot. One possible solution of

CE-VRP could be that customers 1, 3, 4, 7, and 8 are selected as the delivery sites, or they can cover the other customers. On the other hand, customers 2, 5, and 6 are not chosen as the delivery sites, so they must be covered by the customers which are delivery sites (Fig 3.1). Fig. 3.1 indicates that customers 1 and 2 are covered by customer 1 (if customer i covers the customer i, this means the customer i is a delivery site), customer 3 is a delivery site, and covers only itself, customers 4, 5, and 6 are assigned to the customer 4, and finally, customers 7 and 8 are chosen as the delivery sites which do not cover any customer except themselves. Moreover, in this example, the first vehicle, or route, starts its trip from the depot (0) and visits customers 1 and 3 and then returns to the depot (0); again the second vehicle leaves the depot and visits the customer 4 and returns to the depot (0); finally, the third vehicle leaves the depot (0), visits the customers 7 and 8 and reruns to the depot (0). All these three routes must satisfy the load capacity, route distance, and route duration constraints.

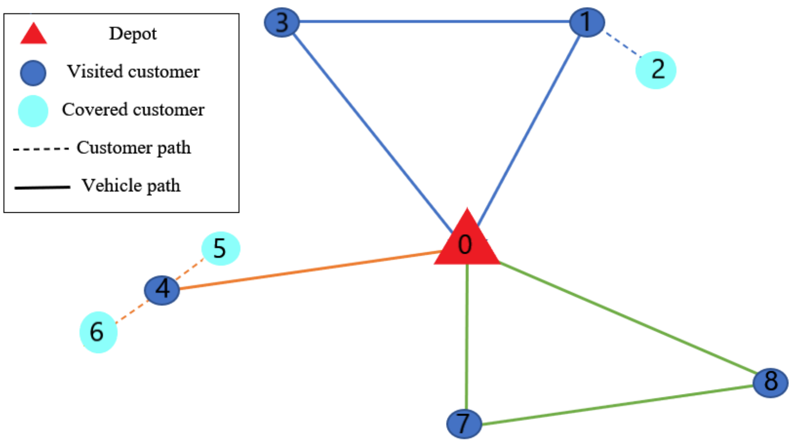


Figure 3.1 The graph of the example for CE-VRP

3.1 Mathematical model for CE-VRP

The CE-VRP can be defined on an undirected graph $G = (V, E)$, where $V = \{0, 1, 2, ..., n\}$, a set of nodes (vertices) including the customers $\{1, 2, ..., n\}$, and a depot 0. Also, E is the set of edges linking each pair of nodes $i, j \in V$. A fleet of homogeneous *autonomous delivery electric vehicles* is situated at the depot, where the vehicles start and end their trip while serving the customers. Each route has the maximum distance (D) and maximum duration (T). For each edge $[i, j] \in E$, a non-negative travel cost d_{ij}, an assignment cost c_{ij} are defined (For the sake of simplicity, assume that both c_{ij} and d_{ij} are the distances between two nodes $i, j \in V$). Each customer has a certain amount of demand q_i ($i \in V \setminus \{0\}$) with a fixed service time of s. The total demand of each route (cycle) is equal to the sum of the demands of its customers. Moreover, the walking distance for each customer to reach a delivery site has an upper bound of d_{max}. Also, each *autonomous delivery vehicle* has a capacity Q and a fixed (utilization) cost F. The objective function of CE-VRP is to minimize the total traveled distances for both customers (assignment or intra-cluster cost) and vehicles (route or inter-cluster cost) and minimize the number of used vehicles. The sets, parameters and decision variables of CE-VRP are provided in Table 3.1.

Table 3.1 Notation and parameters of CE-VRP

Sets:	
V	Set of nodes (customers and depot),$V = \{0,1,...,n\}$, $\{0\}$ is (home) depot
Parameters:	
n	Total number of customers
q_i	Demand of customer $i \in V \setminus \{0\}$
d_{ij}	The route cost (distance) between pair of nodes $i, j \in V$
c_{ij}	The assignment cost (distance) between pair of customers $i, j \in V \setminus \{0\}$
t_{ij}	The travel time between pair of nodes $i, j \in V$
s_i	Service time at node $i \in V \setminus \{0\}$
Q	Cargo capacity of each vehicle
F	Fixed usage (utilization) cost of vehicle
D	Driving range of each vehicle (maximum tour length)
T	Maximum planning horizon (maximum tour time)
d_{max}	Maximum walking distance to stopping location
Decision variables:	
x_{ij}	1 if the vehicle travels from customer i to customer j; 0 otherwise
z_{ij}	1 if a customer i is assigned to (covered by) the customer j; 0 otherwise
w_{ij}	The total travelled distance from the depot to the node j, where i is the predecessor of the node j
w'_{ij}	The total travelled time from the depot to the node j, where i is the predecessor of the node j
u_i	The remaining cargo for a vehicle before visiting the node i

Two flow-based MIP models are proposed for CE-VRP. In the first model, CE-VRP-1, the assignment cost is included in the objective function; on the other hand, in the second model, CE-VRP-2, the assignment cost is removed from the objective function, but an upper bound for the traveled distance for the assigned customers is considered as the constraint. In both proposed models, an efficient flow-based (arc-based) formulation for distance and time-constrained routes is considered from the literature, which is computationally efficient (Almoustafa, 2013; Kara, 2011) and outperforms the formulations proposed by Golden, Magnanti & Nguyen (1977); Kulkarni & Bhave (1985); Laporte, Nobert & Desrochers (1985); Li, Simchi-Levi & Desrochers (1992). Therefore, the first flow-based MIP model, CE-VRP-1, is presented as follows:

$$\text{(3.1)} \qquad Min \quad \sum_{i,j \in V} d_{ij} x_{ij} + \sum_{i,j \in V \setminus \{0\}} c_{ij} z_{ij} + \sum_{j \in V \setminus \{0\}} F x_{0j}$$

s.t.,

$$\text{(3.2)} \qquad \sum_{j \in V \setminus \{0\}} z_{ij} = 1, \quad \forall i \in V \setminus \{0\}$$

$$\text{(3.3)} \qquad z_{ij} \leq z_{jj}, \quad \forall i \in V \setminus \{0\}, \quad \forall j \in V \setminus \{0\}$$

$$\text{(3.4)} \qquad \sum_{j \in V, j \neq i} x_{ij} = z_{ii}, \quad \forall i \in V \setminus \{0\}$$

$$\text{(3.5)} \qquad \sum_{i \in V} x_{ii} = 0$$

$$\text{(3.6)} \qquad \sum_{i \in V} x_{ik} - \sum_{j \in V} x_{kj} = 0, \quad \forall k \in V \setminus \{0\}$$

$$\text{(3.7)} \qquad 0 \leq u_i \leq u_j - \sum_{k \in V \setminus \{0\}} q_k z_{kj} + Q(1 - x_{ij}), \quad \forall i, j \in V \setminus \{0\}$$

$$\text{(3.8)} \qquad \sum_{j \in V} (w_{ij} - w_{ji}) = \sum_{j \in V} d_{ij} x_{ij}, \quad \forall i \in V \setminus \{0\}$$

$$\text{(3.9)} \qquad w_{0i} = d_{0i} x_{0i}, \quad \forall i \in V \setminus \{0\}$$

$$w_{ij} \leq (D - d_{j0})x_{ij}, \quad \forall i,j \in V, j \neq 0 \tag{3.10}$$

$$w_{ij} \geq (d_{ij} + d_{0i})x_{ij}, \quad \forall i,j \in V, i \neq 0 \tag{3.11}$$

$$w_{i0} \leq Dx_{i0}, \quad \forall i \in V \setminus \{0\} \tag{3.12}$$

$$\sum_{j \in V} (w'_{ij} - w'_{ji}) = \sum_{j \in V} (t_{ij} + s_i)x_{ij}, \quad \forall i \in V \setminus \{0\} \tag{3.13}$$

$$w'_{0i} = t_{0i}x_{0i}, \quad \forall i \in V \setminus \{0\} \tag{3.14}$$

$$w'_{ij} \leq (T - t_{j0})x_{ij}, \quad \forall i,j \in V, j \neq 0 \tag{3.15}$$

$$w'_{ij} \geq (t_{ij} + t_{0i} + s_i)x_{ij}, \quad \forall i,j \in V, i \neq 0 \tag{3.16}$$

$$w'_{i0} \leq Tx_{i0}, \quad \forall i \in V \setminus \{0\} \tag{3.17}$$

$$x_{ij}, z_{ij} \in \{0,1\}, \quad \forall i,j \in V \tag{3.18}$$

In the proposed MIP for CE-VRP-1, the objective function (3.1) minimizes the total travel and assignment cost in the network and the number of used *autonomous delivery vehicles*. Constraints (3.2) ensure that each customer is allocated to exactly one stopping customer location (delivery site). Constraints (3.3) guarantee that service from a stopping customer location at a node is provided only if there exists a stopping customer location at that node. Constraints (3.4) ensure that an *autonomous delivery vehicle*, stopped at the delivery site, can visit another delivery site or return to the depot. Constraint (3.5) ensures that there must not be a route from a node to itself. Constraints (3.6) ensure that the number of *autonomous delivery vehicles* entering every delivery site (stopping customer location) and the depot must be equal to the number of *autonomous delivery vehicles* leaving. The capacity and sub-tour elimination constraint are ensured by constraints (3.7). Constraints (3.8)-(3.12) ensure that the traveled distance for each vehicle (tour or cycle length) to reach the customer stopping locations must not exceed a specific upper bound (D). Constraints (3.13)-(3.17) ensure that the travelled time for each vehicle during the tour must not exceed a specific upper bound (T). Constraints (3.18) indicate the type of decision variables of the problem.

Moreover, the second flow-based MIP model, CE-VRP-2, is presented as follows:

$$Min \quad \sum_{i,j\in V} d_{ij}x_{ij} + \sum_{j\in V\setminus\{0\}} Fx_{0j} \tag{3.19}$$

s.t., constraints (3.2)-(3.18), and

$$\sum_{j\in V\setminus\{0\}} c_{ij}z_{ij} \leq d_{max}, \forall i \in V \setminus \{0\} \tag{3.20}$$

In the proposed CE-VRP-2, objective function (3.19) minimizes the route cost and the number of used vehicles. Also, constraints (3.20) ensure that the travel distance of each customer to reach the delivery site must not be greater than a specific upper bound (d_{max}).

In the proposed models, when d_{max} is large enough, the problem reduces to the 1-Median problem; and when d_{max} is lower than the minimum distance, the problem reduces to CVRP. Moreover, for the first model, CE-VRP-1, the number of constraints and decision variables are $O(n^2)$ and $O(n^2)$, respectively. Also, the second model, similar to the first model, has $O(n^2)$ constraints and decision variables. Therefore, both proposed models are polynomial in the number of constraints and decision variables. In the next chapter, the proposed heuristic method to solve the large-sized instances of CE-VRP is introduced.

4. SOLUTION PROCEDURE

In this chapter, the methodology of the proposed solution method is explained which is designed to obtain the solutions with high quality for the main problem of the present thesis where the exact solvers are not able to find the optimal for large-sized instances.

4.1 Variable Neighbourhood Search Algorithm

VNS, first introduced by Mladenović & Hansen (1997), is among the popular meta-heuristics due to its simplicity and effectiveness. Its main power is in the "systematic change of neighborhood within a local search algorithm" (Mladenović & Hansen, 1997). Several different versions for VNS are provided in the literature as follows: (1) Variable Neighborhood Descent (VND) (Chen, Huang & Dong, 2010), which changes the neighbourhoods in a deterministic way; (2) Reduced Variable Neighborhood Search (RVNS) (Xiao, Kaku, Zhao & Zhang, 2011), which applies a stochastic way (*shaking*) in each neighbourhood; (3) Basic VNS (BVNS) (Hansen, Mladenović, Brimberg & Pérez, 2019), which combines both deterministic and stochastic mechanisms (the algorithm of BVNS in the literature is given in Algorithm 4 in Appendix A, in which three sub-functions including *Shake* (moving to a neighbour solution in a randomized way), $First\_Improvement$ function, and $Neighborhood\_Change$ function are embedded, and k_{max} is the number of neighbourhood mechanisms); (4) General Variable Neighborhood Search (GVNS) (Mladenović, Dražić, Kovačevic-Vujčić & Čangalović, 2008), which uses the VND as the local search within the BVNS; (5) Skewed Variable Neighborhood Search (SVNS) (Macedo, Alves, Hanafi, Jarboui, Mladenović, Ramos & De Carvalho, 2015), which diversifies the solution of the neighborhoods which are not close to the current (incumbent) solution; (6) Variable Neighborhood Decomposition Search (VNDS)(Hansen, Mladenović & Perez-

Britos, 2001), which divides the VNS into two levels by decomposing the problem and extends the BVNS.

VNS has been proposed for various covering-routing problems in the literature; for instance, Pérez et al. (2003) proposed a Variable Neighbourhood Tabu Search to solve the MCP instances. Their hybrid proposed metaheuristic applied standard moves and shakes for VNS part and TS as the local search to avoid the local optima solutions according to the Tabu list. Moreover, (Salari et al., 2010) proposed a VNS-based solver combined with IP-based improvement heuristic for Cm-RSP. In addition, Naji-Azimi et al. (2012) has improved and developed the method in the work by Salari et al. (2010) by designing a VNS-based solver enhanced with several improvement operators and IP-based improvement method. Also, they used a *Shaking* function to explore the solution space and then exploit by applying a *local search* consisting of swap and *Extraction & Assignment* improvement heuristics, ILP-based Procedure, Modified Assignment Problem (to find the optimal assignment of the customers to the other customers on the route or *Steiner* nodes), and Lin–Kernighan TSP Procedure (Lin & Kernighan, 1973) (to decrease the routing costs).

4.2 Simulated Annealing Algorithm

Simulated Annealing (SA) was first introduced by Kirkpatrick, Gelatt Jr & Vecchi (1983) and according to the reports in the literature, it is an efficient metaheuristic in solving the combinatorial optimization problems (Aarts & Van Laarhoven, 1989). SA is among the local search-based metaheuristics, which generate a *single solution* and improve it during the optimization process. Also, SA exploits the neighborhood by local search operators which are designed specifically for each problem and explores the solution space by accepting worse solutions at the first iterations of the algorithm. Accepting or rejecting worse solution also helps the SA to escape from the local minima or maxima points (for more details on SA see (Aarts, Korst & Michiels, 2005; Bertsimas & Tsitsiklis, 1993; Van Laarhoven & Aarts, 1987)). The pseudo-code of the general SA is given in Algorithm 5 (see Appendix A), in which T_{max}, T_{min}, $F(T)$, S_0, and $f(S)$ are the maximum (initial) temperature, the minimum (final) temperature, cooling function, initial solution, and fitness function, respectively.

4.3 The proposed two-phase heuristic for CE-VRP

When the parameters are set as $T = \infty$, D_∞, $s = 0$, and $d_{max} = 0$, CE-VRP reduces to the well-known CVRP, which is proven to be an NP-hard problem. As a result, the addressed problem, CE-VRP, is also an NP-hard problem based on the computational complexity theory. In this section, due to the NP-hardness of the CE-VRP, a new two-phase heuristic algorithm consisting of selecting the delivery sites (the first phase) and routing the vehicles (the second phase) is proposed to solve the large-sized CE-VRP instances in an acceptable time. In the first phase, a greedy *Repair_Improvement* approach with perturbation moves is applied to select the delivery sites and assign the unvisited customers to the delivery sites. Next, in the second phase, a hybrid Variable Neighbourhood Search with Simulated Annealing (VNS-SA) algorithm is proposed to find the routes visiting the delivery sites. The proposed VNS-SA for finding the routes is composed of the BVNS (for shaking, diversification, and a systematic neighbourhood change) with SA (for intensification) as its local search. As mentioned in the previous section, VNS is an efficient metaheuristic in solving combinatorial problems including TSP, and CVRP; however, VNS needs a local search heuristic to find the high-quality solutions within the solution space (Pekel & Kara, 2019). Moreover, SA is an efficient metaheuristic for searching the neighbourhood effectively by applying both intensification and diversification mechanisms. Accordingly, combining the VNS with SA (considering SA as a local search heuristic within the VNS) will result in a high-performance solution procedure that can benefit from SA's capability in escaping from the local optimum and VNS's efficiency in searching the solution space and its neighbourhood in a systematized manner.

The pseudo-code of the proposed two-phase heuristic is given in Algorithm 1. Also, the pseudo-code of the proposed VNS-SA for the second phase is given in Algorithm 2 (notations defined for the proposed two-phase heuristic are given in Table **??**). In the proposed two-phase heuristic, the *Repair_Improvement*(.) function is followed by the *Perturbation*(.) function because it is possible that *Repair_Improvement*(.) function may not improve the solution quality, so perturbing the current solution is needed to explore the solution space (Naji-Azimi et al., 2010). Also, the proposed *Repair_Improvement*(.) algorithms are executed sequentially in the order that at first *improvement* operators are executed one by one and, finally, the *repair* operator is performed (see section 4.3.3 for more details).

Table 4.1 Notations of the proposed two-phase heuristic

Notation	**Definition**
N	Total number of iterations of VNS-SA
M	The number of iterations at each temperature
N'	The maximum number of iterations of the two-phase method
k'_{max}	Total number of perturbation moves on matrix $\mathcal{R}$
α	Cooling rate
m_{max}	The number of move operators
k_{max}	Total number of shaking moves on $\mathcal{RO}$

Algorithm 1 The pseudo-code of the proposed two-phase heuristic for CE-VRP

Require: k'_{max}, N'

$S \leftarrow S_0$ $\triangleright$ S_0 is constructed by the greedy construction algorithm (see Appendix A)
$n \leftarrow 1$
while $n \leq N'$ **do**
    $k' \leftarrow 1$
    while $k' \leq k'_{max}$ **do**
        <First phase starts>
        $S' \leftarrow Repair\_Improvement(S)$
        $S'' \leftarrow Perturbation(S', k')$ $\triangleright$ $Perturbation(.)$ moves are defined in section 4.3.4.1
        <First phase ends>
        <Second phase starts>
        $S''' \leftarrow VNS\_SA(S'')$ $\triangleright$ $VNS\_SA(.)$ is given in Alg. 2
        <Second phase ends>
        $k' \leftarrow k' + 1$
    end while
    $n \leftarrow n + 1$
end while

Algorithm 2 The proposed VNS-SA algorithm in the second phase ($VNS\_SA(.)$)

Require: $S, N, M, T_{max}, T_{min}, \alpha, k_{max}$

$n \leftarrow 1$
while $n \leq N$ **do**
  $k \leftarrow 1$
  while $k \leq k_{max}$ **do**
    $S' \leftarrow Shake\_VNS(S,k)$ ▷ $Shake\_VNS(.)$ moves are defined in section 4.3.4.2
    `<SA (local search) starts>`
    $T \leftarrow T_{max}$
    while $T \geq T_{min}$ **do**
      $t \leftarrow 1$
      while $t \leq M$ **do**
        $S'' \leftarrow First\_Improvement(S')$ ▷ $First\_Improvement$ is given in Alg. 3
        $t \leftarrow t+1$
      end while
      $T \leftarrow \alpha * T$ ▷ Temperature is updated
    end while
    `<SA (local search) ends>`
    $Neighborhood\_Change(S, S'', k)$
  end while
  $n \leftarrow n+1$
end while

The steps of the proposed two-phase heuristic for CE-VRP are briefly described in the following.

4.3.1 Solution representation

To encode a solution of the instance of CE-VRP, the following representation is used: $\mathcal{R}$ is a n-row matrix indicating that the customer i $(1 \leq i \leq n)$ is selected as a delivery site or not; thus, $\mathcal{R}$ is a n-row matrix with binary elements e.g., $r_i \in \{0,1\}$. $\mathcal{X}$ is a n-row matrix indicating the the customer i $(1 \leq i \leq n)$ is assigned to (covered by) the customer j $(1 \leq j \leq n)$; thus, $\mathcal{X}$ is a n-row matrix with bounded integer elements e.g., $x_i \in \{1,2,...,n\}$. $\mathcal{RO}$ is a $(2n+1)$-row matrix indicating the routes of

Algorithm 3 *First_Improvement* algorithm in the second phase (VNS-SA)

Require: S, P ▷ P: the maximum number of opportunities that is given to *First_Improvement* to accept a solution

$r \leftarrow 0$
$p \leftarrow 0$
while $(r < 1)$ & $(p < P)$ **do**
  $S' \leftarrow Neighbourhood(S)$ ▷ $Neighbourhood(.)$ is the move operators applied on matrix $\mathcal{RO}$ (see section 4.3.4) plus the lines 8-30 in the Algorithm 7
  if $f(S') < f(S)$ **then**
    $S \leftarrow S'$
    $r \leftarrow r+1$
  else
    Accept S' with the probability $e^{(f(S)-f(S')/T}$
    $r \leftarrow r+1$
  end if
  $p \leftarrow p+1$
end while

each vehicle, so its elements are either 0 or an integer value between 1 and n (coding of the routes for MCP by Pérez et al. (2003)). Also, it is obvious that this solution representation, or encoding, can be used for CVRP by considering only the matrix $\mathcal{RO}$.

To clarify the explained representation, a numerical example is given as follows: suppose that in an example of CE-VRP, there are 8 customers and 1 depot (Fig. 3.1). A solution of this instance could be represented by the following row matrices, or vectors:

$$\mathcal{R} : \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$

The matrix $\mathcal{R}$ indicates that customers 1, 3, 4, 7, and 8 are selected as the delivery sites, or they can cover the other customers. On the other hand, customers 2, 5, and 6 are not chosen as the delivery sites, so they must be covered by the customers which are delivery sites.

$$\mathcal{X} : \begin{bmatrix} 1 & 1 & 3 & 4 & 4 & 4 & 7 & 8 \end{bmatrix}$$

The matrix $\mathcal{X}$ indicates that customers 1 and 2 are covered by customer 1 (if customer i covers the customer i, this means the customer i is a delivery site), customer 3 is a delivery site, and covers only itself, customers 4, 5, and 6 are assigned to the customer 4, and finally, customers 7 and 8 are chosen as the delivery sites which do not cover any customer except themselves.

$$\mathcal{RO} : \begin{bmatrix} 0 & 1 & 3 & 0 & 4 & 0 & 7 & 8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The matrix $\mathcal{RO}$ represents the routes of each vehicle. For example, the first vehicle, or route, starts its trip from the depot (0) and visits the customers 1 and 3 and then returns to the depot (0); again the second vehicle leaves the depot and visits customer 4 and returns to the depot (0); finally, the third vehicle leaves the depot (0), visits customers 7 and 8 and reruns to the depot (0). Since the Matrix $\mathcal{RO}$ is a 17-row matrix, the remaining 0's mean that there is no other route in this solution.

As mentioned above, after generating the matrices $\mathcal{R}$, $\mathcal{X}$, and $\mathcal{RO}$, they may be the infeasible solutions. Initializing the proposed algorithm, repairing the infeasible solutions, or improving them, and the local search mechanism used in the proposed VNS-SA are explained briefly in the following.

4.3.2 Initialization algorithms

To initialize the optimization process of the CVRP, or CE-VRP instances, an initial solution must be generated. Here, in the proposed VNS-SA, a feasible single solution is constructed by applying a greedy construction algorithm.

4.3.2.1 A greedy construction heuristic

The proposed greedy construction heuristic starts by finding the minimum number of delivery sites among the customers, such that, the selected customers are able to cover the nearest remaining customers considering the maximum coverage distance. If a customer cannot be covered by another customer, or it is not within the coverage distance of any customer, then it becomes a delivery site. Next, the near-optimal routes are found among the selected delivery sites by iterative *2-opt operator* satisfying the capacity, and distance/time constraints. The detailed description is provided in Appendix A. Moreover, the proposed greedy construction heuristic is a unique construction algorithm in the literature (also can be applied on CoVRP). The other approach in the literature which is used as an construction step for covering-routing problems is *Clustering algorithm* by Naji-Azimi et al. (2010) proposed for Cm-RSP. The proposed greedy construction heuristic in this thesis, unlike the *Clustering algorithm* (Naji-Azimi et al., 2010), does not find as far as customers to pick them as the delivery site; while it selects the customers which have more access to cover the other customers. The efficiency of the proposed greedy construction algorithm is analyzed in section 5.3.1.

4.3.3 *Repair_Improvement* operators

Designing a simple operator to improve or repair the solutions of the covering-routing problem like MCP, RSP, MTP, CoVRP, or CE-VRP will lead to a poor performance and weak-quality solutions (Renaud et al., 2003). Thus, a problem-specific operator must be designed to enhance the performance of the solution algorithm when there are covering and routing elements in the problem. In this section, five *improvement* and one *repair* operators are introduced to improve the feasible solution and repair the infeasible solution to reach a solution with better quality. The proposed five *improvement* operators enhance the current solution by checking if it improves the objective function or not; on the other hand, the proposed *repair* operator repairs the infeasible solution so that it satisfies the constraints of the problem such as vehicle capacity, distance and time constraints. Here, five *improvement* and one *repair* operators are called I_1, I_2, I_3, I_4, I_5, and R_1, respectively.

The first proposed *improvement* operator exchanges the role of a delivery site with one of its assigned customers (the customer which is the delivery site becomes an inactive stopping location and the assigned customer becomes a new delivery site). If after exchanging, the customers which were covered by the old delivery site are still within the coverage distance of the new active delivery site, and the length of the route, which includes the old delivery site, becomes shorter, then this *improvement* operator applies successfully and exchanges the role of the old delivery site with its assigned customer and then a new delivery site becomes active and the process is repeated for all delivery sites (the algorithm of the first *improvement* operator is given in Appendix A). The first *improvement* operator is similar to *Vertex exchange* procedure introduced by Renaud et al. (2004) and *Swap* procedure by Naji-Azimi et al. (2010).

The second *improvement* operator only applies on the delivery sites which have only one covered customer. The role of the delivery site with its one assigned customer is changed. If this change decreases the length of the tour, on which the old delivery site was located, then this *improvement* operator exchanges the role of the delivery site with its one assigned customer, and this process is repeated for all delivery sites with one assigned customer. In other words, this heuristic is a special case of the first *improvement* operator in which the delivery sites with only one covered node are found (the algorithm of the second *improvement* operator is given in Appendix

A). Also, the second *improvement* operator is the same as the *General exchange (GE)* procedure used by Pandiri et al. (2020) for CSP instances.

The third *improvement* operator finds the common customer who is within the coverage distance of two or more delivery sites (stopping locations). If after activating this common customer, the other customers, which are covered by these delivery sites, are still within the coverage distance of this common customer or other active delivery sites, then this *improvement* operator chooses this common customer as a new delivery site, and the old delivery sites, which were covering the common customer, become inactive. This process is repeated for all nonvisited customers (the algorithm of the third *improvement* operator is given in Appendix A). Also, the third *improvement* operator is close to the *Cycle augmentation* procedure presented in Renaud et al. (2003); however, the *Cycle augmentation* checks all unvisited nodes even if they are covered by a single delivery site or a customer on the route.

The fourth *improvement* operator finds the delivery sites which are "redundant"; by redundant delivery site, it means that if the delivery site becomes inactive, then its covered customers can be still covered by the other active delivery sites. So, by inactivating redundant delivery sites, the vehicles will travel less distance and this will leads to a shorter tour. This process is again applied to all remaining delivery sites (the algorithm of the fourth *improvement* operator is given in Appendix A). The fourth *improvement* operator is similar to the *Cycle reduction* procedure introduced in Renaud et al. (2004) for MCP, and *Extraction & Reassignment* procedure used in Naji-Azimi et al. (2010,1); Salari & Naji-Azimi (2012) for Cm-RSP. Also, the fourth *improvement* operator is similar to the *Steiner node removal* procedure used by Naji-Azimi et al. (2010); Zhang et al. (2014). Moreover, the fourth *improvement* procedure is close to the *Extract-assign operator* introduced by Zhang et al. (2014), but the *Extract-assign operator* extracts a subset of visited nodes with their covered nodes and then reassigns them to their best position (while considering all possible positions) in random order. In addition, the other operator close to the fourth *improvement* operator is *Delivery site removal* proposed by Reihaneh & Ghoniem (2018), which removes the redundant delivery sites and reassigns their covered nodes according to the criteria of "total estimated savings" calculated after removing that delivery site.

The fifth *improvement* operator is a special case of the fourth proposed *improvement* operator and the *Cycle_reduction* procedure introduced in Renaud et al. (2004). This *improvement* operator removes the redundant delivery sites by finding two delivery sites where at least one of them has no covered customers and both are within the maximum walking (coverage) distance from each other. Then the delivery site with no covered customer is deactivated and assigned to the second delivery site. This operator improves the solution quality by reducing the number of delivery sites that must be visited by the vehicles (the algorithm of the fifth *improvement* operator is given in Appendix A).

Finally, the *repair* operator applies to the infeasible solutions, which could be generated when a delivery site covers the demands of the customers more than the capacity of the vehicles. In other words, in *repair* operator, if a delivery site covers the demands greater than the vehicle's capacity, then no vehicle is able to visit that delivery site. So, the proposed *repair* operator searches and finds these kinds of infeasible delivery sites and makes them feasible by reducing the customers assigned to them. Then, the unassigned customers are allocated to another delivery site that has available capacity; otherwise, they become active delivery sites if no delivery site is found for them (the algorithm of the *repair* operator is given in Appendix A). It is noteworthy to say that the *improvement* and *repair* operators are only applied on the matrices $\mathcal{R}$ and $\mathcal{X}$.

4.3.4 Move operators (Neighbourhoods)

In this section, the perturbation and shaking moves used within the two-phase heuristic with neighbourhood moves are briefly explained.

4.3.4.1 Perturbation moves on matrix $\mathcal{R}$ in the first phase

For perturbing the current solution (incumbent) of matrix $\mathcal{R}$, three move operators are used. All these three move operators are applied on the matrix $\mathcal{R}$. The first perturbation, $P1$, is the simple *1-mutation* operator. It chooses one of the elements of matrix $\mathcal{R}$ randomly, and if the element is 1, then it becomes 0, and vice versa. The second perturbation, $P2$, is *1-1 adjacent exchange* move for adjacent nodes. It randomly chooses one of the elements of the matrix $\mathcal{R}$ which is 1, and if the selected element has at least one covered node, then one of its covered nodes is selected at random and then they are swapped. The third perturbation, $P3$, is *1-1 exchange* operator. It first chooses two random elements in the matrix $\mathcal{R}$, which are not adjacent to each other, and then swaps (exchanges) them. At final, the obtained new $\mathcal{R}$ generates the matrix $\mathcal{X}$ and then by *Route_Generate*(.) function in Alg. 7 (see Appendix A), the matrix $\mathcal{RO}$ is generated and sent to the second phase ($VNS\_SA(.)$ in Alg. 2).

4.3.4.2 Shaking operators on matrix $\mathcal{RO}$ in the second phase

In this section, the shaking moves used during the VNS-SA are explained. Four move operators are used for shaking the solution space of the matrix $\mathcal{RO}$, called $S'1$-$S'4$ while all these shaking operators are applied to the customers of the different routes i.e., inter-route moves. Also, these shaking moves are ordered in a way that the first shaking move shakes the solution space with small changes (smaller neighbourhood) and the last move shakes the solution space with a larger neighbourhood. The shaking moves are 1-1 exchange (swap), 1-0 node relocation (insertion), subsequences swap (equivalent to *cross-exchange* of Taillard, Badeau, Gendreau, Guertin & Potvin (1997)), and subsequences insertion operators. Moreover, the feasibility of the shaking moves on the matrix $\mathcal{RO}$ is not checked.

To clarify the move operators in the proposed VNS-SA, one example is defined and then the moves are explained on the example. Suppose a network with 1 depot $\{0\}$ and 8 customers, the current solution (matrix $\mathcal{RO}$) for the routing problem is as follows:

$$\mathcal{RO}: \begin{bmatrix} 0 & 1 & 3 & 0 & 4 & 2 & 7 & 0 & 8 & 5 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

First of all, to apply the move operators, the zero elements in the matrix $\mathcal{RO}$ are removed (it becomes a TSP solution) and then the obtained matrix is called $\mathcal{RO}'$ presented as follows:

$$\mathcal{RO}': \begin{bmatrix} 1 & 3 & 4 & 2 & 7 & 8 & 5 & 6 \end{bmatrix}$$

The first shaking ($S'1$), 1-1 exchange move, chooses two elements of the current solution ($\mathcal{RO}'$) and changes (swaps) them; by applying this move operator, the new solution ($\mathcal{RO}''$) will be as follows (assume that the nodes 3 and 8 are chosen for swapping):

$$\mathcal{RO}'': \begin{bmatrix} 1 & \mathbf{8} & 4 & 2 & 7 & \mathbf{3} & 5 & 6 \end{bmatrix}$$

The second shaking ($S'2$), 1-0 node relocation (insertion) operator, chooses one element of the current solution ($\mathcal{RO}'$) and moves it to another position; by applying this move operator, the new solution ($\mathcal{RO}''$) will be as follows (assume that the node 3 is chosen and it is moved to the fifth position):

$$\mathcal{RO}'' : \begin{bmatrix} 1 & 4 & 2 & 7 & \mathbf{3} & 8 & 5 & 6 \end{bmatrix}$$

The third shaking, subsequences swap operator ($S'3$), chooses two subsets of elements (each subset has more than 1 element) of the current solution ($\mathcal{RO}'$) and swaps them; by applying this move operator, the new solution ($\mathcal{RO}''$) will be as follows (assume that the set of nodes 3, 4, and 2 with the set of nodes 5 and 6 are chosen and swapped):

$$\mathcal{RO}'' : \begin{bmatrix} 1 & \mathbf{5} & \mathbf{6} & 7 & 8 & \mathbf{3} & \mathbf{4} & \mathbf{2} \end{bmatrix}$$

The fourth shaking ($S'4$), subsequences insertion operator, chooses a subset of elements (more than 1 element) of the current solution ($\mathcal{RO}'$) and moves it to another position; by applying this move operator, the new solution ($\mathcal{RO}''$) will be as follows (assume that the set of nodes 3, 4, and 2 are chosen and it is moved to the last position):

$$\mathcal{RO}'' : \begin{bmatrix} 1 & 7 & 8 & 5 & 6 & \mathbf{3} & \mathbf{4} & \mathbf{2} \end{bmatrix}$$

4.3.4.3 Move operators on matrix $\mathcal{RO}$ in the local search (SA) of the second phase

After shaking a solution, the neighbourhood of the current solution must be searched efficiently to find a near-optimal solution in its local solutions. The local search (SA) embedded in the VNS-SA uses seven move operators including two intra-route ($M1$-$M2$), and five two-adjacent-route moves ($M3$-$M7$). Two intra-routes are 1-1 exchange (swap) and 2-opt operators. Assume that the current solution is the following matrix:

$$\mathcal{RO} : \begin{bmatrix} 0 & 1 & 3 & 0 & 4 & 2 & 7 & 0 & 8 & 5 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Assume that the third route is chosen randomly to apply the intra-route move operators. The sub-matrix of the matrix above (one route) is obtained as follows:

$$\mathcal{RO}' : \begin{bmatrix} 4 & 2 & 7 \end{bmatrix}$$

And now the 1-1 exchange and 2-opt operates are applied on this route. The 1-1 exchange move is explained in the previous section. The 2-opt operator ($M2$) chooses two elements of the current sub-matrix ($\mathcal{RO}'$) and reverses all elements between them including themselves; by applying this move operator, the new solution ($\mathcal{RO}''$) will be as follows (assume that the customers 4 and 7 are chosen and reversed):

$$\mathcal{RO}'' : \begin{bmatrix} \mathbf{7} & \mathbf{2} & \mathbf{4} \end{bmatrix}$$

The next five two-adjacent-routes move operators are 2-opt ($M3$), subsequences swap ($M4$), subsequences insertion ($M5$), reversed subsequences swap ($M6$), and reversed subsequence insertion ($M7$) operators. The moves $M3$-$M5$ are described before. Now the last two moves $M6$ and $M7$ are explained with the example. Assume that the current solution is the following matrix:

$$\mathcal{RO} : \begin{bmatrix} 0 & 1 & 3 & 0 & 4 & 2 & 7 & 0 & 8 & 5 & 0 & 6 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

And assume that the second and third routes which are two-adjacent-routes are chosen. The sub-matrix after removing the zero elements will be obtained as follows:

$$\mathcal{RO}' : \begin{bmatrix} 4 & 2 & 7 & 8 & 5 \end{bmatrix}$$

The sixth move operator ($M6$), reversed subsequences swap operator chooses two subsets of elements of the current solution ($\mathcal{RO}'$) and swaps them and also reverses all elements within each set; by applying this move operator, the new solution ($\mathcal{RO}''$) will be as follows (assume that the subset of the customers 4 and 2 with the subset of the customers 8 and 5 are chosen and then swapped and finally reversed):

$$\mathcal{RO}'' : \begin{bmatrix} \mathbf{5} & \mathbf{8} & 7 & \mathbf{2} & \mathbf{4} \end{bmatrix}$$

The seventh move operator ($M7$), reversed subsequence insertion operator, chooses one subset of elements of the current solution ($\mathcal{RO}'$) and inserts it in another position and also reverses all elements within the set; by applying this move operator, the new solution ($\mathcal{RO}''$) will be as follows (assume that the subset of customers 4 and 2 are chosen to insert in the last position and then reversed):

$$\mathcal{RO}'' : \begin{bmatrix} 7 & 8 & 5 & \mathbf{2} & \mathbf{4} \end{bmatrix}$$

Finally, after applying each seven move operators on the matrix $\mathcal{RO}$ in the local search (SA), the obtained matrix becomes feasible with lines 8-30 presented in the Algorithm 7 (see Appendix A).

4.3.5 Handling the constraints of the problem

4.3.5.1 Capacity constraint

The load capacity of each vehicle, vehicle, is satisfied during the VNS-SA in two places: one in the *repair* operator (Alg. 13 in Appendix A) and the other one with the lines 8-30 of *Route_Generate*(.) function (Alg. 7 in Appendix A) within the VNS_SA(.), Alg. 2, after moving to the new neighbourhoods.

4.3.5.2 Maximum walking distance constraint

This constraint is first satisfied with line 3 in the greedy construction heuristic (Alg. 6 in Appendix A) and then it is kept from being violated by the four *improvement* operators (Algs. 8-11 in Appendix A) while assigning the customers to the delivery sites.

4.3.5.3 Distance-constrained routes

This constraint is satisfied with the lines 8-30 in the *Route_Generate* algorithm (Alg. 7 in Appendix A) within the VNS_SA(.), Alg. 2, after moving to the new neighbourhoods.

4.3.5.4 Time-constrained routes

This constraint is also satisfied with the lines 8-30 in the *Route_Generate* algorithm (Alg. 7 in Appendix A) within the VNS_SA(.), Alg. 2, after moving to the new neighbourhoods.

4.3.6 Fitness function

The fitness function is the summation of total routing costs, and total usage cost of the vehicles. The fitness function in the $Neighborhood\_Change$(.) function is calculated by the following steps:

1.1 $f = 0$; (f is the value of the fitness function);

1.2 For ($i = 0$ to $2N - 1$) do $\{f = f + d_{ro_i, ro_{i+1}}\}$ (N is the number of customers, d_{ij} is the distance between nodes i, j, ro_i is the i-th element of the matrix $\mathcal{RO}$);

1.3 $f = f + F \times K$ (K is the number of vehicles used);

1.4 Return f;

However, within the $VNS\_SA(.)$, after moving to a new feasible solution, then the new solution is compared with the current (incumbent) solution to accept or reject it (lines 12-15 in the VNS_SA algorithm, Alg. 2). To avoid time-consuming computations, the comparison between the two solutions is only done by calculating the saving values obtained after applying a move. If the saving value is positive, then the new solution is accepted; on the other hand, the new solution is accepted according to a given probability (line 15 in Alg. 2).

5. COMPUTATIONAL STUDY

In this chapter, computational results and experiments related to the performance of the proposed two-phase heuristic are explained briefly. Moreover, parameter tuning, comparison with exact solver (Gurobi) over the small-sized instances, performance evaluation over the medium and large-sized instances, and sensitivity analysis for the proposed two-phase heuristic are conducted. All experiments and runs are performed on a laptop with features of Intel Core, 1.60 GHz, and 16 GB RAM. Also, the Gurobi solver is executed on the Anaconda navigator Spyder environment written in Python. The proposed metaheuristic is coded in C++ programming Language compiled with Dev-C++ software.

5.1 Instance generation for CE-VRP

To evaluate the performance of the proposed heuristic and validate the results, three datasets are generated for the addressed problem, CE-VRP, in this thesis. The value chosen for each parameter determines the complexity of an instance; moreover, choosing an inappropriate value will result in an infeasible solution or redundant parameter. Here, the parameters Q (vehicle capacity), d_{max} (maximum walking distance), D (maximum route length), and T (maximum route time) must be chosen as a value that is applicable in generating the CE-VRP instances. For example, the load capacity Q must be greater than the minimum demand and lower than the sum of the total demands; otherwise, it will not be a CVRP instance. d_{max} must be greater and lower than the minimum and maximum distance between the customers, respectively; otherwise, it will not be a covering-routing problem.

To validate the mathematical model developed and analyze the performance of the proposed heuristic for the CE-VRP, three datasets are generated in this thesis. The first dataset (small-sized instances) are generated based on 6 CVRP instances including *A-n32-k5*, *A-n33-k5*, *B-n31-k5*, *P-n16-k8*, and *P-n19-k2* with different values of n (number of customers), Q, s, T, and d_{max}. 20 small-sized instances with several customers from 8 to 33 are generated for the first dataset. This dataset is able to evaluate the performance of the different solvers when the new parameters related to covering problem are changed, but the distance matrix is fixed. The features of the instances of the first dataset are given in Table B.1 in Appendix B.

In the second generated dataset for the CE-VRP, unlike the first dataset, most of the parameters related to the covering problem are fixed, but the matrix distance (VRP part) are changed. This dataset is created based on the CVRP instances of the set A provided by Augerat et al. (1995). The distance between the nodes, n, and Q are the values of the original instance; however, the new parameters are added to each instance of set A to generate the second dataset of the CE-VRP. Therefore, a total of 20 medium and large-sized instances are generated. The features of the instances of the second dataset are given in Table B.2 in Appendix B.

The third dataset is based on the data related to a real case study of covering-routing problem originally provided by Kara & Savaser (2018) which is related to one of the cities in Turkey, Burdur, with 45 villages (customers). In these instances, the maximum walking distance (coverage distance), and the speed of the vehicles are considered as 50 or 90 min. and 1 km/h, respectively. The distance matrix between the customers and the number of the customers of the original data is not changed while the new parameters including Q, s, and T are added. Also, to include the load capacity constraint to the instances of the third dataset, the demand of each customer is set 1. The modified dataset is called the third dataset of the CE-VRP instances. The features of the instances of the third dataset are given in Table B.3 in Appendix B. Furthermore, it is noteworthy that in all generated datasets, the assignment cost, distances between the nodes, and travel time between nodes are equal to each other and have the same value as the distance matrix in the instances.

5.2 Parameter tuning of the proposed two-phase heuristic

In this section, the parameters of the proposed two-phase heuristics are tuned over the ten CE-VRP instances with various sizes chosen from the three datasets (*A-n10-50.cevrp*, *A-n10-100.cevrp*, *A-n39.cevrp*, *A-n45.cevrp*, *A-n54.cevrp*, *A-n60.cevrp*, *A-n64.cevrp*, *A-n69.cevrp*, *KS_1*, *KS_2*). The parameters to be tuned are as follows:

maximum temperature (T_{max}), minimum temperature (T_{min}), cooling rate (α), the maximum number of iterations as the stopping criterion of two-phase method (N'), the maximum number of iterations as the first stopping criterion of VNS-SA (N), and the maximum number of iterations as the second stopping criterion of VNS-SA (M). To tune a parameter, it gets values from a range while other parameters are remained fixed. A value of the parameter, which causes the objective function to have minimum value, is considered as a proper value for that parameter.

First of all, the parameter of maximum (initial) temperature is tuned. The range of $[1, 1000]$ is considered for T_{max} while the other parameters have the fixed value as follows: $T_{min} = 0.1$, $\alpha = 0.98$, $N = 100$, and $M = 10$. The best value for the parameter of T_{max}, by which the objective function is minimized, is found for every instance in this experiment and also each instance is executed 10 times for each value of T_{max}. According to the best-found value of T_{max} in each instance, T_{max} is assigned a value by which the proposed two-phase method returns the minimum objective function in most instances. If in one instance, there is more than one proper value for T_{max} the value leading to lower execution time is selected.

Similar process to tune the T_{max} is conducted for other parameters. The range of the parameter T_{min} is $[0.001, 0.1]$ while the other parameters have the fixed values of $T_{max} = 10$, $\alpha = 0.98$, $N' = 100$ $N = 10$, and $M = 10$. Also, the values of parameter α is chosen from the range of $[0.9, 0.99]$ while other parameters are fixed with values of $T_{max} = 10$, $T_{min} = 0.01$, $N' = 100$ $N = 10$, and $M = 10$. In addition, the range of the parameter N' is set as $[10, 1000]$ while other parameters are fixed as $T_{max} = 10$, $T_{min} = 0.01$, $\alpha = 0.98$, $N = 10$, and $M = 10$. Moreover, the value of the parameter N is chosen from the range of $[10, 5000]$ and other parameters have the fixed values of $T_{max} = 10$, $T_{min} = 0.01$, $\alpha = 0.98$, $N' = 100$ and $M = 10$. For the last parameter, M is assigned a value from the range of $[5, 200]$ while other parameters are assigned a fixed value of $T_{max} = 10$, $T_{min} = 0.01$, $\alpha = 0.98$, $N' = 100$ and $N = 10$. Finally, the best tuned values for the parameters of the proposed two-phase method are given in Table 5.1.

Table 5.1 Values of the two-phase heuristic parameters after parameter tuning

	T_{max}	T_{min}	α	N'	N	M
Value of parameter after parameter tuning	10	0.01	0.98	500	10	10

5.3 Performance evaluation of the proposed two-phase heuristic

To evaluate the performance of the proposed two-phase heuristic, the instances of five problems including CVRP, TCVRP, CoVRP (introduced by Buluc et al. (2021)), Time-constrained capacitated CoVRP (TCCVRP), and CE-VRP (Problem addressed in the present thesis) are solved by two-phase heuristic and the results are compared with the existing solvers in the literature and Gurobi solver, which is designed to solve mixed-integer, integer linear programming models. The relationship among the above problems is given in Fig. 5.1. In this figure, the parameters near the arrows indicate how the main problem, CE-VRP, is reduced to the other problems, and also compared to the other instances mentioned, CE-VRP is the only problem which has the assignment cost in its objective function. Also, it is noteworthy to say that the objective function of the CoVRP, and TCCVRP have no assignment cost; on the other hand, the addressed problem in this thesis, CE-VRP, has the assignment cost in its objective function. According to Fig 5.1, CE-VRP reduces to CoVRP by removing capacity and route time constraints. Also, CE-VRP reduces to TCVRP by removing the route distance and walking distance constraints. Then, TCVRP reduces to CVRP if time-constrained route are removed. Also, CE-VRP reduces to another new problem in the literature, TCCVRP, after removing the distance-constrained routes assumption i.e., $D = \infty$.

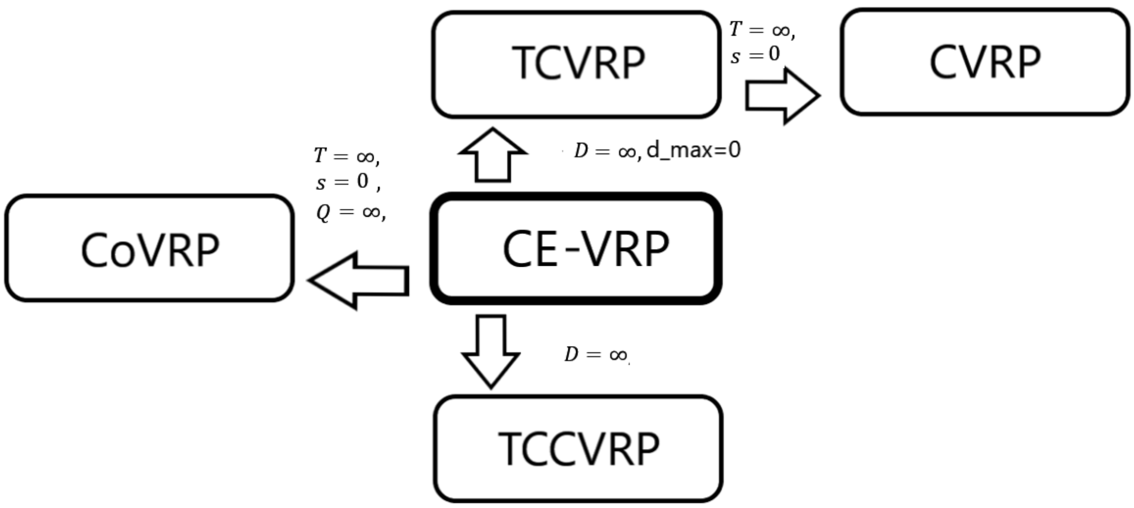


Figure 5.1 The relationship among the various addressed problems in the present thesis

5.3.1 Investigating the performance of the proposed greedy construction algorithm

To evaluate the performance of the proposed greedy construction heuristic in the first phase of the two-phase method, its performance is compared with the solution returned by the Gurobi-Python interface over the first dataset (see Appendix B) for both TCCVRP and CE-VRP. The comparisons between the results of the greedy construction algorithm and the objective function found by the Gurobi solver over the instances of TCCVRP and CE-VRP are given in Tables 5.2 and 5.3, respectively. In Tables 5.2 and 5.3, OFV, *Gap*%, Δ%, and t indicate the best-found objective function value, the gap returned by Gurobi-Python interface, the gap between the best-found solutions by Gurobi solver and greedy construction heuristic, and the execution time in seconds, respectively. To use the instances of the first dataset for TCCVRP, the value of parameter D (maximum route length) is considered as a very large number e.g., ∞. The results show that the proposed greedy construction heuristic, despite its simplicity and using no *improvement* operator, is able to generate an initial solution that has a lower than 2% gap with the optimal solution in lower than 0.5 seconds over TCCVRP instances (even it finds the optimal solution of the instance of *A-n15-100.tccvrp*). Also, the greedy construction heuristic has an acceptable performance over CE-VRP instances by proving the solutions within the gap of 5% to the optimal solutions on average for small and medium-sized instances.

Table 5.2 Computational results of solving the instances of the first dataset by the proposed greedy construction heuristic for TCCVRP

Instance	**Gurobi**			**Greedy construction heuristic**		
	OFV	*Gap*%	t	OFV	Δ%	t
A-n9-50.tccvrp	3427.97	0.00	63.61	3461.57	0.98	0.13
A-n9-100.tccvrp	2305.27	0.00	134.30	2323.16	0.78	0.19
A-n10-50.tccvrp	3427.97	0.00	590.39	3437	0.26	0.08
A-n10-100.tccvrp	2305.27	0.00	1023.71	2314.23	0.39	0.10
A-n15-100.tccvrp	2374.88	41.69	7200.00	2374.88	0.00	0.13
A-n20-100.tccvrp	3464.66	60.58	7200.00	3486.81	0.64	0.14
A-n32-100.tccvrp	5611.66	76.16	7200.00	5665.71	0.96	0.18
A-n33-100.tccvrp	5484.86	77.36	7200.00	5497.48	0.23	0.19
B-n8-100.tccvrp	2306.27	0.00	28.95	2331.79	1.11	0.08
B-n9-100.tccvrp	2302.39	0.00	136.07	2331.79	1.28	0.07
B-n10-100.tccvrp	2302.39	0.00	1956.15	2345.19	1.86	0.09
B-n12-100.tccvrp	2302.39	44.55	7200.00	2369.19	2.90	0.05

Continued on next page

Table 5.2 Continued from previous Table

Instance	Gurobi			Greedy construction heuristic		
	OFV	*Gap%*	*t*	*OFV*	Δ%	*t*
B-n8-90.tccvrp	2306.27	0.00	11.76	2338.04	1.38	0.10
B-n9-90.tccvrp	2302.39	0.00	173.62	2358.77	2.45	0.43
B-n10-90.tccvrp	2302.39	0.00	1757.77	2365.92	2.76	0.07
B-n12-90.tccvrp	2302.39	44.60	7200.00	2379.50	3.35	0.08
P-n10-35.tccvrp	6246.91	0.00	0.22	6271.47	0.39	0.07
P-n12-35.tccvrp	6271.81	0.00	42.00	6309.68	0.60	0.14
P-n10-160.tccvrp	2051.87	0.00	669.17	2092.68	1.99	0.08
P-n9-160.tccvrp	2051.87	0.00	44.01	2092.68	1.99	0.06
Average	3172.59	17.25	2491.62	3207.37	1.31	0.12

Table 5.3 Computational results of solving the small-sized instances of the first dataset by the proposed greedy construction heuristic for CE-VRP

Instance	Gurobi			Greedy construction heuristic		
	OFV	*Gap%*	*t*	*OFV*	Δ%	*t*
A-n9-50.cevrp	3427.97	0.00	74.21	3494.76	1.95	0.21
A-n9-100.cevrp	2305.27	0.00	62.55	2356.35	2.22	0.27
A-n10-50.cevrp	3427.97	0.00	483.75	3489.4	1.79	0.18
A-n10-100.cevrp	2305.27	0.00	384.16	2366.63	2.66	0.26
B-n8-100.cevrp	2306.27	0.00	16.73	2437.06	5.67	0.11
B-n9-100.cevrp	2302.39	0.00	129.49	2459.15	6.81	0.14
B-n10-100.cevrp	2302.38	0.00	1541.64	2466.31	7.12	0.17
B-n12-100.cevrp	2302.38	44.72	7200.00	2498.87	8.53	0.14
B-n8-90.cevrp	2306.27	0.00	42.68	2437.06	5.67	0.12
B-n9-90.cevrp	2302.39	0.00	220.22	2479.89	7.71	0.11
B-n10-90.cevrp	2302.38	0.00	2008.39	2487.04	8.02	0.18
P-n10-35.cevrp	6246.90	0.00	0.32	6290.68	0.70	0.17
P-n12-35.cevrp	6271.81	0.00	10.83	6328.89	0.91	0.19
P-n10-160.cevrp	2051.86	0.00	1224.31	2220.37	8.21	0.12
P-n9-160.cevrp	2051.86	0.00	180.54	2193.45	6.90	0.11
Average	2947.55	2.98	905.32	3360.37	4.99	0.21

5.3.2 Investigating the performance of the proposed *improvement* operators for CE-VRP

To analyze the performance of five *improvement* operators used in the first phase of the proposed two-phase method, six versions of the two-phase heuristics are compared with each other. These six versions are presented as follows:

- *TPH_0*: Two-phase heuristic without any *improvement* operator;
- *TPH_1*: Two-phase heuristic with only the first *improvement* operator;
- *TPH_2*: Two-phase heuristic with only the second *improvement* operator;
- *TPH_3*: Two-phase heuristic with only the third *improvement* operator;
- *TPH_4*: Two-phase heuristic with only the fourth *improvement* operator;
- *TPH_5*: Two-phase heuristic with only the fifth *improvement* operator;

Table 5.4 shows the solution results of the CE-VRP instances by six versions of the two-phase method (*TPH_0*-*TPH_5*) and original two-phase heuristic (with all *improvement* operators) over the second dataset. According to Table 5.4, all *improvement* operators, used in the first phase of the two-phase method, improve the solution quality returned by the two-phase method without any *improvement* operator, or *TPH_0*, which shows the validity and efficiency of the used *improvement* operators. Also, the fourth *improvement* operator is more efficient compared to the other operators; however, the fifth operator has the best performance in three instances. In Table 5.4, OFV, t, and Imp% are the average of the objective function values in ten runs, execution time in seconds (which are rounded to the nearest integer number), and improvement percentage which is calculated as follows (f: the objective function returned by the *TPH_0*, f': the objective function returned by one of the versions *TPH_1* to *TPH_5*):

$$Imp\% = ((f - f')/f) \times 100$$

5.3.3 Solution of CVRP and TCVRP by the proposed two-phase heuristic

To solve the instances of well-known CVRP, the proposed two-phase heuristic is modified to solve the problem in which the maximum walking distance (d_{max}) be-

Table 5.4 Comparison of the different proposed *improvement* operators over the CE-VRP instances in the second dataset (execution times are in seconds)

Instance	*TPH_0*		*TPH_1*			*TPH_2*			*TPH_3*			*TPH_4*			*TPH_5*			Original *TPH*		
	OFV	t	OFV	t	Imp.&	OFV	t	Imp.&	OFV	t	Imp.&	OFV	t	Imp.&	OFV	t	Imp.&	OFV	t	Imp.&
A-n34.cevrp	5612.89	29	5587.39	31	0.45	5604.68	34	0.14	5584.87	30	0.49	5551.25	20	**1.09**	5590.01	26	0.41	5498.64	18	2.03
A-n36.cevrp	5604.33	30	5597.09	32	0.12	5601.2	30	0.05	5600.45	30	0.06	5556.21	18	**0.85**	5599.46	25	0.08	5544.79	16	1.06
A-n37.cevrp	5455.02	27	5433.46	34	0.39	5421.93	28	0.60	5420.61	34	0.63	5368.25	13	**1.59**	5422.09	22	0.60	5326.7	12	2.35
A-n38.cevrp	5572.95	32	5551.38	32	0.38	5546.06	30	0.48	5513.72	28	1.06	5508.2	20	**1.16**	5540.38	23	0.58	5459.19	18	2.04
A-n39.cevrp	5668.03	31	5631.71	35	0.64	5630.28	43	0.66	5636.27	31	0.56	5613.42	15	0.96	5604.66	28	**1.11**	5484.71	22	3.23
A-n44.cevrp	6751.1	32	6719.58	36	0.46	6734.34	33	0.24	6705.14	33	0.68	6688.59	30	**0.92**	6729.37	28	0.32	6661.04	28	1.33
A-n45.cevrp	7781.5	46	6826.96	49	12.26	6885.64	48	11.51	6831.81	45	12.20	6720.87	33	**13.63**	6856.27	43	11.89	6685.79	40	14.08
A-n46.cevrp	7732.31	47	7720.51	54	0.15	7730.03	46	0.02	7718.9	49	0.17	7703.17	36	0.37	7676.78	40	**0.72**	7647.28	35	1.09
A-n48.cevrp	7955.76	42	7934.52	49	0.26	7891.97	42	0.80	7942.32	46	0.16	7817.88	41	**1.73**	7899.32	39	0.70	7800.57	34	1.95
A-n53.cevrp	7892.05	50	7884.45	61	0.09	7867.30	53	0.31	7857.90	54	0.43	7777.55	25	**1.45**	7891.86	45	0.01	7720.56	23	2.17
A-n54.cevrp	8036.44	51	8017.15	57	0.24	8005.03	50	0.39	8013.97	53	0.27	7970.79	26	**0.81**	8035.85	50	0.01	7893.88	27	1.77
A-n55.cevrp	9912.81	58	9847.86	52	0.65	9882.92	48	0.30	9857.75	53	0.55	9881.86	36	0.31	9834.99	51	**0.78**	9833.42	38	0.80
A-n60.cevrp	10236.40	43	10226.80	54	0.09	10169.70	44	0.65	10225.40	56	0.10	10128.01	33	**1.05**	10229.60	37	0.06	10105.90	41	1.27
A-n61.cevrp	10919.70	61	10917.10	52	0.02	10067.30	67	7.80	10008.50	49	8.34	9877.23	41	**9.54**	10886.40	38	0.30	9877.23	42	9.54
A-n62.cevrp	9217.09	47	9188.17	54	0.31	9158.75	49	0.63	9160.41	48	0.61	9002.92	34	**2.32**	9167.14	47	0.54	8942.17	33	2.98
A-n63.cevrp	10687.22	56	10554.90	58	1.23	10522.30	47	1.54	10595.71	47	0.85	10462.90	41	**2.09**	10569.40	42	1.10	10420.50	47	2.49
A-n64.cevrp	10292.78	48	10246.01	53	0.45	10247.40	49	0.44	10288.90	49	0.03	10124.20	39	**1.63**	10271.60	43	0.20	10106.20	42	1.81
A-n65.cevrp	10176.55	51	10090.80	61	0.84	10110.90	48	0.64	10152.20	45	0.23	9961.94	40	**2.10**	10136.80	36	0.39	9952.59	49	2.20
A-n69.cevrp	10034.7	55	10002.5	60	0.32	9944.01	53	0.90	9993.93	53	0.40	9913.17	37	**1.21**	10006.30	44	0.28	9835.09	54	1.98
A-n80.cevrp	11764.70	71	11757.60	79	0.06	11735.22	76	0.25	11730.20	66	0.29	11562.90	61	**1.71**	11589.50	63	1.48	11550.40	79	1.82
Average	8365.21	45.35	8286.79	49.65	0.97	8237.84	45.90	1.40	8241.94	44.95	1.41	8159.56	31.95	**2.33**	8276.88	38.50	1.08	8117.33	34.90	2.90

comes zero, and the parameters D and T get a very large number, and s becomes zero i.e., the first phase of the proposed two-phase heuristic is disabled to solve the CVRP instances. Here, four well-known CVRP datasets are solved by the proposed two-phase heuristic to validate their performance. Tables C.1-C.4 (see Appendix C) show the computational results of solving the instances of the set A, B, P, and CMT (CVRP instances), respectively. In all of these results, the two-phase heuristic is executed 10 times, and the best objective function (*Best*) and the average of the objective functions (*Average*) are reported with their best computational time in seconds. Moreover, sets A, B, and P are first provided by Augerat et al. (1995), and the CMT dataset is introduced by Christofides et al. (1979). All CVRP instances and their optimal or best-known solutions are available in the CVRPLIB database[1]. Also, the results of the two-phase heuristic over CVRP instances are compared with the existing solvers in the literature as given in Tables D.1-D.4 (see Appendix C) for sets A, B, P and CMT (CVRP instances), respectively. The computational results show that the proposed two-phase heuristic is efficient in solving the various CVRP instances with different sizes and reaching the best-known solutions in most of the instances. Also, the proposed two-phase heuristic outperforms the other solvers of the literature in terms of solution quality which shows its efficiency and competitiveness. Therefore, according to the results, the proposed local search (VNS-SA), which finds the routes between the delivery sites, is validated and is an efficient local search in the second phase of the two-phase heuristic.

If in the previous problem, CVRP, the constraint of the maximum route time is added, then the other variant, called TCVRP, appeared. One famous benchmark instances for TCVRP is the instances introduced by Christofides et al. (1979), in which instances CMT 6-10 and CMT 13-14 have the maximum duration for each tour (T) and the service time (s) for each node. Here, the computational results of solving all CMT instances including CVRP and TCVRP by the proposed two-phase heuristic and the comparison between the proposed solver and the solution methods in the literature are given in Tables C.5 and D.5 (see Appendix C). The results show the efficiency and validity of the proposed two-phase heuristic in solving the various CVRP and TCVRP instances.

[1] http://vrp.atd-lab.inf.puc-rio.br/index.php/en/

5.3.4 Comparison of the proposed two-phase heuristic with Gurobi solver over small and medium-sized instances

In this section, the performance of the proposed two-phase method and Gurobi solver (in the Python interface) are compared over the small and medium-sized instances for CoVRP, TCCVRP, and CE-VRP. In the following, after shortly describing the problems, the computational results of the two-phase method and Gurobi solver are presented.

If in the CE-VRP the capacity and time-constrained conditions are relaxed and the objective function is changed to minimize the route cost among visited customers, then CE-VRP becomes CoVRP, which was introduced by Buluc et al. (2021). In CoVRP, there are two constraints including the distance-constrained routes, and the maximum walking distance for each customer to reach a delivery site. In Table 5.5, the comparison of the results between the proposed two-phase heuristic and the Gurobi solver is presented (Time execution is limited to 7200 seconds for Gurobi). The instances of CoVRP used in this thesis as the benchmark are the third dataset of CE-VRP (see Appendix B) with $s = 0$, $T = \infty$, and $Q = \infty$. In these instances, the maximum coverage distance, and the speed of the vehicles are considered as 50 or 90 min. and 1 km/h, respectively. In Table 5.5, n, m, D, d_{max}, OFV, $Gap\%$, $\Delta\%$, and t are the number of nodes, fleet size, maximum route length, maximum walking (coverage) distance, the objective function value, the gap returned by Gurobi, the gap between the objective functions of the two-phase heuristic and Gurobi, and the execution time in seconds, respectively. Table 5.5 shows that the proposed two-phase heuristic has found the optimal solution in 12 of 14 CoVRP instances and all solutions returned by it have zero gaps with the Gurobi solver in lower than 75 seconds on average; which shows the efficiency of the proposed heuristic in solving the CoVRP instances.

Moreover, the performance of the proposed two-phase heuristic is evaluated by comparing its solutions against the results of the Gurobi-Python interface on TCCVRP instances. Here, the two-phase heuristic is implemented on a new version of CE-VRP, called TCCVRP, which is a reduced version of the CE-VRP by relaxing the distance-constrained routes; in other words, TCCVRP considers only four constraints including time-constrained routes, capacity, and maximum walking distance. It is noteworthy to say that the TCCVRP has not been studied in the literature in terms of both mathematical models and heuristics.

Table 5.6 shows the solution results of the TCCVRP instances (the first dataset in Appendix B) by the proposed two-phase heuristic and Gurobi solver. Table 5.6

Table 5.5 Comparison of computational results of the proposed two-phase heuristic with Gurobi solver on the instances of CoVRP

Instance features					Gurobi			Two-phase heuristic		
Name	**n**	**m**	D	d_{max}	OFV	$Gap\%$	t	OFV	$\Delta\%$	t
KS_1	45	2	1200	90	2091.50	0.00	630.54	2091.50	0.00	52
KS_2	45	2	1300	90	2071.26	2.61	7200.00	2071.26	0.00	73
KS_3	45	2	1500	90	1869.29	0.00	68.74	1869.29	0.00	31
KS_4	45	2	1700	90	1856.70	0.00	251.37	1856.70	0.00	49
KS_5	45	2	1500	50	2670.80	0.00	630.54	2670.80	0.00	66
KS_6	45	2	1700	50	2669.53	0.00	1227.49	2669.53	0.00	91
KS_7	45	3	1000	90	2533.37	0.00	564.78	2533.37	0.00	106
KS_8	45	3	1050	90	2270.40	0.00	353.35	2270.40	0.00	56
KS_9	45	3	1100	90	2270.40	0.00	947.87	2270.40	0.00	129
KS_10	45	3	1200	90	2166.86	0.00	679.86	2166.86	0.00	88
KS_11	45	3	1200	50	2916.27	0.00	452.31	2916.27	0.00	74
KS_11	45	3	1300	50	2844.35	0.00	599.33	2844.35	0.00	68
KS_13	45	4	1000	50	3607.07	0.00	1628.26	3607.07	0.00	85
KS_14	45	4	1050	50	3341.32	3.32	7200.00	3341.32	0.00	71
Average					2512.79	0.42	1574.19	2512.79	0.00	74.21

shows that the proposed two-phase method has reached the solution returned by Gurobi (most of them are optimal due to gap% of zero) over TCCVRP instances. Also, the two-phase heuristic has solved the instances with various sizes in about 19 seconds on average which shows the efficiency of the proposed heuristic in solving TCCVRP instances.

Table 5.6 Computational results of solving the instances of the first dataset by the proposed two-phase heuristic for TCCVRP

Instance	Gurobi			Two-phase heuristic		
	OFV	$Gap\%$	t	OFV	$\Delta\%$	t
A-n9-50.tccvrp	3427.97	0.00	63.61	3427.97	0.00	3
A-n9-100.tccvrp	2305.27	0.00	134.30	2305.27	0.00	14
A-n10-50.tccvrp	3427.97	0.00	590.39	3427.9	0.00	15
A-n10-100.tccvrp	2305.27	0.00	1023.71	2305.27	0.00	2
A-n15-100.tccvrp	2374.88	41.69	7200.00	2374.88	0.00	0.32
A-n20-100.tccvrp	3464.66	60.58	7200.00	3464.66	0.00	4
A-n32-100.tccvrp	5611.66	76.16	7200.00	5611.66	0.00	83
A-n33-100.tccvrp	5484.86	77.36	7200.00	5484.86	0.00	121
B-n8-100.tccvrp	2306.27	0.00	28.95	2306.27	0.00	0.29
B-n9-100.tccvrp	2302.39	0.00	136.07	2302.39	0.00	2
B-n10-100.tccvrp	2302.39	0.00	1956.15	2302.39	0.00	1

Continued on next page

Table 5.6 Continued from previous Table

Instance	Gurobi			Two-phase heuristic		
	OFV	*Gap%*	*t*	*OFV*	Δ%	*t*
B-n12-100.tccvrp	2302.39	44.55	7200.00	2302.39	0.00	14
B-n8-90.tccvrp	2306.27	0.00	11.76	2306.27	0.00	0.63
B-n9-90.tccvrp	2302.39	0.00	173.62	2302.39	0.00	5
B-n10-90.tccvrp	2302.39	0.00	1757.77	2302.39	0.00	27
B-n12-90.tccvrp	2302.39	44.60	7200.00	2302.39	0.00	44
P-n10-35.tccvrp	6246.91	0.00	0.22	6246.91	0.00	1
P-n12-35.tccvrp	6271.81	0.00	42.00	6271.81	0.00	13
P-n10-160.tccvrp	2051.87	0.00	669.17	2051.87	0.00	18
P-n9-160.tccvrp	2051.87	0.00	44.01	2051.87	0.00	3
Average	3172.59	17.25	2491.62	3172.59	0.00	18.56

Finally, the performance of the proposed two-phase method is compared with the results by the Gurobi solver over CE-VRP (the main addressed problem in the present thesis). In this experiment, the small-sized instances of the first dataset are solved by both the two-phase method and the Gurobi (Table 5.7). The results show that the proposed method is able to reach the optimal solutions in about 22 seconds on average for small-sized CE-VRP instances.

Table 5.7 Computational results of solving the small-sized instances of the first dataset by the proposed two-phase heuristic for CE-VRP

Instance	Gurobi			Two-phase heuristic		
	OFV	*Gap%*	*t*	*OFV*	Δ%	*t*
A-n9-50.cevrp	3427.97	0.00	74.21	3427.97	0.00	5
A-n9-100.cevrp	2305.27	0.00	62.55	2305.27	0.00	8
A-n10-50.cevrp	3427.97	0.00	483.75	3427.97	0.00	30
A-n10-100.cevrp	2305.27	0.00	384.16	2305.27	0.00	28
B-n8-100.cevrp	2306.27	0.00	16.73	2306.27	0.00	1
B-n9-100.cevrp	2302.39	0.00	129.49	2302.39	0.00	7
B-n10-100.cevrp	2302.38	0.00	1541.64	2302.38	0.00	52
B-n12-100.cevrp	2302.38	44.72	7200.00	2302.38	0.00	82
B-n8-90.cevrp	2306.27	0.00	42.68	2306.27	0.00	2
B-n9-90.cevrp	2302.39	0.00	220.22	2302.39	0.00	22
B-n10-90.cevrp	2302.38	0.00	2008.39	2302.38	0.00	39
P-n10-35.cevrp	6246.90	0.00	0.32	6246.9	0.00	1

Continued on next page

Table 5.7 Continued from previous Table

Instance	**Gurobi**			**Two-phase heuristic**		
	OFV	*Gap%*	*t*	*OFV*	$\Delta\%$	*t*
P-n12-35.cevrp	6271.81	0.00	10.83	6271.81	0.00	1
P-n10-160.cevrp	2051.86	0.00	1224.31	2051.86	0.00	41
P-n9-160.cevrp	2051.86	0.00	180.54	2051.86	0.00	9
Average	2947.55	2.98	905.32	2947.55	0.00	21.86

5.3.5 Solution of the large-sized instances of CE-VRP by the proposed two-phase method

In this section, the large-sized instances of CE-VRP in all three generated instances are solved by the proposed two-phase heuristic. Also, to verify efficiency of the hybrid VNS-SA in the second phase, two other versions of the second phase including solo VNS and solo SA. These two versions with the original version are called *TPH_VNS*, *TPH_SA*, and *TPH_VNS-SA*, respectively (for algorithms of *TPH_VNS* and *TPH_SA* see Appendix A). Tables 5.8-5.10 show the computational results obtained by these three versions of the two-phase heuristic method over the large-sized instances of the first and seconds dataset and real data (third dataset), respectively. Moreover, in Tables 5.8-5.10, columns of *best*, *Avg.*, t, $\Delta_1\%$ and $\Delta_2\%$ are the best objective function value returned by the related solver in 10 runs, average of the objective function values, the average execution time in seconds, the gap between the solutions of *TPH_VNS* and *TPH_VNS-SA*, and the gap between the solutions of *TPH_SA* and *TPH_VNS-SA*, respectively. Also, in this experiment, the parameter of *TPH_VNS-SA* are adjusted to the values by which the execution time of it is close to the execution time of the *TPH_VNS* and *TPH_SA*. The parameters of the *TPH_VNS-SA* are set as $T_{max} = 10$, $T_{min} = 0.1$, $\alpha = 0.98$, $N' = 200$, $N = 10$ and $M = 10$. Also, the parameters of *TPH_VNS* are set $N' = 500$ $N = 100$, and parameters of *TPH_SA* are set as $T_{max} = 10$, $T_{min} = 0.1$, $\alpha = 0.98$, and $N' = 200$ $M = 10$.

According to Tables 5.8-5.10, when the proposed two-phase method is applied with VNS-SA as its second phase outperforms the other two versions including the two-phase method with VNS or SA as the heuristics in the second phase. Therefore, hybrid VNS and SA is an efficient heuristic in the second phase of the two-phase method to find the near-optimal routes visiting the delivery sites.

Table 5.8 Comparison of the proposed two-phase method with two version of it over the instances of the first dataset of CE-VRP

Instance		*TPH_VNS*				*TPH_SA*				*TPH_VNS-SA*		
Name	**Optimal**[2]	*best*	*Avg.%*	Δ_1%	*t*	*best*	*Avg.%*	Δ_2%	*t*	*best*	*Avg.%*	*t*
A-n9-50.cevrp	3427.98	**3427.98**	3427.98	0.00	4	**3427.98**	3427.98	0.00	5	**3427.98**	3427.98	5
A-n9-100.cevrp	2305.27	**2305.27**	2305.27	0.00	4	**2305.27**	2305.27	0.00	5	**2305.27**	2305.27	5
A-n10-50.cevrp	3427.97	**3427.97**	3427.97	0.00	10	**3427.97**	3427.97	0.00	6	**3427.97**	3427.97	13
A-n10-100.cevrp	2305.27	**2305.27**	2305.27	0.00	10	**2305.27**	2305.27	0.00	6	**2305.27**	2305.27	14
A-n15-100.cevrp	-	**2374.88**	2374.88	0.00	13	**2374.88**	2374.88	0.00	7	**2374.88**	2374.88	13
A-n20-100.cevrp	-	**3464.66**	3465.01	0.00	14	**3464.66**	3465.02	0.00	8	**3464.66**	3464.89	16
A-n32-100.cevrp	-	5621.63	5633.42	0.00	13	5634.95	5642.77	0.24	11	**5616.05**	5627.08	15
A-n33-100.cevrp	-	**5486.59**	5496.09	0.00	14	5492.07	5506.98	0.10	11	**5486.59**	5504.17	15
B-n8-100.cevrp	2306.27	**2306.27**	2306.27	0.00	10	**2306.27**	2306.27	0.00	5	**2306.27**	2306.27	11
B-n9-100.cevrp	2302.39	**2306.27**	2302.39	0.00	11	**2306.27**	2302.39	0.00	5	**2306.27**	2302.39	11
B-n10-100.cevrp	2302.38	**2302.38**	2302.38	0.00	10	**2302.38**	2302.38	0.00	6	**2302.38**	2302.38	12
B-n12-100.cevrp	2302.38	2306.28	2307.88	0.17	11	2309.74	2312.09	0.32	7	**2302.38**	2302.38	12
B-n8-90.cevrp	2306.27	**2306.27**	2306.27	0.00	9	**2306.27**	2306.27	0.00	5	**2306.27**	2306.27	10
B-n9-90.cevrp	2302.39	**2302.39**	2302.39	0.00	10	**2302.39**	2302.39	0.00	6	**2302.39**	2302.39	10
B-n10-90.cevrp	2302.38	**2302.38**	2302.38	0.00	11	**2302.38**	2302.38	0.00	6	**2302.38**	2302.38	12
B-n12-90.cevrp	-	**2313.43**	2313.43	0.00	11	2314.60	2318.09	0.05	6	**2313.43**	2314.01	11
P-n10-35.cevrp	6246.90	**6246.90**	6246.90	0.00	1	**6246.90**	6246.90	0.00	4	**6246.90**	6246.90	5
P-n12-35.cevrp	6271.81	6285.11	6286.11	0.21	6	6286.11	6287.24	0.23	6	**6271.81**	6271.81	9
P-n10-160.cevrp	2051.86	**2051.86**	2051.86	0.00	11	**2051.86**	2051.86	0.00	6	**2051.86**	2051.86	12
P-n9-160.cevrp	2051.86	**2051.86**	2051.86	0.00	9	**2051.86**	2051.86	0.00	6	**2051.86**	2051.86	11
Average		3174.58	3175.80	0.02	9.60	3175.81	3177.31	0.05	6.35	3173.72	3175.02	11.10

Table 5.9 Comparison of the proposed two-phase method with two version of it over the instances of the second dataset of CE-VRP

Instance	*TPH_VNS*				*TPH_SA*				*TPH_VNS-SA*		
Name	*best*	*Avg.%*	Δ_1%	*t*	*best*	*Avg.%*	Δ_2%	*t*	*best*	*Avg.%*	*t*
A-n34.cevrp	5459.79	5476.01	0.61	13	5432.61	5439.03	0.11	12	**5426.61**	5438.77	16
A-n36.cevrp	5496.88	5513.88	0.48	13	5529.27	5537.11	1.08	12	**5470.38**	5511.89	17
A-n37.cevrp	5282.62	5306.53	0.02	14	5293.25	5307.79	0.23	11	**5281.30**	5298.34	16
A-n38.cevrp	5387.91	5410.11	0.72	14	5443.16	5466.79	1.75	12	**5349.51**	5486.19	17
A-n39.cevrp	5480.26	5501.33	0.14	13	5517.81	5539.22	0.83	13	**5472.53**	5487.70	17
A-n44.cevrp	6618.59	6634.55	0.34	19	6638.74	6650.64	0.65	14	**6595.95**	6612.12	21
A-n45.cevrp	6683.95	6705.69	0.12	16	6717.35	6750.11	0.62	14	**6676.01**	6702.01	20
A-n46.cevrp	7683.56	7689.03	0.25	17	7667.02	7702.85	0.04	15	**7664.28**	7688.91	19
A-n48.cevrp	7814.13	7831.49	0.58	16	7835.50	7839.69	0.85	15	**7769.44**	7784.47	20
A-n53.cevrp	7750.34	7765.01	0.09	18	7749.00	7759.66	0.07	15	**7743.40**	7759.42	22
A-n54.cevrp	7934.71	7959.66	0.86	18	7930.69	7938.79	0.80	16	**7867.37**	7908.66	22
A-n55.cevrp	9956.80	9972.29	2.16	19	9896.76	9900.50	1.54	16	**9746.51**	9860.93	23
A-n60.cevrp	10184.00	10228.39	1.15	20	10118.20	10156.71	0.49	17	**10068.70**	10109.32	26
A-n61.cevrp	10070.20	10136.52	1.30	21	10070.40	10118.22	1.30	20	**9941.42**	9968.93	26
A-n62.cevrp	9033.66	9082.03	0.52	21	9010.04	9045.44	0.25	17	**8987.22**	9017.40	27
A-n63.cevrp	10617.40	10728.30	1.83	22	10591.10	10622.09	1.58	19	**10426.10**	10506.54	25
A-n64.cevrp	10271.70	10308.33	1.58	22	10277.10	10314.41	1.63	20	**10070.1**	10160.86	28
A-n65.cevrp	10200.40	10250.66	3.38	23	10115.50	10152.91	2.52	20	**9866.91**	10007.82	26
A-n69.cevrp	9952.94	10070.39	1.03	23	9980.05	9993.09	1.31	21	**9851.33**	9868.94	27
A-n80.cevrp	11867.50	11959.89	2.27	28	11811.60	11871.22	1.79	23	**11604.20**	11634.40	30
Average	8187.36	8226.48	0.97	18.50	8181.25	8205.30	0.97	16.10	8096.06	8140.67	22.25

Table 5.10 Comparison of the proposed two-phase method with two version of it over the instances of the third dataset of CE-VRP

Instance	*TPH_VNS*				*TPH_SA*				*TPH_VNS-SA*		
Name	*best*	*Avg.%*	Δ_1%	*t*	*best*	*Avg.%*	Δ_2%	*t*	*best*	*Avg.%*	*t*
KS_1	7079.58	7105.99	0.14	14	7086.36	7119.04	0.24	12	**7069.44**	7102.09	17
KS_2	5537.51	5651.21	4.34	18	5445.59	5516.42	2.61	13	**5307.17**	5463.59	18
KS_3	5420.43	5485.95	1.27	16	5373.64	5420.19	0.39	13	**5352.50**	5405.01	19
KS_4	5423.68	5471.01	27.40	18	4324.83	5022.85	1.61	13	**4256.18**	4879.44	20
KS_5	7922.99	7929.33	4.99	18	7655.25	7695.07	1.44	14	**7546.36**	7595.07	20
KS_6	7839.95	7901.29	25.70	18	6368.47	6394.46	2.13	14	**6235.67**	6388.29	22
KS_7	9940.59	9972.05	12.00	16	8914.49	9871.92	0.45	13	**8874.57**	8903.92	17
KS_8	7209.51	7247.83	0.67	17	7187.44	7223.54	0.36	13	**7161.56**	7164.14	20
KS_9	7226.44	7278.09	1.53	15	7177.66	7221.05	0.84	12	**7117.80**	7188.02	19

Continued on next page

[2]The optimal solutions were obtained by Gurobi solver (see Table 5.7)

Table 5.10 Continued from previous Table

Instance Name	TPH_VNS				TPH_SA				TPH_VNS-SA		
	best	*Avg.%*	$\Delta_1\%$	*t*	*best*	*Avg.%*	$\Delta_2\%$	*t*	*best*	*Avg.%*	*t*
KS_10	7179.62	7226.66	0.76	15	7224.70	7231.62	1.40	13	**7125.24**	7201.33	23
KS_11	10651.50	10923.41	0.54	17	10723.60	10737.59	1.22	15	**10594.10**	10653.52	21
KS_12	10809.90	10830.54	16.90	16	9318.70	9402.85	0.73	16	**9148.01**	9252.32	22
KS_13	10809.90	10830.54	16.90	16	9318.70	9402.85	0.73	16	**9148.01**	9252.32	22
KS_14	12360.50	12902.01	3.21	18	12350.70	13005.04	3.13	16	**11975.80**	12114.99	23
Average	8243.72	8339.70	8.31	16.57	7747.86	7947.45	1.24	13.78	7651.27	7768.85	20.21

Moreover, Fig. 5.2 shows the convergence history of the various versions of the proposed two-phase method in the instances of *A-n64.cevrp*-*A-n80.cevrp*, and *KS_13*-*KS_14*, respectively, in which the horizontal axis is time in seconds, and vertical axis is the objective function value. This figure shows that the original two-phase method with hybrid VNS-SA is more efficient than the two-phase method with only either VNS or SA. Finally, as a sample, the best-found solution by the proposed greedy construction algorithm and the two-phase heuristic for the instance of *A-n32-100.cevrp* is given in Figs. 5.3-5.4, respectively.

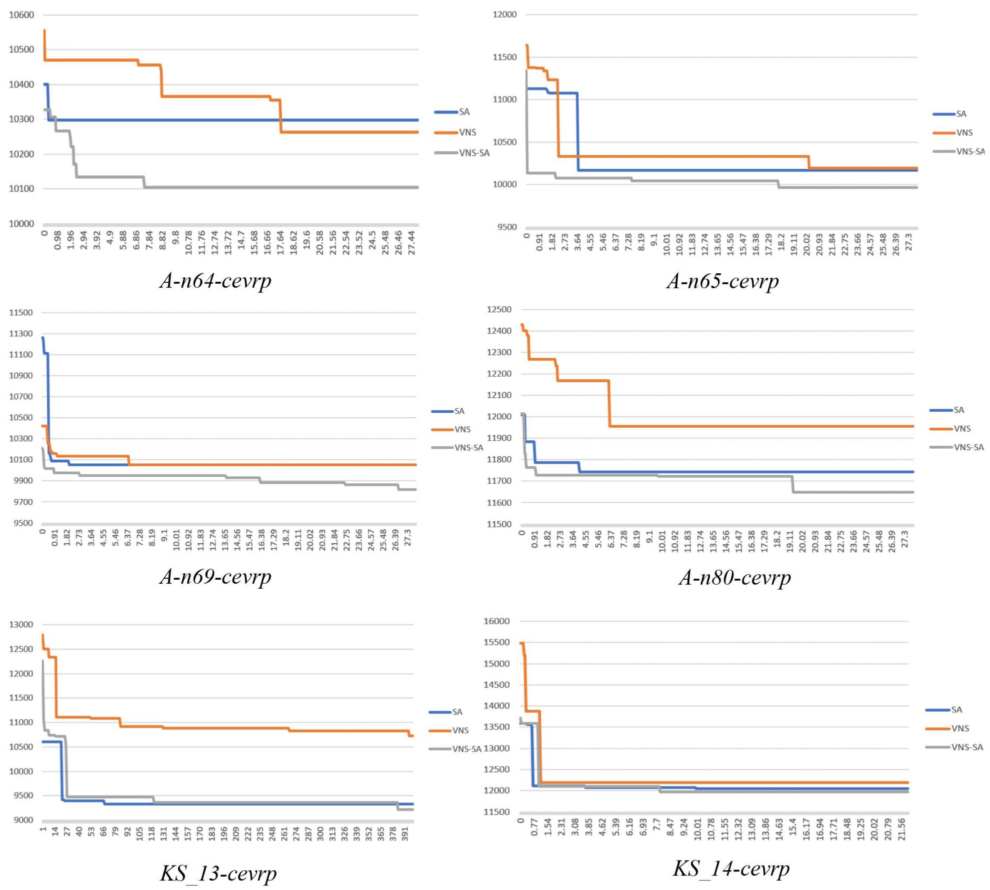


Figure 5.2 The convergence history of the various versions of the two-phase heuristic for several CE-VRP instance

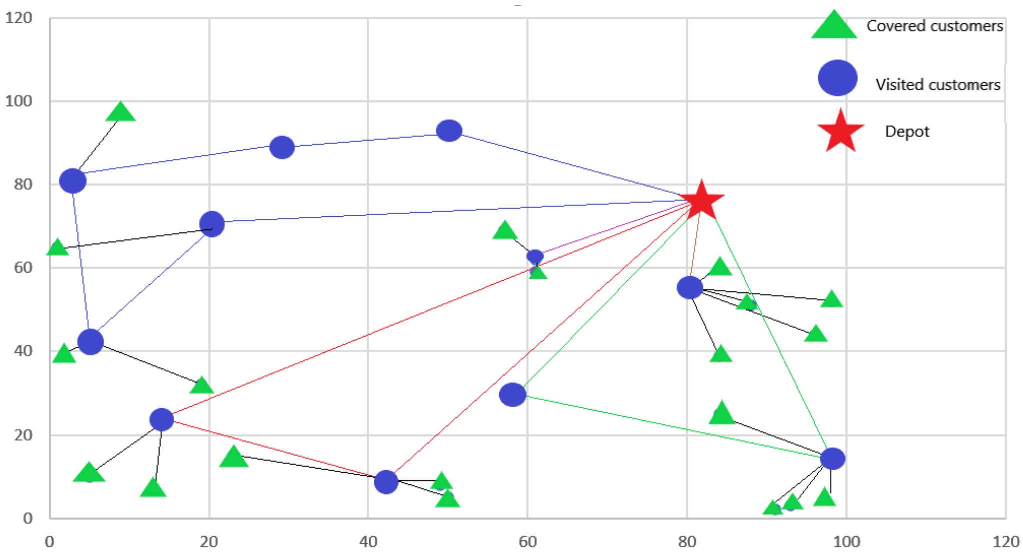


Figure 5.3 The solution found by the proposed greedy construction heuristic for the instance of *A-n32-100.cevrp*

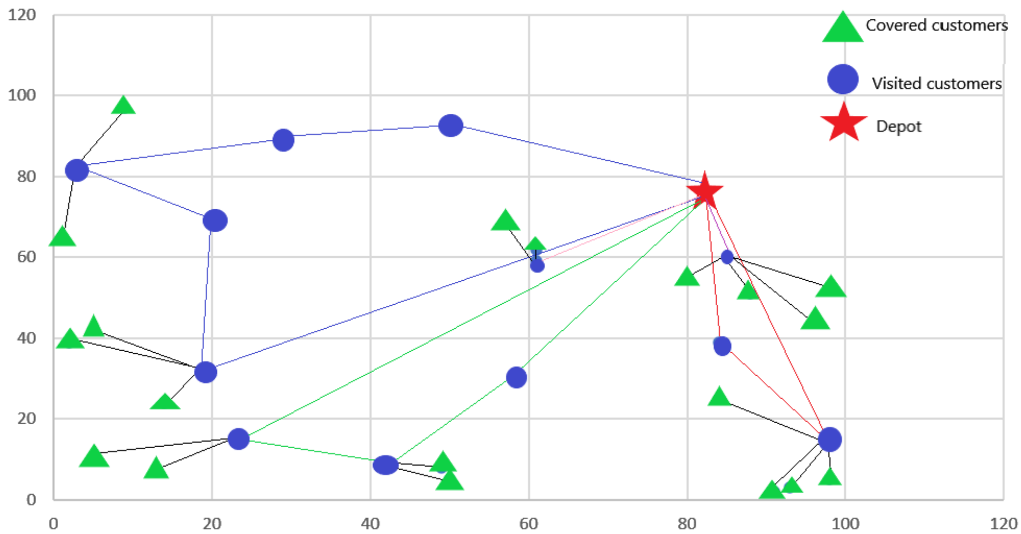


Figure 5.4 The solution found by the proposed two-phase heuristic for the instance of *A-n32-100.cevrp*

5.4 Sensitivity analysis

In this section, several analyses on the behavior of the objective function of CE-VRP are conducted. In all experiments of this section, the Gurobi solver is used to obtain the results with a time limit of 60 seconds where the first model of the CE-VRP is solved.

5.4.1 Analysis of the maximum walking distance (d_{max})

First of all, one of the important parameters in the CE-VRP is the maximum walking distance (d_{max}) for customers to reach the nearest delivery sites. d_{max} must be greater than the minimum distance between the customers; otherwise, there is no covering in the problem and CE-VRP reduces to TCVRP. Also, choosing a value for d_{max} greater than the maximum distance between the customers reduces the CE-VRP to the 1-Median Problem and no routing decision. It is important for decision-makers or managers of the companies providing distribution or delivery services to consider a proper upper bound for the walking distance (walking cost) for each customer. A larger maximum walking distance, although, reduces the number of delivery sites leading to the lower route cost of the vehicles, it will enforce the customers to travel more distance to reach the delivery site leading to raising the lack of satisfaction.

Moreover, although considering a lower d_{max} increases the customers' accessibility (decreases the assignment cost) for the company, it increases the routing costs and usage cost of the vehicles. Therefore, the trade-off analysis is needed to observe the behaviour of the objective function with respect to the different values of d_{max}. To do the trade-off analysis, the sum of the total traveled distance by the customers to reach the delivery sites is included in the objective function i.e., the objective function (3.1), where c_{ij} is equal to d_{ij} for the sake of simplicity. Fig. 5.5 shows the behaviour of the objective function with respect to the different values of d_{max} over ten CE-VRP instances. Also, Fig. 5.6 presents the behaviour of the normalized value[3] of objective function for instances of *A-n32-100*-*A-n33-100*, and *A-n64*-*A-n80* to clarify the observation.

[3] The process of converting a distribution into a Normal distribution

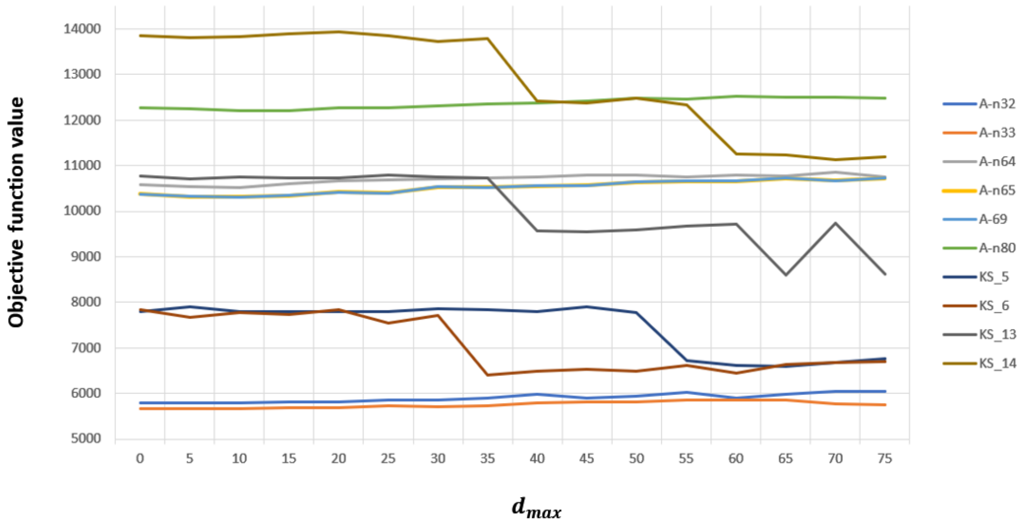


Figure 5.5 The behaviour of the objective function value in respect to the different values of d_{max} over ten CE-VRP instances

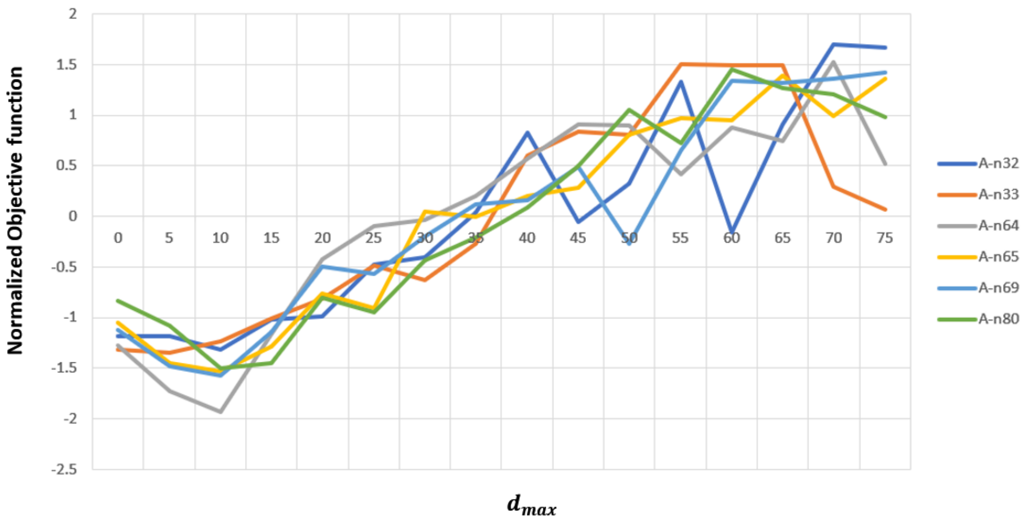


Figure 5.6 The behaviour of the normalized objective function value in respect to the different values of d_{max} over six CE-VRP instances

According to Figs. 5.5-5.6, if traveling costs for vehicles and customers have the same priority for the managers, then a large value of d_{max} is not a profitable decision for the company. For example, in Fig. 5.6, the proper value of d_{max}, by which the objective function has the minimum value, lies in the range of 5-15, which is comparatively a small value. Also, Fig. 5.7 shows the behaviour of the vehicles' route cost (vertical axis) and assignment cost of the customers (horizontal axis) in respect to the different values of d_{max} (data labels) over the instance of *A-n80.cevrp*. The red point on Fig. 5.7 indicates the minimum total cost where the distribution of the solutions is similar to a Pareto frontier (red curve) and also this distribution shows that the trade-off decision between the route and assignment costs is affected by the value of d_{max}.

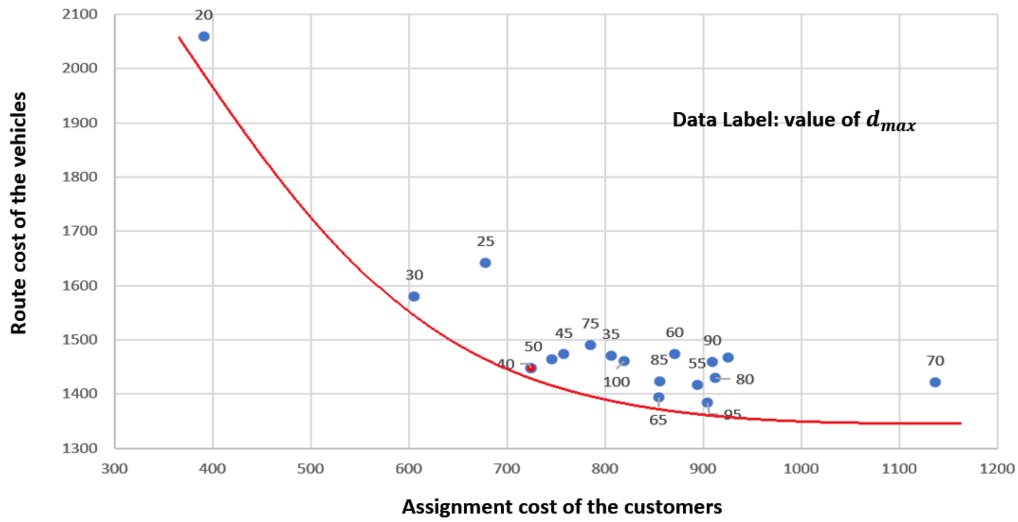


Figure 5.7 The trade-off analysis between the route and assignment costs with respect to various values of d_{max} over the instance of *A-n80.cevrp*

5.4.2 Analysis of the maximum number of available delivery sites (P)

One of the factors that has an impact on choosing a value of d_{max} is the maximum number of delivery sites. This case could happen in the situation that company has no sufficient number of vehicles to serve the customers or several stopping times for a vehicle is not applicable because of battery and charging considerations; so, the managers should know how the lower maximum number of delivery sites can affect the behaviour of the objective function.

To do the analysis on the maximum number of delivery sites, a new constraint of (5.1) is added to the CE-VRP model as follows:

$$\sum_{j \in V \setminus \{0\}} z_{jj} \leq P \tag{5.1}$$

where P is the maximum number of delivery sites. Table 5.11 shows the behaviour of the objective function values with respect to the different maximum number of delivery sites over five CE-VRP instances in which the infeasible solutions are left blank and optimal value of P is made bold. Table 5.11 shows that considering a lower value for P could make the problem infeasible; in this case, an alternative to avoid the infeasibility is to increase the maximum walking (coverage) distance. Also, the larger value of P does not have an impact on the objective function since it is not profitable for the company to use more delivery sites increasing the route costs while the maximum walking distance is unchanged.

Table 5.11 The behaviour of the objective function values in respect to the different values of the maximum number of delivery sites over CE-VRP instances

Instance	**P (the maximum number of delivery sites)**								
A-n32-100.cevrp	17	16	15	14	13	**12**	11	10	9
	5616.05	5616.05	5616.05	5616.05	5616.05	5616.05	5633.94	-	-
A-n33-100.cevrp	14	13	12	11	**10**	9	8	7	6
	5486.59	5486.59	5486.59	5486.59	5486.59	6538.66	-	-	-
A-n64.cevrp	22	21	20	19	**18**	17	16	15	14
	10070.10	10070.10	10070.10	10070.10	10070.10	10113.40	10113.40	10113.40	-
KS_13	34	33	32	31	30	**29**	28	27	26
	9148.01	9148.01	9148.01	9148.01	9148.01	9148.01	9191.44	-	-
KS_14	32	31	30	29	**28**	27	26	25	24
	11974.60	11974.60	11974.60	11974.60	11974.60	13470.20	-	-	-

5.5 Discussion on results

In the previous section, the computational results of solving the various variants of the covering-routing problem by the proposed two-phase method and Gurobi solver (an exact general solver) with sensitivity analysis on the results were presented. After parameter tuning of the proposed method, the performance of the proposed greedy heuristic to find the initial solutions was evaluated and validated. Next, the *repair* and *improvement* operators used to find the solutions with better quality were evaluated and their efficiency was shown over the various-sized instances compared to the two-phase method without them. Moreover, the solutions found by the proposed two-phase method for the CVRP (with its well-known benchmarks), and TCVRP showed that it outperforms the existing solvers proposed for VRP instances although the focus of the present thesis is not to compete with those solvers. Furthermore, several covering-routing problems including CoVRP, TCCVRP, and CE-VRP were solved by the proposed method in a short time on average and a comparison of the results with the Gurobi solver showed the efficiency and validity of the proposed two-phase heuristic; the best-found solutions have zero gaps with the Gurobi results. Also, the two-phase method with VNS-SA heuristic as the second phase outperformed the two-phase method with either solo VNS or solo SA in all CE-VRP instances, which shows the efficiency of the combination of VNS and SA in solving the routing part. Moreover, the convergence history of the proposed method shows the capability of the method in finding high-quality solutions in a low running time; at initial seconds, a solution with good quality is found and then improved during the execution of the algorithm.

Finally, the impact of the important covering parameters such as d_{max} (maximum walking distance for customers) and P (maximum available delivery sites) on the behaviour of the objective function value is analyzed. Sensitivity analysis shows that d_{max} has a significant role in finding a trade-off between the routing cost of the vehicles and customers' covering costs. The larger values for d_{max} decrease the route cost of the vehicle and increase the cost of the customers' walking; on the other hand, smaller values of d_{max} increase the route costs and decrease the assignment cost. So that, by increasing the d_{max}, the objective function is decreased which is due to decreasing the route costs of the vehicles.

6. CONCLUSION AND SUGGESTIONS FOR FUTURE STUDIES

This thesis studies a covering-routing problem that arises in the real-world situation of routing the *autonomous delivery electric vehicles* while satisfying the technical constraints of the last mile delivery problem. So, a new problem in the context of the covering-routing problem is introduced as Covering-routing of the autonomous electric vehicles, or CE-VRP, which extends both covering-routing and VRP problems. In CE-VRP, there is no obligation to visit all customers unlike most VRP's, but all customers must be visited directly or covered by (assigned to) the customers on the routes if they are within a maximum walking (coverage) distance to those customers. The customers en route are called delivery site or stopping customer location. Hence CE-VRP finds the optimal routes for the vehicles visiting a subset of the customers (delivery sites) and optimal assignment of the remaining customers to the delivery sites according to the maximum walking distance while satisfying the load capacity of the vehicles, maximum route distance, and maximum route duration. The maximum route distance (length) must be taken into account since there is no recharging stations for the (electric) vehicles during their trip; also, the maximum route duration (time) is equivalent to considering a time window for the vehicles to exit from the depot and return to it.

After defining the problem and indicating its position in the literature on the covering-routing and VRP's, two MIP models are developed for CE-VRP; one with the assignment cost in its objective function, and the other with the maximum walking distance as its constraint. Two MIP models are developed according to efficient mathematical models proposed for handling the constraints of the maximum route distance/duration in the literature with the polynomial number of constraints and decision variables. Due to the NP-hardness of the studied problem, CE-VRP, a new two-phase heuristic with a greedy initialization algorithm consisting of selecting the delivery sites and assigning the customers to the delivery sites (first phase), and routing the vehicles visiting the delivery sites (second phase) is proposed to solve the large-sized instances. The first phase is composed of various efficient *repair* and *improvement* operators which repair the infeasible solutions and improve the quality of the feasible solutions. For the second phase, a hybrid VNS-SA metaheuristic is proposed which finds the high-quality routes by diversifying (with neighbourhood change and shaking mechanisms of VNS) and intensifying (with intra-route and adjacent-routes move operators defined within SA) the solution space.

The computational results show the efficiency of the greedy initialization algorithm, the *repair* and *improvement* operators used in the first phase, and SA as the local search within the hybrid VNS-SA in the second phase. Also, the proposed two-phase heuristic is efficient in solving the other problems of the literature such as CVRP, TCVRP, CoVRP, and TCCVRP, where it outperforms the existing solvers and Gurobi solver and reaches the best-known solution in the most instances. Next, the proposed two-phase method is validated by solving the small-sized CE-VRP instances to the optimality. Finally, the sensitivity analysis indicates that the maximum walking distance has a significant impact on finding a trade-off between the routing cost of the vehicles and customers' covering (assignment) cost.

For future studies, some suggestions are presented as follows: extending the problem by considering the assumption that a subset of the customers can be left isolated, which creates a new problem of time & distance-constrained maximal covering-routing problem; the stochastic or fuzzy parameters can be added to the problem to make it close to real-world situations; solving the problem related to a real case-study to verify the proposed method and compare the solutions given by both heuristic and the methods (policies) the companies apply; developing the proposed method by considering the population-based VNS-SA to search the solution space extensively.

BIBLIOGRAPHY

Aarts, E., Korst, J., & Michiels, W. (2005). Simulated annealing. In *Search methodologies* (pp. 187–210). Springer.

Aarts, E. & Van Laarhoven, P. (1989). Simulated annealing: an introduction. *Statistica Neerlandica*, *43*(1), 31–52.

Abe, F. H. N., Hoshino, E. A., & Hill, A. (2015). The ring tree facility location problem. *Electronic Notes in Discrete Mathematics*, *50*, 331–336.

Afsar, H. M., Prins, C., & Santos, A. C. (2014). Exact and heuristic algorithms for solving the generalized vehicle routing problem with flexible fleet size. *International Transactions in Operational Research*, *21*(1), 153–175.

Ai, T. J. & Kachitvichyanukul, V. (2009). A particle swarm optimization for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, *36*(5), 1693–1702.

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

Alinaghian, M. & Goli, A. (2015). Two new heuristic algorithms for covering tour problem. *Journal of Industrial and Systems Engineering*, *8*(3), 24–41.

Alipour, M. M., Emami, H., & Abdolhosseinzadeh, M. (2022). A mas approach for vehicle routing problem. *Neural Computing and Applications*, (34), 4387–4411.

Allahyari, S., Salari, M., & Vigo, D. (2015). A hybrid metaheuristic algorithm for the multi-depot covering tour vehicle routing problem. *European Journal of Operational Research*, *242*(3), 756–768.

Almoustafa, S. (2013). *Distance-constrained vehicle routing problem: exact and approximate solution (mathematical programming).* PhD thesis, Brunel University, School of Information Systems, Computing and Mathematics.

Álvarez-Miranda, E. & Sinnl, M. (2020). A branch-and-cut algorithm for the maximum covering cycle problem. *Annals of Operations Research*, *284*(2), 487–499.

Amiri, A. & Salari, M. (2019). Time-constrained maximal covering routing problem. *Or Spectrum*, *41*(2), 415–468.

Angelelli, E. & Speranza, M. G. (2002). The periodic vehicle routing problem with intermediate facilities. *European journal of Operational research*, *137*(2), 233–247.

Archetti, C., Carrabs, F., & Cerulli, R. (2018). The set orienteering problem. *European Journal of Operational Research*, *267*(1), 264–272.

Archetti, C., Feillet, D., Hertz, A., & Speranza, M. G. (2010). The undirected capacitated arc routing problem with profits. *Computers & Operations Research*, *37*(11), 1860–1869.

Archetti, C., Speranza, M. G., & Vigo, D. (2014). Chapter 10: Vehicle routing problems with profits. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition* (pp. 273–297). SIAM.

Arkin, E. & Hassin, R. (1991). Approximate algorithms for the geometric covering salesman problem. Technical report, Cornell University Operations Research and Industrial Engineering.

Arkin, E. M. & Hassin, R. (1994). Approximation algorithms for the geometric covering salesman problem. *Discrete Applied Mathematics*, *55*(3), 197–218.

Augerat, P., Naddef, D., Belenguer, J., Benavent, E., Corberan, A., & Rinaldi, G. (1995). Computational results with a branch and cut code for the capacitated vehicle routing problem. (INPG-RR-949-M).

Baker, B. M. & Ayechew, M. (2003). A genetic algorithm for the vehicle routing problem. *Computers & Operations Research*, *30*(5), 787–800.

Balas, E. (1989). The prize collecting traveling salesman problem. *Networks*, *19*(6), 621–636.

Balas, E. (2007). The prize collecting traveling salesman problem and its applications. In *The traveling salesman problem and its variations* (pp. 663–695). Springer.

Baldacci, R., Bartolini, E., & Laporte, G. (2010). Some applications of the generalized vehicle routing problem. *Journal of the operational research society*, *61*(7), 1072–1077.

Baldacci, R., Boschetti, M. A., Maniezzo, V., & Zamboni, M. (2005). Scatter search methods for the covering tour problem. In *Metaheuristic optimization via memory and evolution* (pp. 59–91). Springer.

Baldacci, R., Dell'Amico, M., & González, J. S. (2007). The capacitated m-ring-star problem. *Operations research*, *55*(6), 1147–1162.

Baldacci, R., Hill, A., Hoshino, E. A., & Lim, A. (2017). Pricing strategies for capacitated ring-star problems based on dynamic programming algorithms. *European Journal of Operational Research*, *262*(3), 879–893.

Barma, P. S., Dutta, J., Mukherjee, A., & Kar, S. (2021). A multi-objective ring star vehicle routing problem for perishable items. *Journal of Ambient Intelligence and Humanized Computing*, *13*, 2355–2380.

Baya, G., Mauttone, A., & Robledo, F. (2017). The capacitated m two node survivable star problem. *Yugoslav Journal of Operations Research*, *27*(3), 341–366.

Bayá, G., Mauttone, A., Robledo, F., & Romero, P. (2016a). Capacitated m two-node survivable star problem. *Electronic Notes in Discrete Mathematics*, *52*, 253–260.

Bayá, G., Mauttone, A., Robledo, F., & Romero, P. (2016b). The capacitated m two-node survivable star problem: A hybrid metaheuristic approach. In *International Workshop on Hybrid Metaheuristics*, (pp. 171–186). Springer.

Bayá, G., Mauttone, A., Robledo, F., Romero, P., & Rubino, G. (2016). Capacitated m ring star problem under diameter constrained reliability. *Electronic Notes in Discrete Mathematics*, *51*, 23–30.

Bayá Mantani, G., Mauttone Vidales, A. D., & Robledo Amoza, F. R. (2015). The capacitated m two node survivable star problem. *Reportes Técnicos 15-11*, (11).

Beasley, J. & Nascimento, E. (1996). The vehicle routing-allocation problem: A unifying framework. *Top*, *4*(1), 65–86.

Bektaş, T., Erdoğan, G., & Røpke, S. (2011). Formulations and branch-and-cut algorithms for the generalized vehicle routing problem. *Transportation Science*, *45*(3), 299–316.

Berger, J. & Barkaoui, M. (2003). A new hybrid genetic algorithm for the capacitated vehicle routing problem. *Journal of the operational Research Society*, *54*(12), 1254–1262.

Berinsky, H. & Zabala, P. (2011). An integer linear programming formulation and branch-and-cut algorithm for the capacitated m-ring-star problem. *Electronic*

Notes in Discrete Mathematics, *37*, 273–278.

Bertsimas, D. & Tsitsiklis, J. (1993). Simulated annealing. *Statistical science*, *8*(1), 10–15.

Biesinger, B., Hu, B., & Raidl, G. (2016). An integer l-shaped method for the generalized vehicle routing problem with stochastic demands. *Electronic Notes in Discrete Mathematics*, *52*, 245–252.

Biesinger, B., Hu, B., & Raidl, G. R. (2015). A variable neighborhood search for the generalized vehicle routing problem with stochastic demands. In *European Conference on Evolutionary Computation in Combinatorial Optimization*, (pp. 48–60). Springer.

Biesinger, B., Hu, B., & Raidl, G. R. (2018). A genetic algorithm in combination with a solution archive for solving the generalized vehicle routing problem with stochastic demands. *Transportation Science*, *52*(3), 673–690.

Binney, J. & Sukhatme, G. S. (2012). Branch and bound for informative path planning. In *2012 IEEE international conference on robotics and automation*, (pp. 2147–2154). IEEE.

Bolduc, M.-C., Renaud, J., Boctor, F., & Laporte, G. (2008). A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society*, *59*(6), 776–787.

Bouzid, M. C., Haddadene, H. A., & Salhi, S. (2017). An integration of lagrangian split and vns: The case of the capacitated vehicle routing problem. *Computers & Operations Research*, *78*, 513–525.

Briskorn, D. (2019). On approximating maximum covering cycles in undirected graphs. *Optimization Letters*, *13*(2), 445–448.

Bullnheimer, B., Hartl, R. F., & Strauss, C. (1999). Applying the ant system to the vehicle routing problem. In *Meta-heuristics* (pp. 285–296). Springer.

Buluc, E., Peker, M., Kara, B. Y., & Dora, M. (2021). Covering vehicle routing problem: application for mobile child friendly spaces for refugees. *OR Spectrum*, (44), 461–484.

Butt, S. E. & Cavalier, T. M. (1994). A heuristic for the multiple tour maximum collection problem. *Computers & Operations Research*, *21*(1), 101–111.

Calvete, H. I., Galé, C., & Iranzo, J. A. (2013a). An efficient evolutionary algorithm for the ring star problem. *European Journal of Operational Research*, *231*(1), 22–33.

Calvete, H. I., Galé, C., & Iranzo, J. A. (2013b). An evolutionary algorithm for the biobjective capacitated m-ring star problem. In *International Conference on Algorithmic Decision Theory*, (pp. 116–129). Springer.

Calvete, H. I., Galé, C., & Iranzo, J. A. (2016). Meals: A multiobjective evolutionary algorithm with local search for solving the bi-objective ring star problem. *European Journal of Operational Research*, *250*(2), 377–388.

Ceselli, A., Righini, G., & Tresoldi, E. (2014). Combined location and routing problems for drug distribution. *Discrete applied mathematics*, *165*, 130–145.

Chen, P., Huang, H.-k., & Dong, X.-Y. (2010). Iterated variable neighborhood descent algorithm for the capacitated vehicle routing problem. *Expert Systems with Applications*, *37*(2), 1620–1627.

Chen, X., Hu, X., Tang, Z., Wang, C., & Zhang, Y. (2017). Algorithms for the ring star problem. In *International conference on combinatorial optimization and applications*, (pp. 3–16). Springer.

Christofides, N., Mingozzi, A., & Toth, P. (1979). Loading problems. *N. Christofides and al., editors, Combinatorial Optimization*, (), 339–369.

Clarke, G. & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, *12*(4), 568–581.

Clímaco, G., Rosseti, I., Silva, R., & Guerine, M. (2021). Reactive grasp for the prize-collecting covering tour problem. *RAIRO: Recherche Opérationnelle*, *55*, 1441.

Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, *52*(8), 928–936.

Créput, J.-C. & Koukam, A. (2008). The memetic self-organizing map approach to the vehicle routing problem. *Soft Computing*, *12*(11), 1125–1141.

Current, J. R. & Schilling, D. A. (1989). The covering salesman problem. *Transportation science*, *23*(3), 208–213.

Current, J. R. & Schilling, D. A. (1994). The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, *73*(1), 114–126.

Current, J. R. & Schilling, D. A. (1996). The median tour and maximal covering tour problems: formulations and heuristics. *Location Science*, *4*(4), 275–276.

Dam, T.-L., Li, K., & Fournier-Viger, P. (2017). Chemical reaction optimization with unified tabu search for the vehicle routing problem. *Soft Computing*, *21*(21), 6421–6433.

Dantzig, G. B. & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, *6*(1), 80–91.

Dastmardi, M., Mohammadi, M., & Naderi, B. (2020). Maximal covering salesman problems with average travelling cost constrains. *International Journal of Mathematics in Operational Research*, *17*(2), 153–169.

Deitch, R. & Ladany, S. P. (2000). The one-period bus touring problem: Solved by an effective heuristic for the orienteering tour problem and improvement algorithm. *European Journal of Operational Research*, *127*(1), 69–77.

Dell'Amico, M., Maffioli, F., & Värbrand, P. (1995). On prize-collecting tours and the asymmetric travelling salesman problem. *International Transactions in Operational Research*, *2*(3), 297–308.

Dumez, D., Tilk, C., Irnich, S., Lehuédé, F., & Péton, O. (2021). Hybridizing large neighborhood search and exact methods for generalized vehicle routing problems with time windows (revision 2). *EURO Journal on Transportation and Logistics*, *10*, 100040.

Erdelić, T. & Carić, T. (2019). A survey on the electric vehicle routing problem: variants and solution approaches. *Journal of Advanced Transportation*, *2019*.

Erdoğan, S. & Miller-Hooks, E. (2012). A green vehicle routing problem. *Transportation research part E: logistics and transportation review*, *48*(1), 100–114.

Farham-Nia, S. & Ghaffari-Hadigheh, A. (2021). Optimal pricing strategy in uncertain dual distribution channel with retail services. *Journal of Modelling in Management*, *16*.

Feillet, D., Dejax, P., & Gendreau, M. (2005). Traveling salesman problems with profits. *Transportation science*, *39*(2), 188–205.

Flood, M. M. (1956). The traveling-salesman problem. *Operations research*, *4*(1), 61–75.

Flores-Garza, D. A., Salazar-Aguilar, M. A., Ngueveu, S. U., & Laporte, G. (2017). The multi-vehicle cumulative covering tour problem. *Annals of Operations Research*, *258*(2), 761–780.

Fouilhoux, P. & Questel, A. (2012). The non-disjoint m-ring-star problem: polyhedral results and sdh/sonet network design. In *International Symposium on Combinatorial Optimization*, (pp. 93–104). Springer.

Fouilhoux, P. & Questel, A. (2014a). Branch-and-cut-and-price using stable set polytope inequalities for the m-capacitated-ring-star problem. In *International Symposium on Combinatorial Optimization (ISCO 2014)*.

Fouilhoux, P. & Questel, A. (2014b). A branch-and-cut for the non-disjoint m-ring-star problem. *RAIRO-Operations Research*, *48*(2), 167–188.

Franco, C., López-Santana, E., & Mendez-Giraldo, G. (2016). A variable neighborhood search approach for the capacitated m-ring-star problem. In *International Conference on Intelligent Computing*, (pp. 3–11). Springer.

Gao, P., Kaas, H.-W., Mohr, D., & Wee, D. (2016). Automotive revolution–perspective towards 2030: How the convergence of disruptive technology-driven trends could transform the auto industry. *Advanced Industries, McKinsey & Company*, (1).

Gavalas, D., Konstantopoulos, C., Mastakas, K., & Pantziou, G. (2014). A survey on algorithmic approaches for solving tourist trip design problems. *Journal of Heuristics*, *20*(3), 291–328.

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G., & Vathis, N. (2016). Efficient metaheuristics for the mixed team orienteering problem with time windows. *Algorithms*, *9*(1), 6.

Gendreau, M., Hertz, A., & Laporte, G. (1994). A tabu search heuristic for the vehicle routing problem. *Management science*, *40*(10), 1276–1290.

Gendreau, M., Laporte, G., & Semet, F. (1997). The covering tour problem. *Operations Research*, *45*(4), 568–576.

Gendreau, M., Laporte, G., & Semet, F. (1998). A tabu search heuristic for the undirected selective travelling salesman problem. *European Journal of Operational Research*, *106*(2-3), 539–545.

Ghaziri, H. (1996). Supervision in the self-organizing feature map: Application to the vehicle routing problem. In *Meta-heuristics* (pp. 651–660). Springer.

Ghiani, G. & Improta, G. (2000). An efficient transformation of the generalized vehicle routing problem. *European Journal of Operational Research*, *122*(1), 11–17.

Ghoniem, A., Scherrer, C. R., & Solak, S. (2013). A specialized column generation approach for a vehicle routing problem with demand allocation. *Journal of the Operational Research Society*, *64*(1), 114–124.

Glize, E., Roberti, R., Jozefowiez, N., & Ngueveu, S. U. (2020). Exact methods for mono-objective and bi-objective multi-vehicle covering tour problems. *European Journal of Operational Research*, *283*(3), 812–824.

Glock, K. & Meyer, A. (2022). Spatial coverage in routing and path planning problems. *European Journal of Operational Research*, ().

Goel, R. & Maini, R. (2018). A hybrid of ant colony and firefly algorithms (hafa) for solving vehicle routing problems. *Journal of Computational Science*, *25*, 28–37.

Gokalp, O. & Ugur, A. (2020). A multi-start ils–rvnd algorithm with adaptive solution acceptance for the cvrp. *Soft Computing*, *24*(4), 2941–2953.

Goldani, S. (2020). *The bi-objective maximal-covering minimal-tour problem.* PhD thesis, North Carolina State University.

Golden, B., Naji-Azimi, Z., Raghavan, S., Salari, M., & Toth, P. (2012). The generalized covering salesman problem. *INFORMS Journal on Computing*, *24*(4), 534–553.

Golden, B. L., Levy, L., & Vohra, R. (1987). The orienteering problem. *Naval Research Logistics (NRL)*, *34*(3), 307–318.

Golden, B. L., Magnanti, T. L., & Nguyen, H. Q. (1977). Implementing vehicle routing algorithms. *Networks*, *7*(2), 113–148.

Groër, C., Golden, B., & Wasil, E. (2011). A parallel algorithm for the vehicle routing problem. *INFORMS Journal on Computing*, *23*(2), 315–330.

Grosso, A., Salassa, F., & Vancroonenburg, W. (2016). Searching for a cycle with maximum coverage in undirected graphs. *Optimization Letters*, *10*(7), 1493–1504.

Ha, M. H., Bostel, N., Langevin, A., & Rousseau, L.-M. (2013). An exact algorithm and a metaheuristic for the multi-vehicle covering tour problem with a constraint on the number of vertices. *European Journal of Operational Research*, *226*(2), 211–220.

Hà, M. H., Bostel, N., Langevin, A., & Rousseau, L.-M. (2014). An exact algorithm and a metaheuristic for the generalized vehicle routing problem with flexible fleet size. *Computers & Operations Research*, *43*, 9–19.

Ha, M. H., Bostel, N., Langevin, A., & Rousseau, L.-M. (2014). Solving the close-enough arc routing problem. *Networks*, *63*(1), 107–118.

Hachicha, M., Hodgson, M. J., Laporte, G., & Semet, F. (2000). Heuristics for the multi-vehicle covering tour problem. *Computers & Operations Research*, *27*(1), 29–42.

Handoko, S. D., Chuin, L. H., Gupta, A., Soon, O. Y., Kim, H. C., & Siew, T. P. (2015). Solving multi-vehicle profitable tour problem via knowledge adoption in evolutionary bi-level programming. In *2015 IEEE Congress on Evolutionary Computation (CEC)*, (pp. 2713–2720). IEEE.

Hansen, P., Mladenović, N., Brimberg, J., & Pérez, J. A. M. (2019). Variable neighborhood search. In *Handbook of metaheuristics* (pp. 57–97). Springer.

Hansen, P., Mladenović, N., & Perez-Britos, D. (2001). Variable neighborhood decomposition search. *Journal of Heuristics*, *7*(4), 335–350.

Hill, A. & Voß, S. (2016). Optimal capacitated ring trees. *EURO Journal on Computational Optimization*, *4*(2), 137–166.

Ho, S. C. & Gendreau, M. (2006). Path relinking for the vehicle routing problem. *Journal of heuristics*, *12*(1), 55–72.

Hoshino, E. A. & de Souza, C. C. (2009). A branch-and-cut-and-price approach for the capacitated m-ring-star problem. *Electronic Notes in Discrete Mathematics*, *35*, 103–108.

Hoshino, E. A. & De Souza, C. C. (2012). A branch-and-cut-and-price approach for the capacitated m-ring–star problem. *Discrete Applied Mathematics*, *160*(18), 2728–2741.

Hoshino, E. A. & Souza, C. C. d. (2008). Column generation algorithms for the capacitated m-ring-star problem. In *International Computing and Combinatorics Conference*, (pp. 631–641). Springer.

Imase, M. & Waxman, B. M. (1991). Dynamic steiner tree problem. *SIAM Journal on Discrete Mathematics*, *4*(3), 369–384.

James, J. & Lam, A. Y. (2017). Autonomous vehicle logistic system: Joint routing and charging strategy. *IEEE Transactions on Intelligent Transportation Systems*, *19*(7), 2175–2187.

Jepsen, M. K., Petersen, B., Spoorendonk, S., & Pisinger, D. (2014). A branch-and-cut algorithm for the capacitated profitable tour problem. *Discrete Optimization*, *14*, 78–96.

Jiang, L., Zang, X., Dong, J., & Liang, C. (2022). A covering traveling salesman problem with profit in the last mile delivery. *Optimization Letters*, *16*(1), 375–393.

Jozefowiez, N. (2014). A branch-and-price algorithm for the multivehicle covering tour problem. *Networks*, *64*(3), 160–168.

Jozefowiez, N., Semet, F., & Talbi, E.-G. (2007). The bi-objective covering tour problem. *Computers & operations research*, *34*(7), 1929–1942.

Kammoun, M., Derbel, H., & Jarboui, B. (2018). A general variable neighborhood search with mixed vnd for the multi-vehicle multi-covering tour problem. In *International Conference on Variable Neighborhood Search*, (pp. 259–273). Springer.

Kammoun, M., Derbel, H., & Jarboui, B. (2020). Two meta-heuristics for solving the multi-vehicle multi-covering tour problem with a constraint on the number of vertices. *Yugoslav Journal of Operations Research*, *31*(3), 299–318.

Kammoun, M., Derbel, H., Ratli, M., & Jarboui, B. (2015). A variable neighborhood search for solving the multi-vehicle covering tour problem. *Electronic Notes in Discrete Mathematics*, *47*, 285–292.

Kammoun, M., Derbel, H., Ratli, M., & Jarboui, B. (2017). An integration of mixed vnd and vns: the case of the multivehicle covering tour problem. *International Transactions in Operational Research*, *24*(3), 663–679.

Kara, B. & Savaser, S. (2018). Periodic location routing problem: An application of mobile health services in rural areas. *1*.

Kara, I. (2011). Arc based integer programming formulations for the distance constrained vehicle routing problem. In *3rd IEEE International Symposium on Logistics and Industrial informatics*, (pp. 33–38). IEEE.

Karaoğlan, İ., Erdoğan, G., & Koç, Ç. (2018). The multi-vehicle probabilistic covering tour problem. *European Journal of Operational Research*, *271*(1), 278–287.

Katz, G. J. & Kider, J. T. (2008). All-pairs shortest-paths for large graphs on the gpu. *1*.

Kedad-Sidhoum, S. & Nguyen, V. H. (2010). An exact algorithm for solving the ring star problem. *Optimization*, *59*(1), 125–140.

Kheirkhahzadeh, M. & Barforoush, A. A. (2009). A hybrid algorithm for the vehicle routing problem. In *2009 ieee congress on evolutionary computation*, (pp. 1791–1798). IEEE.

Kır, S., Yazgan, H. R., & Tüncel, E. (2017). A novel heuristic algorithm for capacitated vehicle routing problem. *Journal of Industrial Engineering International*, *13*(3), 323–330.

Kirkpatrick, S., Gelatt Jr, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *science*, *220*(4598), 671–680.

Kucukoglu, I., Dewil, R., & Cattrysse, D. (2021). The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, *161*, 107650.

Kulkarni, R. & Bhave, P. R. (1985). Integer programming formulations of vehicle routing problems. *European journal of operational research*, *20*(1), 58–67.

Labbé, M., Laporte, G., Martín, I. R., & González, J. J. S. (1999). The median cycle problem. (1).

Labbé, M., Laporte, G., Martín, I. R., & González, J. J. S. (2004). The ring star problem: Polyhedral analysis and exact algorithm. *Networks: An International Journal*, *43*(3), 177–189.

Labbé, M., Laporte, G., Martın, I. R., & González, J. J. S. (2005). Locating median cycles in networks. *European Journal of Operational Research*, *160*(2), 457–470.

Laporte, G. (1987). Location-routing problems. (1).

Laporte, G. & Martello, S. (1990). The selective travelling salesman problem. *Discrete applied mathematics*, *26*(2-3), 193–207.

Laporte, G. & Martín, I. R. (2007). Locating a cycle in a transportation or a telecommunications network. *Networks: An International Journal*, *50*(1), 92–108.

Laporte, G., Nobert, Y., & Desrochers, M. (1985). Optimal routing under capacity and distance restrictions. *Operations research*, *33*(5), 1050–1073.

Lee, C.-Y., Lee, Z.-J., Lin, S.-W., & Ying, K.-C. (2010). An enhanced ant colony optimization (eaco) applied to capacitated vehicle routing problem. *Applied Intelligence*, *32*(1), 88–95.

Lee, Y., Chiu, S., & Sanchez, J. (1998). A branch and cut algorithm for the steiner ring star problem. *Management Science and Financial Engineering*, *4*(1), 21–34.

Leticia Vargas, B., Jozefowiez, N., & Ngueveu, S. U. (2015). A selector operator-based adaptive large neighborhood search for the covering tour problem. In *Learning and Intelligent Optimization: 9th International Conference, LION 9, Lille, France, January 12-15, 2015. Revised Selected Papers*, volume 8994, (pp. 170). Springer.

Li, C.-L., Simchi-Levi, D., & Desrochers, M. (1992). On the distance constrained vehicle routing problem. *Operations research*, *40*(4), 790–799.

Li, K., Zhang, T., Wang, R., Wang, Y., Han, Y., & Wang, L. (2021). Deep reinforcement learning for combinatorial optimization: Covering salesman problems. *IEEE Transactions on Cybernetics*, (34437087).

Liefooghe, A., Jourdan, L., & Talbi, E.-G. (2010). Metaheuristics and cooperative approaches for the bi-objective ring star problem. *Computers & Operations Research*, *37*(6), 1033–1044.

Lin, C., Choy, K. L., Ho, G. T., Chung, S. H., & Lam, H. (2014). Survey of green vehicle routing problem: past and future trends. *Expert systems with applications*, *41*(4), 1118–1138.

Lin, J., Zhou, W., & Wolfson, O. (2016). Electric vehicle routing problem. *Transportation research procedia*, *12*, 508–521.

Lin, S. & Kernighan, B. W. (1973). An effective heuristic algorithm for the traveling-salesman problem. *Operations research*, *21*(2), 498–516.

Lin, S.-W., Lee, Z.-J., Ying, K.-C., & Lee, C.-Y. (2009). Applying hybrid metaheuristics for capacitated vehicle routing problem. *Expert Systems with Ap-*

plications, *36*(2), 1505–1512.

Lopes, R., Souza, V. A., & da Cunha, A. S. (2013). A branch-and-price algorithm for the multi-vehicle covering tour problem. *Electronic Notes in Discrete Mathematics*, *44*, 61–66.

Lu, Y., Benlic, U., & Wu, Q. (2021). A highly effective hybrid evolutionary algorithm for the covering salesman problem. *Information Sciences*, *564*, 144–162.

Macedo, R., Alves, C., Hanafi, S., Jarboui, B., Mladenović, N., Ramos, B., & De Carvalho, J. V. (2015). Skewed general variable neighborhood search for the location routing scheduling problem. *Computers & Operations Research*, *61*, 143–152.

Maervoet, J., Brackman, P., Verbeeck, K., Causmaecker, P. D., & Berghe, G. V. (2013). Tour suggestion for outdoor activities. In *International Symposium on Web and Wireless Geographical Information Systems*, (pp. 54–63). Springer.

Margolis, J. T., Song, Y., & Mason, S. J. (2022). A multi-vehicle covering tour problem with speed optimization. *Networks*, *79*(2), 119–142.

Marinakis, Y., Marinaki, M., & Dounias, G. (2010). Honey bees mating optimization algorithm for large scale vehicle routing problems. *Natural Computing*, *9*(1), 5–27.

Martín, I. R. (2000). *Cycle location problems*. PhD thesis, Universidad de La Laguna (Canary Islands, Spain).

Matsuura, T. & Kimura, T. (2017). Covering salesman problem with nodes and segments. *American Journal of Operations Research*, *7*(04), 249.

Mattila, M. (2016). The distance-constrained generalized vehicle routing problem. *language*, (1), 35.

Mattila, M. (2018). A cyclic exchange neighborhood search algorithm and transformations for the generalized vehicle routing problem. (1).

Mauttone, A., Nesmachnow, S., Olivera, A., & Amoza, F. R. (2008). Solving a ring star problem generalization. In *2008 International Conference on Computational Intelligence for Modelling Control & Automation*, (pp. 981–986). IEEE.

Maziero, L. P., Usberti, F. L., & Cavellucci, C. (2021). Branch-and-cut algorithms for the covering salesman problem. *arXiv preprint arXiv:2104.01173*, (1).

Medak, J. & Gogoi, P. P. (2018). Review and analysis of single-source shortest path problem using dijkstra's algorithm. *IOSR Journal of Computer Engineering (IOSR-JCE)*, *20*(2), 10–15.

Mennell, W. K. (2009). *Heuristics for solving three routing problems: Close-enough traveling salesman problem, close-enough vehicle routing problem, and sequence-dependent team orienteering problem*. University of Maryland, College Park.

Mester, D. & Bräysy, O. (2007). Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & operations research*, *34*(10), 2964–2975.

Mitra, S. (2005). An algorithm for the generalized vehicle routing problem with backhauling. *Asia-Pacific Journal of Operational Research*, *22*(02), 153–169.

Mladenović, N., Dražić, M., Kovačevic-Vujčić, V., & Čangalović, M. (2008). General variable neighborhood search for the continuous optimization. *European Journal of Operational Research*, *191*(3), 753–770.

Mladenović, N. & Hansen, P. (1997). Variable neighborhood search. *Computers & operations research*, *24*(11), 1097–1100.

Moccia, L., Cordeau, J.-F., & Laporte, G. (2012). An incremental tabu search heuristic for the generalized vehicle routing problem with time windows. *Journal of the Operational Research Society*, *63*(2), 232–244.

Modares, A., Somhom, S., & Enkawa, T. (1999). A self-organizing neural network approach for multiple traveling salesman and vehicle routing problems. *International Transactions in Operational Research*, *6*(6), 591–606.

Moghdani, R., Salimifard, K., Demir, E., & Benyettou, A. (2021). The green vehicle routing problem: A systematic literature review. *Journal of Cleaner Production*, *279*, 123691.

Morandi, N., Leus, R., & Yaman, H. (2021). Exact algorithms for budgeted prize-collecting covering subgraph problems. *arXiv preprint arXiv:2101.03546*, (1).

Motta, L., Nogueira, L., & Ochi, L. (2010). Improving performance of algorithms for the covering tour problem by applying reduction rules. In *Proceedings of the 25th Mini EURO Conference on Uncertainty and Robustness in Planning and Decision Making–URPDM 2010*.

Motta, L., Ochi, L. S., & Martinhon, C. (2001a). Grasp metaheuristics for the generalized covering tour problem. In *MIC2001-4th Metaheuristics Int Conf. Porto, Portugal*. Citeseer.

Motta, L. C., Ochi, L. S., & Martinhon, C. A. (2001b). Reduction rules for the covering tour problem. *Electronic Notes in Discrete Mathematics*, *7*, 142–145.

Murakami, K. (2014). A column generation approach for the multi-vehicle covering tour problem. In *2014 IEEE international conference on automation science and engineering (CASE)*, (pp. 1063–1068). IEEE.

Murakami, K. (2018a). A generalized model and a heuristic algorithm for the large-scale covering tour problem. *RAIRO–Operations Research*, *52*(2).

Murakami, K. (2018b). Iterative column generation algorithm for generalized multi-vehicle covering tour problem. *Asia-Pacific Journal of Operational Research*, *35*(04), 1850021.

Nagata, Y. & Bräysy, O. (2009). Edge assembly-based memetic algorithm for the capacitated vehicle routing problem. *Networks: An International Journal*, *54*(4), 205–215.

Naji-Azimi, Z., Renaud, J., Ruiz, A., & Salari, M. (2012). A covering tour approach to the location of satellite distribution centers to supply humanitarian aid. *European Journal of Operational Research*, *222*(3), 596–605.

Naji-Azimi, Z. & Salari, M. (2014). The time constrained maximal covering salesman problem. *Applied Mathematical Modelling*, *38*(15-16), 3945–3957.

Naji Azimi, Z., Salari, M., Golden, B., Raghavan, S., & Toth, P. (2009). Generalized covering salesman problem. In *XL Annual Conference of Italian Operational Research Society*.

Naji Azimi, Z., Salari, M., & Toth, P. (2009). A heuristic method to solve the capacitated m-ring-star problem. In *XL Annual Conference of Italian Operational Research Society*.

Naji-Azimi, Z., Salari, M., & Toth, P. (2010). A heuristic procedure for the capacitated m-ring-star problem. *European Journal of Operational Research*, *207*(3), 1227–1234.

Naji-Azimi, Z., Salari, M., & Toth, P. (2012). An integer linear programming based heuristic for the capacitated m-ring-star problem. *European Journal of Operational Research*, *217*(1), 17–25.

Navidadham, M., Arbabsadeghi, M., Bayat, A. A., & Didehvar, F. (2015). Solving generalized vehicle routing problem by parallel universes and tabu search. In *2015 6th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, (pp. 1–7). IEEE.

Nazif, H. & Lee, L. S. (2012). Optimised crossover genetic algorithm for capacitated vehicle routing problem. *Applied Mathematical Modelling*, *36*(5), 2110–2117.

Nolz, P. C., Doerner, K. F., Gutjahr, W. J., & Hartl, R. F. (2010). A bi-objective metaheuristic for disaster relief operation planning. In *Advances in multi-objective nature inspired computing* (pp. 167–187). Springer.

Obreque, C., Paredes-Belmar, G., Miranda, P. A., Campuzano, G., & Gutiérrez-Jarpa, G. (2020). The generalized median tour problem: modeling, solving and an application. *IEEE Access*, *8*, 178097–178107.

Oliveira, W. A. d., Moretti, A. C., & Reis, E. F. (2015). Multi-vehicle covering tour problem: building routes for urban patrolling. *Pesquisa Operacional*, *35*(3), 617–644.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of operations research*, *41*(4), 421–451.

Ozbaygin, G., Yaman, H., & Karasan, O. E. (2016). Time constrained maximal covering salesman problem with weighted demands and partial coverage. *Computers & Operations Research*, *76*, 226–237.

Pandiri, V. & Singh, A. (2019). An artificial bee colony algorithm with variable degree of perturbation for the generalized covering traveling salesman problem. *Applied Soft Computing*, *78*, 481–495.

Pandiri, V., Singh, A., & Rossi, A. (2020). Two hybrid metaheuristic approaches for the covering salesman problem. *Neural Computing and Applications*, *32*(19), 15643–15663.

Pekel, E. & Kara, S. S. (2019). Solving fuzzy capacitated location routing problem using hybrid variable neighborhood search and evolutionary local search. *Applied Soft Computing*, *83*, 105665.

Pérez, J. A. M., Moreno-Vega, J. M., & Martın, I. R. (2003). Variable neighborhood tabu search and its application to the median cycle problem. *European Journal of Operational Research*, *151*(2), 365–378.

Pham, T. A., Hà, M. H., & Nguyen, X. H. (2017). Solving the multi-vehicle multi-covering tour problem. *Computers & operations research*, *88*, 258–278.

Pham, T. A., Hà, M. H., & Nguyen, X. H. (2018). Description of the datasets for the experiments in the paper "solving the multi-vehicle multi-covering tour problem". *Data in brief*, *18*, 1146–1148.

Pop, P., Matei, O., Burzu, E., & Gyorodi, C. (2011). A computational analysis of heuristic algorithms on generalized vehicle routing problem. *Journal of Computer Science and Control Systems*, *4*(1), 143.

Pop, P., Matei, O., & Valean, H. (2011). An efficient hybrid soft computing approach to the generalized vehicle routing problem. In *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*, (pp. 281–289). Springer.

Pop, P. C., Fuksz, L., & Marc, A. H. (2014). A variable neighborhood search approach for solving the generalized vehicle routing problem. In *International Conference on Hybrid Artificial Intelligence Systems*, (pp. 13–24). Springer.

Pop, P. C., Kara, I., & Marc, A. H. (2012). New mathematical models of the generalized vehicle routing problem and extensions. *Applied Mathematical Modelling*, *36*(1), 97–107.

Pop, P. C., Matei, O., & Sitar, C. P. (2013). An improved hybrid algorithm for solving the generalized vehicle routing problem. *Neurocomputing*, *109*, 76–83.

Pop, P. C., Pintea, C., Zelina, I., & Dumitrescu, D. (2009). Solving the generalized vehicle routing problem with an acs-based algorithm. In *AIP Conference Proceedings*, volume 1117, (pp. 157–162). American Institute of Physics.

Pop, P. C., Pintea, C.-M., & Dumitrescu, D. (2009). An ant colony algorithm for solving the dynamic generalized vehicle routing problem. *" Ovidius" University Annals Constantza. Series Civil Engineering*, *1*(11), 373.

Pop, P. C., Zelina, I., Lupşe, V., Sitar, C. P., & Chira, C. (2011). Heuristic algorithms for solving the generalized vehicle routing problem. *International Journal of Computers Communications & Control*, *6*(1), 158–165.

Prins, C. (2009). A grasp× evolutionary local search hybrid for the vehicle routing problem. In *Bio-inspired algorithms for the vehicle routing problem* (pp. 35–53). Springer.

Punriboon, C., So-In, C., Aimtongkham, P., & Rujirakul, K. (2019). A bio-inspired capacitated vehicle-routing problem scheme using artificial bee colony with crossover optimizations. *J. Internet Serv. Inf. Secur.*, *9*(3), 21–40.

Püskül, O. M., Aslan, D., Onay, C., Erdogan, M. S., & Taşgetiren, M. F. (2022). Designing a railway network in cesme, izmir with bi-objective ring star problem. In *Digitizing Production Systems* (pp. 665–674). Springer.

Quttineh, N.-H., Larsson, T., Lundberg, K., & Holmberg, K. (2013). Military aircraft mission planning: a generalized vehicle routing model with synchronization and precedence. *EURO Journal on Transportation and Logistics*, *2*(1-2), 109–127.

Rego, C. (1998). A subpath ejection method for the vehicle routing problem. *Management science*, *44*(10), 1447–1459.

Rego, C. (2001). Node-ejection chains for the vehicle routing problem: Sequential and parallel algorithms. *Parallel Computing*, *27*(3), 201–222.

Rego, C. & Roucairol, C. (1996). A parallel tabu search algorithm using ejection chains for the vehicle routing problem. In *Meta-Heuristics* (pp. 661–675). Springer.

Reihaneh, M. & Ghoniem, A. (2017a). A branch-cut-and-price algorithm for the generalized vehicle routing problem. *Journal of the operational research society*, (1), 1–13.

Reihaneh, M. & Ghoniem, A. (2017b). A multi-start optimization-based heuristic for a food bank distribution problem. *Journal of the operational research society*, (1), 1–17.

Reihaneh, M. & Ghoniem, A. (2018). A multi-start optimization-based heuristic for a food bank distribution problem. *Journal of the operational research society*, *69*(5), 691–706.

Reihaneh, M. & Ghoniem, A. (2019). A branch-and-price algorithm for a vehicle routing with demand allocation problem. *European Journal of Operational Research*, *272*(2), 523–538.

Reimann, M., Doerner, K., & Hartl, R. F. (2004). D-ants: Savings based ants divide and conquer the vehicle routing problem. *Computers & Operations Research*, *31*(4), 563–591.

Renaud, J., Boctor, F. F., & Laporte, G. (2003). *Fast and Efficient Heuristics to Solve Two Version of the Median Cycle Problem.* Faculté des sciences de l'administration de l'Université Laval, Direction de ....

Renaud, J., Boctor, F. F., & Laporte, G. (2004). Efficient heuristics for median cycle problems. *Journal of the Operational Research Society*, *55*(2), 179–186.

Rochat, Y. & Taillard, É. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, *1*(1), 147–167.

Sabo, C., Pop, P. C., & Horvat-Marc, A. (2020). On the selective vehicle routing problem. *Mathematics*, *8*(5), 771.

Salari, M. & Naji-Azimi, Z. (2012). An integer programming-based local search for the covering salesman problem. *Computers & Operations Research*, *39*(11), 2594–2602.

Salari, M., Naji-Azimi, Z., & Toth, P. (2010). A variable neighborhood search and its application to a ring star problem generalization. *Electronic Notes in Discrete Mathematics*, *36*, 343–350.

Salari, M., Naji Azimi, Z., & Toth, P. (2011). An integer linear programming based heuristic approach for the capacitated m-ring-star problem. In *4rd international conference of Iranian Operations Research Society.*

Sbai, I., Krichen, S., & Limam, O. (2020). Two meta-heuristics for solving the capacitated vehicle routing problem: the case of the tunisian post office. *Operational Research*, *22*, 1–43.

Schneider, M., Stenger, A., & Goeke, D. (2014). The electric vehicle-routing problem with time windows and recharging stations. *Transportation science*, *48*(4), 500–520.

Schneider, M., Stenger, A., & Hof, J. (2015). An adaptive vns algorithm for vehicle routing problems with intermediate stops. *Or Spectrum*, *37*(2), 353–387.

Sevaux, M. & Sörensen, K. (2008). Hamiltonian paths in large clustered routing problems. In *Proceedings of the EU/MEeting 2008 workshop on Metaheuristics for Logistics and Vehicle Routing, EU/ME*, volume 8, (pp. 411–417). Citeseer.

Shaelaiea, M. H., Naji-Azimib, Z., & Salaria, M. (2014). A variable neighborhood search algorithm for the generalized covering salesman problem. *TSP*, *23*(24), 25.

Sharafi, A. & Bashiri, M. (2016). Green vehicle routing problem with safety and social concerns. *Journal of Optimization in Industrial Engineering*, *10*(21), 93–100.

Shimizu, Y. & Sakaguchi, T. (2013). Generalized vehicle routing problem for reverse logistics aiming at low carbon transportation. *Industrial Engineering and Management Systems*, *12*(2), 161–170.

Singh, P., Kamthane, A. R., & Tanksale, A. N. (2021). Metaheuristics for the distance constrained generalized covering traveling salesman problem. *OPSEARCH*, *58*(3), 575–609.

Sinnl, M. (2021). Mixed-integer programming approaches for the time-constrained maximal covering routing problem. *OR Spectrum*, *43*(2), 497–542.

Snyder, L. V. & Daskin, M. S. (2006). A random-key genetic algorithm for the generalized traveling salesman problem. *European journal of operational research*, *174*(1), 38–53.

Solak, S., Scherrer, C., & Ghoniem, A. (2014). The stop-and-drop problem in nonprofit food distribution networks. *Annals of Operations Research*, *221*(1), 407–426.

Stanojević, M., Stanojević, B., & Vujošević, M. (2013). Enhanced savings calculation and its applications for solving capacitated vehicle routing problem. *Applied Mathematics and Computation*, *219*(20), 10302–10312.

Štefaníková, P., Váňa, P., & Faigl, J. (2020). Greedy randomized adaptive search procedure for close enough orienteering problem. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, (pp. 808–814).

Subramanian, A., Uchoa, E., & Ochi, L. S. (2013). A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, *40*(10), 2519–2531.

Szeto, W. Y., Wu, Y., & Ho, S. C. (2011). An artificial bee colony algorithm for the capacitated vehicle routing problem. *European Journal of Operational Research*, *215*(1), 126–135.

Taillard, É. (1993). Parallel iterative search methods for vehicle routing problems. *Networks*, *23*(8), 661–673.

Taillard, É., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation science*, *31*(2), 170–186.

Tang, L. & Wang, X. (2006). Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem. *The International Journal of Advanced Manufacturing Technology*, *29*(11), 1246–1258.

Tasgetiren, M. F., Bulut, O., Pan, Q.-K., & Suganthan, P. N. (2011). A differential evolution algorithm for the median cycle problem. In *2011 IEEE Symposium on Differential Evolution (SDE)*, (pp. 1–7). IEEE.

Teymourian, E., Kayvanfar, V., Komaki, G. M., & Zandieh, M. (2016). Enhanced intelligent water drops and cuckoo search algorithms for solving the capacitated vehicle routing problem. *Information Sciences*, *334*, 354–378.

Toregas, C., Swain, R., ReVelle, C., & Bergman, L. (1971). The location of emergency service facilities. *Operations research*, *19*(6), 1363–1373.

Toth, P., Naji Azimi, Z., & Salari, M. (2011). An ilp local search algorithm for the capacitated m-ring-star problem. In *ROUTE 2011*.

Toth, P. & Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. *Informs Journal on computing*, *15*(4), 333–346.

Toth, P. & Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*. SIAM.

Tricoire, F., Graf, A., & Gutjahr, W. J. (2012). The bi-objective stochastic covering tour problem. *Computers & operations research*, *39*(7), 1582–1592.

Tripathy, S. P., Biswas, A., & Pal, T. (2021). A multi-objective covering salesman problem with 2-coverage. *Applied Soft Computing*, *113*, 108024.

Tripathy, S. P., Tulshyan, A., Kar, S., & Pal, T. (2017). A metameric genetic algorithm with new operator for covering salesman problem with full coverage. *Int J Control Theory Appl*, *10*(7), 245–252.

Van den Bergh, J., Quttineh, N.-H., Larsson, T., & Beliën, J. (2016). A time-indexed generalized vehicle routing model for military aircraft mission planning. In *Operations Research Proceedings 2014* (pp. 605–611). Springer.

Van Laarhoven, P. J. & Aarts, E. H. (1987). Simulated annealing. In *Simulated annealing: Theory and applications* (pp. 7–15). Springer.

Vargas, L., Jozefowiez, N., & Ngueveu, S. U. (2015). A selector operator-based adaptive large neighborhood search for the covering tour problem. In *International Conference on Learning and Intelligent Optimization*, (pp. 170–185). Springer.

Vargas, L., Jozefowiez, N., & Ngueveu, S. U. (2017). A dynamic programming operator for tour location problems applied to the covering tour problem. *Journal of Heuristics*, *23*(1), 53–80.

Venkatesh, P., Srivastava, G., & Singh, A. (2019). A multi-start iterated local search algorithm with variable degree of perturbation for the covering salesman problem. In *Harmony Search and Nature Inspired Optimization Algorithms* (pp. 279–292). Springer.

Vidal, T., Crainic, T. G., Gendreau, M., Lahrichi, N., & Rei, W. (2012). A hybrid genetic algorithm for multidepot and periodic vehicle routing problems. *Operations Research*, *60*(3), 611–624.

Vidal, T., Maculan, N., Ochi, L. S., & Vaz Penna, P. H. (2016). Large neighborhoods with implicit customer selection for vehicle routing problems with profits. *Transportation Science*, *50*(2), 720–734.

Vincent, F. Y., Redi, A. P., Yang, C.-L., Ruskartina, E., & Santosa, B. (2017). Symbiotic organisms search and two solution representations for solving the capacitated vehicle routing problem. *Applied Soft Computing*, *52*, 657–672.

West, D. B. (2001). *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River.

Xiao, Y., Kaku, I., Zhao, Q., & Zhang, R. (2011). A reduced variable neighborhood search algorithm for uncapacitated multilevel lot-sizing problems. *European Journal of Operational Research*, *214*(2), 223–231.

Xu, J. F., Chiu, S., & Glover, F. (2001). A first level scatter search implementation for solving the steiner ring problem in telecommunications network design. In *Steiner Trees in Industry* (pp. 441–466). Springer.

Yan, F. (2018). Autonomous vehicle routing problem solution based on artificial potential field with parallel ant colony optimization (aco) algorithm. *Pattern recognition letters*, *116*, 195–199.

Yang, W. & Ke, L. (2019). An improved fireworks algorithm for the capacitated vehicle routing problem. *Frontiers of Computer Science*, *13*(3), 552–564.

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.

Yuan, Y., Cattaruzza, D., Ogier, M., & Semet, F. (2020). A branch-and-cut algorithm for the generalized traveling salesman problem with time windows. *European Journal of Operational Research*, *286*(3), 849–866.

Yuan, Y., Cattaruzza, D., Ogier, M., Semet, F., & Vigo, D. (2021). A column generation based heuristic for the generalized vehicle routing problem with time windows. *Transportation Research Part E: Logistics and Transportation Review*, *152*, 102391.

Yücel, E., Salman, F. S., Bozkaya, B., & Gökalp, C. (2020). A data-driven optimization framework for routing mobile medical facilities. *Annals of Operations Research*, *291*(1), 1077–1102.

Yuh, J., Lee, Y., & Park, G. (2014). A new mixed-integer programming modeling for the steiner ring star problem. *Journal of the Korean Operations Research and Management Science Society*, *39*(1), 13–27.

Yurtkuran, A. & Emel, E. (2010). A new hybrid electromagnetism-like algorithm for capacitated vehicle routing problems. *Expert Systems with Applications*, *37*(4), 3427–3433.

Zang, X., Jiang, L., Ding, B., & Fang, X. (2021). A hybrid ant colony system algorithm for solving the ring star problem. *Applied Intelligence*, *51*(6), 3789–3800.

Zang, X., Jiang, L., Ratli, M., & Ding, B. (2020). A parallel variable neighborhood search for solving covering salesman problem. *Optimization Letters*, *16*, 175–190.

Zehetner, M. & Gutjahr, W. J. (2018). Sampling-based genetic algorithms for the bi-objective stochastic covering tour problem. In *Recent Developments in Metaheuristics* (pp. 253–284). Springer.

Zhang, H. & Xu, Y. (2018). Online covering salesman problem. *Journal of Combinatorial Optimization*, *35*(3), 941–954.

Zhang, Z., Qin, H., & Lim, A. (2014). A memetic algorithm for the capacitated m-ring-star problem. *Applied intelligence*, *40*(2), 305–321.

Ziegler, J. S. (2013). The dynamic multi-objective multi-vehicle covering tour problem. (ADA587192).

APPENDIX A

VNS and SA algorithms in the literature

Algorithm 4 The pseudo-code of the BVNS introduced in the literature

Require: k_{max}, S_0

$S \leftarrow S_0$
while `<The stopping condition is not satisfied>` **do**
    $k \leftarrow 1$
    while $k \leq k_{max}$ **do**
        $S' \leftarrow Shake(S, k)$
        $S'' \leftarrow First\_Improvement(S')$
        $Neighborhood\_Change(S, S'', k)$
    end while
end while

Algorithm 5 The pseudo-code of general SA introduced in the literature

Require: S_0

$T \leftarrow T_{max}$
$S \leftarrow S_0$
while $T \geq T_{min}$ **do**
    while `<The stopping condition is not satisfied>` **do**
        $S' \leftarrow Neighbourhood\_Generate(S)$
        if $f(S') < f(S)$ **then**
            $S \leftarrow S'$
        else
            Accept S' with the probability $e^{(f(S)-f(S')/T}$
        end if
    end while
    $T \leftarrow F(T)$ ▷ Temperature updated
end while

Initialization algorithm in the proposed two-phase heuristic

Algorithm 6 The proposed greedy initialization heuristic

Require: d_{ij} (distance between node i and j); d_{max} (maximum walking distance); q_i (Demand of node i); Q (vehicle capacity)

for $i \leftarrow 1$ to N **do**
    for $j \leftarrow 1$ to N **do**
        if $(i \neq j)$ & $(d_{ij} \leq d_{max})$ **then**
            $N\_Cov_i \leftarrow N\_Cov_i + 1$ ▷ N_Cov_i: the number of nodes that node i can cover
        end if
    end for
end for
Sort N_Cov_i in descending order by a *sorting* algorithm;
for $i \leftarrow 1$ to N **do**
    $r_i \leftarrow -1$ ▷ r_i: the element i-th of the matrix $\mathcal{R}$ (section 4.3.1)
    $x_i \leftarrow 0$ ▷ x_i: the element i-th of the matrix $\mathcal{X}$ (section 4.3.1)
end for
for $i \leftarrow 1$ to N **do**
    for $j \leftarrow 1$ to N **do**
        if <Center i is not opened (from top of sorted N_Cov_i) and the node j is within the coverage of node i> **then**
            <The node j is assigned to the node (center) i> or $x_j = i$
        end if
    end for
end for
for $i \leftarrow 1$ to N **do**
    if $x_i = 0$ **then** $x_i \leftarrow i$, $r_i \leftarrow 1$ ▷ Not covered node i becomes a delivery site
    end if
end for
for $i \leftarrow 1$ to N **do**
    if $r_i = 1$ **then**
        $t \leftarrow 0$
        for $j \leftarrow 1$ to N **do**
            if $x_j = i$ **then**
                $t \leftarrow t + q_j$
                if $t > Q$ **then** $x_j \leftarrow 0$, $t \leftarrow t - q_j$, $q'_i \leftarrow t$
                else $q'_i \leftarrow t$ ▷ q'_i: updated demand of node i after assigning the other nodes to it
                end if
            end if
        end for
    end if
end for
for $i \leftarrow 1$ to N **do**
    if $x_i = 0$ **then** $x_i \leftarrow i$, $r_i \leftarrow 1$, $q'_i \leftarrow q_i$
    end if
end for
$\mathcal{RO} \leftarrow Route\_Generate(\mathcal{R}, \mathcal{X}, q'_i)$ ▷ $Route\_Generate(.)$ function is given in Alg. 7
$\mathcal{RO} \leftarrow VNS\_SA(\mathcal{RO})$
return $\mathcal{R}, \mathcal{X}, \mathcal{RO}$

Algorithm 7 The *Route_Generate*(.) function

Require: $\mathcal{R}, \mathcal{X}, q'_i$

$k \leftarrow 0$
for $i \leftarrow 1$ to N **do**
    if $r_i = 1$ **then**
        $ro_k \leftarrow i$ ▷ ro_k: the element k-th of the matrix $\mathcal{RO}$ (section 4.3.1)
        $k \leftarrow k+1$
    end if
end for
$z \leftarrow q'_{ro_0}$
$z' \leftarrow t_{0,ro_0} + s$ ▷ t_{ij} : duration between nodes i, j, s: service time
$z'' \leftarrow d_{0,ro_0}$ ▷ d_{ij} : distance between nodes i, j
$ro'_0 \leftarrow 0$ ▷ ro'_k: the element k-th of the matrix $\mathcal{RO}'$; this matrix is the feasible version of the matrix $\mathcal{RO}$
$ro'_1 \leftarrow ro_0$
$k \leftarrow 2$
for $i \leftarrow 1$ to N **do**
    $z \leftarrow z + q'_{ro_i}$
    $z' \leftarrow z' + t_{i-1,i} + s$
    $z'' \leftarrow d_{i-1,i}$
    if $(z \leq Q)$ & $(z' + t_{ro_i,0} \leq T)$ & $(z'' + d_{ro_i,0} \leq D)$ **then**
        $ro'_k \leftarrow ro_i$
        $k \leftarrow k+1$
    else
        $ro'_k \leftarrow 0$
        $ro'_{k+1} \leftarrow ro_i$
        $k \leftarrow k+2$
        $z \leftarrow q'_{ro_i}$
        $z' \leftarrow t_{0,ro_i} + s$
        $z'' \leftarrow d_{0,ro_i}$
    end if
end for
$ro'_k \leftarrow 0$
`return` $\mathcal{RO}$, $\mathcal{RO}'$

Algorithm of *Repair_Improvement* operators

Algorithm 8 The first proposed *improvement* operator (I_1)

Require: $\mathcal{R}, \mathcal{X}$

$l \leftarrow 0$
$l' \leftarrow 0$
for $i \leftarrow 1$ to N **do**
    if $r_i = 1$ **then**
        $t \leftarrow 0$
        for $j \leftarrow 1$ to N **do**
            if $(i \neq j)$ & $(x_j = i)$ **then**
                for $k \leftarrow 1$ to N **do**
                    if $(x_k = i)$ & $(j \neq k)$ & $(d_{jk} > d_{max})$ **then**
                        $t \leftarrow t+1$
                    end if
                    if $t = 0$ **then**
                        $Save^0_{l'} \leftarrow i$ ▷ $Save^0_i$ saves the potential i-th delivery site for exchange
                        $Save^1_{l'} \leftarrow j$ ▷ $Save^1_i$ saves the potential i-th covered node for exchange
                        $l' \leftarrow l'+1$
                        $l \leftarrow l+1$
                    end if
                end for
            end if
        end for
    end if
end for
if $l > 0$ **then**
    for $i \leftarrow 0$ to l' **do**
        for $j \leftarrow 1$ to $2N$ **do**
            if $Save^0_i = ro_j$ **then**
                if $d_{ro_{j-1},Save^1_i} + d_{Save^1_i,ro_{j+1}} > d_{ro_{j-1},ro_j} + d_{ro_j,ro_{j+1}}$ **then**
                    $r_{Save^0_i} \leftarrow 0$
                    $r_{Save^1_i} \leftarrow 1$ ▷ Matrix $\mathcal{R}$ is updated
                    $x_{Save^0_i} \leftarrow Save^0_i$
                    $x_{Save^1_i} \leftarrow Save^1_i$ ▷ Matrix $\mathcal{X}$ is updated
                end if
            end if
        end for
    end for
end if
`return improved` $\mathcal{R}, \mathcal{X}$

Algorithm 9 The second proposed *improvement* operator (I_2)

Require: $\mathcal{R}, \mathcal{X}$

$l \leftarrow 0$

$l' \leftarrow 0$

for $i \leftarrow 1$ to N **do**

  if $r_i = 1$ **then**

    $t \leftarrow 0$

    for $j \leftarrow 1$ to N **do**

      if $(x_j = i)$ & $(i \neq j)$ **then**

        $t \leftarrow t + 1$

        $a \leftarrow j$

      end if

    end for

    if $t = 1$ **then**

      $Save^0_{l'} \leftarrow i$ ▷ $Save^0_i$ saves the potential i-th delivery site for exchange

      $Save^1_{l'} \leftarrow a$ ▷ $Save^1_i$ saves the potential i-th covered node for exchange

      $l' \leftarrow l' + 1$

      $l \leftarrow l + 1$

    end if

  end if

end for

if $l > 0$ **then**

  for $i \leftarrow 0$ to l' **do**

    for $j \leftarrow 1$ to $2N$ **do**

      if $Save^0_i = ro_j$ **then**

        if $d_{ro_{j-1}, Save^1_i} + d_{Save^1_i, ro_{j+1}} > d_{ro_{j-1}, ro_j} + d_{ro_j, ro_{j+1}}$ **then**

          $r_{Save^0_i} \leftarrow 0$

          $r_{Save^1_i} \leftarrow 1$ ▷ Matrix $\mathcal{R}$ is updated

          $x_{Save^0_i} \leftarrow Save^0_i$

          $x_{Save^1_i} \leftarrow Save^1_i$ ▷ Matrix $\mathcal{X}$ is updated

        end if

      end if

    end for

  end for

end if

`return improved` $\mathcal{R}, \mathcal{X}$

Algorithm 10 The third proposed *improvement* operator (I_3)

Require: $\mathcal{R}, \mathcal{X}$

for $i \leftarrow 1$ to N **do**
    if $r_i = 0$ **then**
        for $j \leftarrow 1$ to N **do**
            if $(r_j = 1)$ & $(i \neq j)$ & $(d_{ij} \leq d_{max})$ **then**
                $t \leftarrow 0$
                for $k \leftarrow 1$ to N **do**
                    $t' \leftarrow 0$
                    if $(x_k = j)$ & $(j \neq k)$ **then**
                        for $y \leftarrow 1$ to N **do**
                            if $(r_y = 1)$ & $(d_{yi} > d_{max})$ & $(d_{yk} \leq d_{max})$ **then**
                                $t' \leftarrow t' + 1$
                            end if
                        end for
                        if $d_{ik} \leq d_{max}$ **then**
                            $t' \leftarrow t' + 1$
                        end if
                    end if
                    if $t' = 0$ **then**
                        $t \leftarrow t + 1$
                    end if
                end for
                if $t = 0$ **then**
                    $r_j \leftarrow 0$
                    $l \leftarrow l + 1$
                end if
            end if
        end for
        if $l \geq 2$ **then**
            $r_i \leftarrow 1$
        end if
    end if
end for
`return improved` $\mathcal{R}, \mathcal{X}$

Algorithm 11 The fourth proposed *improvement* operator (I_4)

Require: $\mathcal{R}, \mathcal{X}$

```
for i ← 1 to N do
    if r_i = 1 then
        t ← 0
        for j ← 1 to N do
            if (i ≠ j) & (x_j = i) then
                t' ← 0
                for k ← 1 to N do
                    if (d_jk ≤ d_max) & (r_k = 1) & (i ≠ k) then
                        t' ← t' + 1
                    end if
                end for
                if t' = 0 then
                    t ← t + 1
                end if
            end if
        end for
        if t = 0 then
            r_i = 0
        end if
    end if
end for
return improved R, X
```

Algorithm 12 The fifth proposed *improvement* operator (I_5)

Require: $\mathcal{R}, \mathcal{X}$, k ▷ k is a delivery site which is selected randomly

$t \leftarrow 0$
for $i \leftarrow 1$ to N **do**
    if $(k \neq i)$ & $(x_i = k)$ **then**
        $t \leftarrow t+1$
    end if
end for
if $t = 0$ **then**
    for $j \leftarrow 1$ to N **do**
        if $(k \neq j)$ & $(d_{k,j} \leq d_{max})$ & $(r_j = 1)$ **then**
            $r_k \leftarrow 0$
            $x_k \leftarrow j$
        end if
    end for
end if
`return improved` $\mathcal{R}, \mathcal{X}$

Algorithm 13 The proposed *repair* operator (R_1)

Require: $\mathcal{R}, \mathcal{X}$

for $i \leftarrow 1$ to N **do**
    if $r_i = 1$ **then**
        $t \leftarrow 0$
        for $j \leftarrow 1$ to N **do**
            if $x_j = i$ **then**
                $t \leftarrow t + q_j$
                if $t > Q$ **then** $x_j \leftarrow 0$, $t \leftarrow t - q_j$, $q'_i \leftarrow t$
                else $q'_i \leftarrow t$
                end if
            end if
        end for
    end if
end for
for $i \leftarrow 1$ to N **do**
    if $x_i = 0$ **then** $x_i \leftarrow i$, $r_i \leftarrow 1$, $q'_i \leftarrow q_i$
    end if
end for
`return repaired` $\mathcal{R}, \mathcal{X}$

Algorithms of *TPH_VNS* and *TPH_SA* (another versions of the second phase of the two-phase method)

Algorithm 14 The proposed solo VNS algorithm as the second phase ($VNS(.)$)

Require: S, N, k_{max} ▷ k_{max} : the number of neighbourhoods

$n \leftarrow 1$
while $n \leq N$ **do**
    $k \leftarrow 1$
    while $k \leq k_{max}$ **do**
        $S' \leftarrow Shake\_VNS(S, k)$ ▷ $Shake\_VNS(.)$ moves are defined in section 4.3.4.2
        $S'' \leftarrow First\_Improvement(S')$ ▷ $First\_Improvement$ is given in Alg. 15
        $Neighborhood\_Change(S, S'', k)$
    end while
    $n \leftarrow n + 1$
end while

Algorithm 15 $First\_Improvement$ algorithm in the second phase (VNS)

Require: S, P ▷ P: the maximum number of opportunities that is given to $First\_Improvement$ to accept a solution

$r \leftarrow 0$
$p \leftarrow 0$
while $(r < 1)$ & $(p < P)$ **do**
    $S' \leftarrow Neighbourhood(S)$ ▷ $Neighbourhood(.)$ is the move operators explained in section 4.3.4 plus the lines 8-30 in the Algorithm 7
    if $f(S') < f(S)$ **then**
        $S \leftarrow S'$
        $r \leftarrow r + 1$
    end if
    $p \leftarrow p + 1$
end while

Algorithm 16 The proposed solo SA algorithm in the second phase ($SA(.)$)

Require: $S, M, T_{max}, T_{min}, \alpha$

$T \leftarrow T_{max}$
while $T \geq T_{min}$ **do**
  $t \leftarrow 1$
  while $t \leq M$ **do**
    $S' \leftarrow Shake(S)$ ▷ $Shake(.)$ moves are defined in section 4.3.4.2
    $S'' \leftarrow First\_Improvement(S')$ ▷ $First\_Improvement$ is given in Alg. 3
    $t \leftarrow t+1$
  end while
  $T \leftarrow \alpha * T$ ▷ Temperature is updated
end while

APPENDIX B

CE-VRP instances generated in this thesis

Table B.1 The features of the first dataset generated for CE-VRP (small-sized instances)

Original instance	Modified instance	n	Q	s	T	d_{max}	D
A-n32-k5.vrp	*A-n9-50.cevrp*	9	50	5	500	20	600
A-n32-k5.vrp	*A-n9-100.cevrp*	9	100	5	500	20	600
A-n32-k5.vrp	*A-n10-50.cevrp*	10	50	5	500	20	600
A-n32-k5.vrp	*A-n10-100.cevrp*	10	100	5	500	20	600
A-n32-k5.vrp	*A-n15-100.cevrp*	15	100	5	500	20	650
A-n32-k5.vrp	*A-n20-100.cevrp*	20	100	5	500	20	700
A-n32-k5.vrp	*A-n32-100.cevrp*	32	100	5	500	20	700
A-n33-k5.vrp	*A-n33-100.cevrp*	33	100	5	500	20	700
B-n31-k5.vrp	*B-n8-100.cevrp*	8	100	10	700	40	800
B-n31-k5.vrp	*B-n9-100.cevrp*	9	100	10	700	40	800
B-n31-k5.vrp	*B-n10-100.cevrp*	10	100	10	700	40	800
B-n31-k5.vrp	*B-n12-100.cevrp*	12	100	10	700	40	800
B-n31-k5.vrp	*B-n8-90.cevrp*	8	90	10	700	40	800
B-n31-k5.vrp	*B-n9-90.cevrp*	9	90	10	700	40	800
B-n31-k5.vrp	*B-n10-90.cevrp*	10	90	10	700	40	800
B-n31-k5.vrp	*B-n12-90.cevrp*	12	90	10	700	40	800
P-n16-k8.vrp	*P-n10-35.cevrp*	10	35	15	900	30	1000
P-n16-k8.vrp	*P-n12-35.cevrp*	12	35	15	900	30	1000
P-n19-k2.vrp	*P-n10-160.cevrp*	10	160	15	900	30	1000
P-n19-k2.vrp	*P-n9-160.cevrp*	9	160	15	900	30	1000

Table B.2 The features of the second dataset generated for CE-VRP (medium and large-sized instances)

Original instance	Modified instance	n	Q	s	T	d_{max}	D
A-n34-k5.vrp	*A-n34.cevrp*	34	100	5	500	50	600

Continued on next page

Table B.2 Continued from previous Table

Original instance	**Modified instance**	n	Q	s	T	d_{max}	D
A-n36-k5.vrp	*A-n36.cevrp*	36	100	5	500	50	600
A-n37-k5.vrp	*A-n37.cevrp*	37	100	5	500	50	600
A-n38-k5.vrp	*A-n38.cevrp*	38	100	5	500	50	600
A-n39-k5.vrp	*A-n39.cevrp*	39	100	5	500	50	600
A-n44-k6.vrp	*A-n44.cevrp*	44	100	5	500	50	600
A-n45-k6.vrp	*A-n45.cevrp*	45	100	5	500	50	600
A-n46-k7.vrp	*A-n46.cevrp*	46	100	5	500	50	600
A-n48-k7.vrp	*A-n48.cevrp*	48	100	5	500	50	600
A-n53-k7.vrp	*A-n53.cevrp*	53	100	10	700	50	800
A-n54-k7.vrp	*A-n54.cevrp*	54	100	10	700	50	800
A-n55-k9.vrp	*A-n55.cevrp*	55	100	10	700	50	800
A-n60-k9.vrp	*A-n60.cevrp*	60	100	10	700	50	800
A-n61-k9.vrp	*A-n61.cevrp*	61	100	10	700	50	800
A-n62-k8.vrp	*A-n62.cevrp*	62	100	10	700	50	800
A-n63-k9.vrp	*A-n63.cevrp*	63	100	10	700	50	800
A-n64-k9.vrp	*A-n64.cevrp*	64	100	10	700	50	800
A-n65-k9.vrp	*A-n65.cevrp*	65	100	10	700	50	800
A-n69-k9.vrp	*A-n69.cevrp*	69	100	10	700	50	800
A-n80-k10.vrp	*A-n80.cevrp*	80	100	10	700	50	800

Table B.3 The features of the third dataset generated for CE-VRP (based on real data)

Name	n	m	Q	D	T	d_{max}	s
KS_1	45	2	25	1200	1000	90	10
KS_2	45	2	25	1300	1100	90	10
KS_3	45	2	25	1500	1200	90	10
KS_4	45	2	25	1700	1300	90	10
KS_5	45	2	25	1500	1200	50	10
KS_6	45	2	25	1700	1300	50	10
KS_7	45	3	25	1000	800	90	10
KS_8	45	3	25	1050	850	90	10
KS_9	45	3	20	1100	900	90	10
KS_10	45	3	20	1200	900	90	10
KS_11	45	3	20	1200	900	50	10
KS_12	45	3	20	1300	1000	50	10

Continued on next page

Table B.3 Continued from previous Table

Name	n	m	Q	D	T	d_{max}	s
KS_13	45	4	20	1300	1000	50	10
KS_14	45	4	20	1050	800	50	10

APPENDIX C

Computational results of solving CVRP and TCVRP instances by the proposed two-phase heuristic

Table C.1 Computational results of solving the instances of set A by the proposed two-phase heuristic (n: number of customers, m: number of vehicles, Δ%: the gap between the best-known solution and the best-found solution by two-phase heuristic)

Instance features				**Two-phase heuristic**			
Name	**n**	**m**	**Best-known solution**	***Best***	***Average***	**Δ%**	**Time (s)**
A-n32-k5.vrp	32	5	784	784	784	0.00	1
A-n33-k5.vrp	33	5	661	661	661	0.00	1
A-n33-k6.vrp	33	6	742	742	742	0.00	7
A-n34-k5.vrp	34	5	778	778	778	0.00	2
A-n36-k5.vrp	36	5	799	799	799	0.00	3
A-n37-k5.vrp	37	5	669	669	669	0.00	6
A-n37-k6.vrp	37	6	949	949	949	0.00	13
A-n38-k5.vrp	38	5	730	730	730	0.00	2
A-n39-k5.vrp	39	5	822	822	822	0.00	31
A-n39-k6.vrp	39	6	831	831	831	0.00	14
A-n44-k6.vrp	44	6	937	937	937	0.00	15
A-n45-k6.vrp	45	6	944	948	948.8	0.42	89
A-n45-k7.vrp	45	7	1146	1146	1146	0.00	2
A-n46-k7.vrp	46	7	914	914	914	0.00	12
A-n48-k7.vrp	48	7	1073	1073	1073	0.00	6
A-n53-k7.vrp	53	7	1010	1010	1010	0.00	58
A-n54-k7.vrp	54	7	1167	1167	1167	0.00	15
A-n55-k9.vrp	55	9	1073	1073	1073	0.00	39
A-n60-k9.vrp	60	9	1354	1354	1354	0.00	13
A-n61-k9.vrp	61	9	1034	1035	1037.6	0.10	522
A-n62-k8.vrp	62	8	1288	1290	1291.8	0.16	219
A-n63-k9.vrp	63	9	1616	1629	1632.5	0.80	133
A-n63-k10.vrp	63	10	1314	1318	1322.1	0.30	302
A-n64-k9.vrp	64	9	1401	1413	1418.2	0.86	98
A-n65-k9.vrp	65	9	1174	1177	1179.9	0.26	35
A-n69-k9.vrp	69	9	1159	1165	1168.3	0.52	275
A-n80-k10.vrp	80	10	1763	1783	1786.1	1.13	311
Average				-	-	0.17	82.37

Table C.2 Computational results of solving the instances of set B by the proposed two-phase heuristic (n: number of customers, m: number of vehicles, Δ%: the gap between the best-known solution and the best-found solution by two-phase heuristic)

Instance features				**Two-phase heuristic**			
Name	**n**	**m**	**Best-known solution**	***Best***	***Average***	**Δ%**	**Time (s)**
B-n31-k5.vrp	31	5	672	672	672	0.00	9
B-n33-k5.vrp	33	5	788	788	788	0.00	1
B-n35-k5.vrp	35	5	955	955	955	0.00	2
B-n38-k6.vrp	38	6	805	805	805	0.00	3
B-n39-k5.vrp	39	5	549	549	549	0.00	17
B-n41-k6.vrp	41	6	829	829	829	0.00	68
B-n43-k6.vrp	43	6	742	742	742	0.00	4
B-n44-k7.vrp	44	7	909	909	909	0.00	2
B-n45-k5.vrp	45	5	751	751	751	0.00	5
B-n45-k6.vrp	33	5	678	678	678	0.00	43
B-n50-k7.vrp	50	7	741	741	741	0.00	1
B-n50-k8.vrp	50	8	1312	1313	1313.2	0.08	98
B-n51-k7.vrp	51	7	1032	1032	1032	0.00	2
B-n52-k7.vrp	52	5	747	747	747	0.00	29
B-n56-k7.vrp	56	7	707	709	711.2	0.28	166
B-n57-k7.vrp	57	7	1153	1153	1154.9	0.00	172
B-n57-k9.vrp	57	9	1598	1599	1560.6	0.06	148
B-n63-k10.vrp	63	10	1496	1503	1508.3	0.47	133
B-n64-k9.vrp	64	9	861	861	864.1	0.00	202
B-n66-k9.vrp	66	9	1316	1319	1321.9	0.23	39
B-n67-k10.vrp	67	10	1032	1034	1040.3	0.19	109
B-n68-k9.vrp	68	9	1272	1275	1276.9	0.24	68
B-n78-k10.vrp	78	10	1221	1234	1237.2	1.06	77
Average				-	-	0.11	60.78

Table C.3 Computational results of solving the instances of set P by the proposed two-phase heuristic (n: number of customers, m: number of vehicles, Δ%: the gap between the best-known solution and the best-found solution by two-phase heuristic)

Instance features				**Two-phase heuristic**			
Name	**n**	**m**	**Best-known solution**	***Best***	***Average***	**Δ%**	**Time (s)**
P-n16-k8.vrp	16	8	450	450	450	0.00	1
P-n19-k2.vrp	19	2	212	212	212	0.00	1
P-n20-k2.vrp	20	2	216	216	216	0.00	1
P-n21-k2.vrp	21	2	211	211	211	0.00	1
P-n22-k2.vrp	22	2	216	216	216	0.00	1
P-n22-k8.vrp	22	8	603	603	603	0.00	2
P-n23-k8.vrp	23	8	529	529	529	0.00	3
P-n40-k5.vrp	40	5	458	458	458	0.00	10
P-n45-k5.vrp	45	5	510	510	510	0.00	16

Continued on next page

Table C.3 Continued from previous Table

Instance features				Two-phase heuristic			
Name	**n**	**m**	**Best-known solution**	***Best***	***Average***	**Δ%**	**Time (s)**
P-n50-k7.vrp	50	7	554	554	555.4	0.00	42
P-n50-k8.vrp	50	8	631	631	631	0.00	8
P-n50-k10.vrp	50	10	696	697	699.8	0.14	99
P-n51-k10.vrp	51	10	741	743	745.2	0.27	87
P-n55-k7.vrp	55	7	568	570	571.9	0.35	91
P-n55-k10.vrp	55	10	694	698	700.8	0.58	55
P-n60-k10.vrp	60	10	744	750	753.4	0.81	59
P-n60-k15.vrp	60	15	968	972	974.3	0.41	73
P-n65-k10.vrp	65	10	792	798	802.8	0.76	68
P-n70-k10.vrp	70	10	827	835	837.3	0.97	106
P-n76-k4.vrp	76	4	593	593	595.1	0.00	199
P-n101-k4.vrp	101	4	681	689	692.7	1.17	202
Average				-	-	0.26	59.27

Table C.4 Computational results of solving the CMT instances (CMT 1-5, and CMT 11-12) by the proposed two-phase heuristic (n: number of customers, m: number of vehicles, Δ%: the gap between the best-known solution and the best-found solution by two-phase heuristic)

Instance features				Two-phase heuristic			
Name	**n**	**m**	**Best-known solution**	***Best***	***Average***	**Δ%**	**Time (s)**
CMT 1	50	5	524.61	524.61	524.61	0.00	18
CMT 2	75	10	835.26	835.26	837.77	0.00	76
CMT 3	100	8	826.14	831.12	832.88	0.60	91
CMT 4	150	12	1028.42	1042.90	1049.01	1.41	299
CMT 5	199	17	1291.29	1328.44	1338.05	2.88	312
CMT 11	120	7	1042.11	1042.11	1048.09	0.00	401
CMT 12	100	10	819.56	819.56	825.04	0.00	437
Average				-	-	0.70	233.43

Table C.5 Computational results of solving the CMT instances by the proposed two-phase heuristic (n: number of customers, m: number of vehicles, T: route maximum time, s: service time at each node, BKS: best-known solution, Δ%: the gap between the best-known solution and the best-found solution by two-phase heuristic)

Instance features						Two-phase heuristic			
Name	**n**	**m**	T	s	**BKS**	***Best***	***Average***	**Δ%**	**Time (s)**
CMT 1	50	5	∞	0	524.61	524.61	524.61	0.00	18
CMT 2	75	10	∞	0	835.26	835.26	837.77	0.00	76
CMT 3	100	8	∞	0	826.14	831.12	832.88	0.60	91
CMT 4	150	12	∞	0	1028.42	1042.90	1049.01	1.41	299

Continued on next page

Table C.5 Continued from previous Table

Instance features						**Two-phase heuristic**			
Name	**n**	**m**	T	s	**BKS**	***Best***	***Average***	**Δ%**	**Time (s)**
CMT 5	199	17	∞	0	1291.29	1328.44	1338.05	2.88	312
CMT 6	50	6	200	10	555.43	555.43	555.43	0.00	27
CMT 7	75	11	160	10	909.68	911.761	914.44	0.23	71
CMT 8	100	9	230	10	865.94	870.322	879.02	0.51	97
CMT 9	150	14	200	10	1162.55	1164.64	1166.21	0.18	73
CMT 10	199	18	200	10	1395.85	1443.02	1459.82	3.38	522
CMT 11	120	7	∞	0	1042.11	1042.11	1048.09	0.00	401
CMT 12	100	10	∞	0	819.56	819.56	825.04	0.00	437
CMT 13	120	11	720	50	1541.14	1557.43	1567.93	1.06	154
CMT 14	100	11	1040	90	866.37	867.352	870.11	0.11	77
Average						-	-	0.74	189.64

APPENDIX D

Performance evaluation of the proposed two-phase heuristic in solving CVRP and TCVRP instances

Table D.1 Comparison of computational results of the proposed two-phase heuristic with other solution methods over the set A instances (Augerat et al., 1995)

Reference	Heuristic	ADFB% [a]	CPU time[b]	Operating machine
Sbai, Krichen & Limam (2020)	GA-VNS	0.03	73.39	Intel Core i3
Stanojević, Stanojević & Vujošević (2013)	SC-ESA[c]	0.16	756.00	N/A
Present work	Two-phase heuristic	0.17	82.37	Intel Core i5 1.60 GHZ
Sbai et al. (2020)	GA	0.29	72.31	Intel Core i3
Kır, Yazgan & Tüncel (2017)	TS-ALNS[d]	0.59	5002.30	N/A
Akpinar (2016)	LNS–ACO	0.60	2335.66	Core i7 2.80 GHz
Akpinar (2016)	LNSa[e]	0.84	2288.95	Core i7 2.80 GHz
Stanojević et al. (2013)	R-ESA[f]	1.62	38.00	N/A
Akpinar (2016)	LNSi[g]	2.05	2277.27	Core i7 2.80 GHz
Stanojević et al. (2013)	ESA	4.37	0.14	N/A[h]

[a] Average deviation from best known results

[b] CPU time in seconds

[c] set-covering-based Extended Savings Algorithm

[d] Adaptive Neighborhood Search Algorithm

[e] LNS by using a solution acceptance criterion

[f] Randomized ESA

[g] LNS by accepting only the improving solutions

[h] Source code available on https://code.google.com/archive/p/esa-vrp/

Table D.2 Comparison of computational results of the proposed two-phase heuristic with other solution methods over the set B instances (Augerat et al., 1995)

Reference	**Heuristic**	**ADFB%**	**CPU time**	**Operating machine**
Present work	Two-phase heuristic	0.11	60.78	Intel Core i5 1.60 GHZ
Sbai et al. (2020)	GA-VNS	0.20	46.76	Intel Core i3
Akpinar (2016)	LNS–ACO	0.35	2489.59	Core i7 2.80 GHz
Akpinar (2016)	LNSa	0.63	2464.70	Core i7 2.80 GHz
Stanojević et al. (2013)	SC-ESA	0.95	490.00	N/A
Sbai et al. (2020)	GA	1.21	46.27	Intel Core i3
Akpinar (2016)	LNSi	1.64	2452.25	Core i7 2.80 GHz
Stanojević et al. (2013)	R-ESA	2.13	39.00	N/A
Stanojević et al. (2013)	ESA	3.91	0.14	N/A

Table D.3 Comparison of computational results of the proposed two-phase heuristic with other solution methods over the set P instances (Augerat et al., 1995)

Reference	**Heuristic**	**ADFB%**	**CPU time**	**Operating machine**
Sbai et al. (2020)	GA-VNS	0.08	74.81	Intel Core i3
Present work	Two-phase heuristic	0.26	59.27	Intel Core i5 1.60 GHZ
Stanojević et al. (2013)	SC-ESA	0.54	376.00	N/A
Akpinar (2016)	LNS–ACO	0.48	2270.75	Core i7 2.80 GHz
Akpinar (2016)	LNSa	0.81	2202.63	Core i7 2.80 GHz
Sbai et al. (2020)	GA	1.70	73.05	Intel Core i3
Akpinar (2016)	LNSi	1.97	2191.28	Core i7 2.80 GHz
Stanojević et al. (2013)	R-ESA	2.08	40.00	N/A
Stanojević et al. (2013)	ESA	7.33	0.15	N/A

Table D.4 Comparison of computational results of the proposed two-phase heuristic with other solution methods over the set CMT instances (CMT 1-5, and CMT 11-12) (Christofides et al., 1979)

Reference	**Heuristic**	**ADFB%**	**CPU time**	**Operating machine**
Subramanian, Uchoa & Ochi (2013)	MILS[1]	0.00	104	Intel Core i7 2.93 GHz
Vidal, Crainic, Gendreau, Lahrichi & Rei (2012)	HGA[2]	0.00	115	AMD Opteron 250 2.4 GHz
Groër, Golden & Wasil (2011)	PA[3]	0.00	N/A	Intel core i2 2.3 GHz
Nazif & Lee (2012)	Enhanced GA	0.09	257	Pentium 4 2.0 GHz
Punriboon, So-In, Aimtongkham & Rujirakul (2019)	ABC	0.13	274	Intel core i3 2.66 GHz
Dam, Li & Fournier-Viger (2017)	CRO-TS	0.21	119	Intel Core i2 2.40 GHz
Yu, Yang & Yao (2009)	ACO	0.22	146	Pentium 1000 MHz

Continued on next page

[1] Modified Iterated Local Search

[2] Hybrid GA

[3] Parallel Algorithm

Table D.4 – Continued from previous Table

Reference	Heuristic	ADFB%	CPU time	Operating machine
Bouzid, Haddadene & Salhi (2017)	LS-VNS[4]	0.22	251	Intel Core i3 2.20 GHz×4
Vincent, Redi, Yang, Ruskartina & Santosa (2017)	SOS[5]	0.24	86	Intel core i7 3.4 GHz
Toth & Vigo (2003)	GTS[6]	0.47	186	Pentium 200 MHz PC
Szeto, Wu & Ho (2011)	ABO	0.58	234	Pentium 1.73 GHz
Present work	Two-phase heuristic	0.70	233	Intel Core i5 1.60 GHZ
Rego (2001)	NEC[7]	0.72	1792	33 MHz Sun IPC
Rego & Roucairol (1996)	PTS[8]	0.96	848	HP 9000/712
Ai & Kachitvichyanukul (2009)	PSO	0.99	143	Intel P4 3.4GHz
Alipour, Emami & Abdolhosseinzadeh (2022)	MAS[9]	1.26	0.93	Intel Core i5 1.6 GHz
Créput & Koukam (2008)	MSOM-L [10]	1.38	934	AMD Athlon 2000MHz
Rego (1998)	SEM[11]	1.41	127	Intel P4 3.4 GHz
Yurtkuran & Emel (2010)	HELA[12]	1.60	131	Intel Core i2 2.00 GHz
Kheirkhahzadeh & Barforoush (2009)	ACO	2.65	32	2.0 GHz CPU
Créput & Koukam (2008)	MSOM-F [13]	3.79	11	AMD Athlon 2000MHz
Modares, Somhom & Enkawa (1999)	SONN[14]	4.42	N/A	SUN SPARC 10
Ghaziri (1996)	HDN[15]	5.22	2.34	N/A
Bullnheimer, Hartl & Strauss (1999)	Ant System	5.61	1260	Pentium 100 MHz
Clarke & Wright (1964)	CWSA	7.05	N/A[16]	N/A

Table D.5 Comparison of computational results of the proposed two-phase heuristic with other solution methods over the set CMT instances (Christofides et al., 1979)

Reference	Heuristic	ADFB%	CPU time	Operating machine
Teymourian, Kayvanfar, Komaki & Zandieh (2016)	LSHA-POHA[17]	0.00	83	Intel Core i7 3.4 GHz
Subramanian et al. (2013)	MILS	0.00	101	Intel Core i7 2.93 GHz
Vidal et al. (2012)	HGA	0.00	132	AMD Opteron 250 2.4 GHz

Continued on next page

[4] Lagrangian split-VNS

[5] Symbiotic Organisms Search

[6] Granular TS

[7] Node-ejection chains

[8] Parallel TS

[9] Multi-agent system

[10] Memetic Self-organizing Map-long

[11] Subpath Ejection Method

[12] Hybrid Electromagnetism-like Algorithm

[13] MSOM-fast

[14] Self-organizing Neural Network

[15] Hierarchical Deformable Nets

[16] The execution time depends on the machine on which the algorithm is run

[17] Local Search Hybrid Algorithm with Post-optimization Hybrid Algorithm

Table D.5 – Continued from previous Table

Reference	Heuristic	ADFB%	CPU time	Operating machine
Nagata & Bräysy (2009)	EAX-MA[18]	0.00	828	Opteron 2.4 GHz
Rochat & Taillard (1995)	PDI-LS[19]	0.00	N/A	Silicon Graphics 100 MHz
Mester & Bräysy (2007)	AGES[20]	0.03	168	Pentium IV 2.8 GHz
Marinakis, Marinaki & Dounias (2010)	HBMOA[21]	0.03	48	Pentium M750 1.86 GHz
Yang & Ke (2019)	FWA[22]	0.03	50	Pentium IV 2.4 GHz
Taillard (1993)	PISM[23]	0.05	N/A	Silicon Graphics 100 MHz
Goel & Maini (2018)	HAFA[24]	0.09	N/A	Core i3 2.3 GHz
Nazif & Lee (2012)	Enhanced GA	0.11	274	Pentium 4 2.0 GHz
Gokalp & Ugur (2020)	MA_ILS-RVND[25]	0.13	169	Intel Core i7 3.40 GHz
Punriboon et al. (2019)	ABC	0.13	291	Intel core i3 2.66 GHz
Yu et al. (2009)	ACO	0.14	172	Pentium 1000 MHz
Dam et al. (2017)	CRO-TS	0.15	119	Intel Core i2 2.40 GHz
Gokalp & Ugur (2020)	M_ILS-RVND[26]	0.18	169	Intel Core i7 3.40 GHz
Vincent et al. (2017)	SOS	0.24	78	Intel core i7 3.4 GHz
Prins (2009)	GRASP-ILS	0.24	311	Pentium 1 GHz
Lin, Lee, Ying & Lee (2009)	SA-TS	0.35	493	Pentium IV 2.8 GHz
Gokalp & Ugur (2020)	A_ILS-RVND[27]	0.38	159	Intel Core i7 3.40 GHz
Gokalp & Ugur (2020)	ILS-RVND[28]	0.45	151	Intel Core i7 3.40 GHz
Szeto et al. (2011)	ABO	0.46	264	Pentium 1.73 GHz
Reimann, Doerner & Hartl (2004)	SAS[29]	0.48	218	Pentium 900 MHz
Berger & Barkaoui (2003)	Hybrid GA	0.49	1275	Pentium 400 MHz
Ho & Gendreau (2006)	PR[30]	0.54	248	Pentium 2.53 GHz
Rego & Roucairol (1996)	PTS	0.55	1440	HP 9000/712
Rego (2001)	NEC	0.55	1501	33 MHz Sun IPC
Akpinar (2016)	LNS–ACO	0.55	N/A	Core i7 2.80 GHz

Continued on next page

[18] Edge assembly-based Memetic Algorithm

[19] Probabilistic diversification and intensification Local Search

[20] Active-guided evolution strategies

[21] Honey Bees Mating Optimization Algorithm

[22] Fireworks Algorithm

[23] Parallel Iterative Search Methods

[24] Hybrid of Ant Colony and Firefly Algorithms

[25] Multi-start ILS-RVND with adaptive acceptance strategies

[26] Multi-start ILS-RVND

[27] ILS-RVND with adaptive acceptance strategy

[28] Iterated Local Search and Random Variable Neighborhood Descent

[29] Savings based Ant System

[30] Path Relinking

Table D.5 – Continued from previous Table

Reference	**Heuristic**	**ADFB%**	**CPU time**	**Operating machine**
Cordeau, Laporte & Mercier (2001)	TS	0.56	1477	Pentium IV 2 GHz
Baker & Ayechew (2003)	GA	0.56	1747	Pentium 266 MHz
Lee, Lee, Lin & Ying (2010)	Enhanced ACO	0.57	3600	Pentium 4 3 GHz
Toth & Vigo (2003)	GTS	0.64	230	Pentium 200 MHz PC
Present work	Two-phase heuristic	0.74	190	Intel Core i5 1.60 GHZ
Gendreau, Hertz & Laporte (1994)	TS	0.86	2808	Silicon Graphics 36 MHz
Ai & Kachitvichyanukul (2009)	PSO	0.88	163	Intel P4 3.4GHz
Yurtkuran & Emel (2010)	HELA	1.04	132	Intel Core i2 2.00 GHz
Créput & Koukam (2008)	MSOM-L	1.20	1060	AMD Athlon 2000MHz
Rego (1998)	SEM	1.54	139	Intel P4 3.4 GHz
Osman (1993)	SA-TS	2.11	9060	VAX 8600
Créput & Koukam (2008)	MSOM-F	3.49	19	AMD Athlon 2000MHz
Ghaziri (1996)	HDN	5.37	7	N/A
Clarke & Wright (1964)	CWSA	7.58	N/A[31]	N/A
Bullnheimer et al. (1999)	Ant System	8.85	2100	Pentium 100 MHz

[31]The execution time depends on the machine on which the algorithm is run