AN EVOLUTIONARY ALGORITHM TO THE TWO-ECHELON LOCATION ROUTING PROBLEMS WITH HARD TIME WINDOWS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

MELİSSA MÜSLİM

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

FEBRUARY 2021

Approval of the thesis:

AN EVOLUTIONARY ALGORITHM TO THE TWO-ECHELON LOCATION ROUTING PROBLEMS WITH HARD TIME WINDOWS

submitted by **MELİSSA MÜSLİM** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences** ______________

Prof. Dr. Esra Karasakal
Head of Department, **Industrial Engineering** ______________

Prof. Dr. Haldun Süral
Supervisor, **Industrial Engineering, METU** ______________

Prof. Dr. Cem İyigün
Co-supervisor, **Industrial Engineering, METU** ______________

Examining Committee Members:

Assoc. Prof. Dr. Canan Sepil
Industrial Engineering, METU ______________

Prof. Dr. Haldun Süral
Industrial Engineering, METU ______________

Prof. Dr. Cem İyigün
Industrial Engineering, METU ______________

Assist. Prof. Dr. Sakine Batun
Industrial Engineering, METU ______________

Assoc. Prof. Dr. Mehmet Rüştü Taner
Industrial Engineering, TED University ______________

Date: 10.02.2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: MELİSSA MÜSLİM

Signature :

ABSTRACT

AN EVOLUTIONARY ALGORITHM TO THE TWO-ECHELON LOCATION ROUTING PROBLEMS WITH HARD TIME WINDOWS

MÜSLİM, MELİSSA
M.S., Department of Industrial Engineering
Supervisor: Prof. Dr. Haldun Süral
Co-Supervisor: Prof. Dr. Cem İyigün

February 2021, 82 pages

Rapid growth in freight distribution networks due to increasing demand created the necessity for effective and efficient methods for freight vehicle movements. Motivated by the effective distribution network design problems, a two-echelon location routing problem with hard time windows (2E-LRPTW) is studied. This problem combines two NP-Hard problems, including strategic and tactical decisions: the facility location problem (FLP) and the vehicle routing problem (VRP). In this study, the first echelon consists of city distribution centers (CDC) and satellites; the second echelon is constituted of interaction between satellites and customers. The network is connected through two types of vehicle fleets with different characteristics. Each type of vehicle leaves the corresponding facility during working hours and returns to it. Imposing capacity restrictions to both facilities and vehicles and adding hard time window constraints to customers, the problem complexity increases. Consequently, an evolutionary algorithm (EA) inspired by a genetic algorithm is proposed to solve large-size instances with good quality within a reasonable time. The EA decides which facilities to open, allocations, and resulting routes originated from each facility at both echelons. Computational experiments and results indicate the proposed EA capable of finding optimal solutions and improving the best-known solutions for some instances.

Keywords: Location-routing problem with time windows, Vehicle routing, Genetic algorithm, Evolutionary algorithm

ÖZ

ZOR ZAMAN PENCERELERİNE SAHİP İKİ KADEMELİ YER SEÇİMİ–ROTALAMA PROBLEMLERİNE EVRİMSEL BİR ALGORİTMA

MÜSLİM, MELİSSA
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Haldun Süral
Ortak Tez Yöneticisi: Prof. Dr. Cem İyigün

Şubat 2021 , 82 sayfa

Artan talep nedeniyle yük dağıtım ağlarında yaşanan hızlı büyüme, yük aracı hareketleri için etkin ve verimli yöntemlerin gerekliliğini ortaya çıkartmıştır. Dağıtım ağı etkili tasarım problemlerinden hareketle, zaman pencereli iki kademeli yer seçimi-rotalama problemi incelenmiştir. Bu problem, stratejik ve taktik kararlar dahil olmak üzere iki NP-Zor problemi birleştirir: tesis konum problemi ve araç yönlendirme problemi. Bu çalışmada, birinci kademe şehir dağıtım merkezleri ve uydular; ikinci kademe uydular ve müşteriler arasındaki etkileşimden oluşur. Ağ, farklı özelliklere sahip iki araç filosu aracılığıyla birbirine bağlıdır. Her araç türü mesai saatleri içinde ilgili tesisten ayrılır ve geri döner. Hem tesislere hem de araçlara kapasite kısıtlamaları getirerek ve müşterilere zor zaman aralığı ekleyerek problem karmaşıklığı artar. Sonuç olarak, büyük boyutlu örnekleri makul bir süre içinde iyi kalitede çözmek için genetik bir algoritmadan esinlenen evrimsel bir algoritma geliştirilmiştir. Algoritma, her iki kademede hangi tesislerin açılacağına, atamalara ve her tesisten çıkan rotalara karar verir. Yapılan deneyler ve elde edilen sonuçlar, bazı örnekler için en iyi çö-

zümleri bulabilen ve en iyi bilinen çözümleri geliştirebilen bir algoritma yaratıldığını işaret eder.

Anahtar Kelimeler: Zaman pencereleriyle yer seçimi-rotalama problemi, Araç rotalama, Genetik algoritma, Evrimsel algoritma

To my family.

ACKNOWLEDGMENTS

First of all, I want to express my sincere gratitude to my supervisor Prof. Dr. Haldun Süral and co-supervisor, Prof. Dr. Cem Iyigün for their encouragement, advice, criticism, and insights throughout the study. I also want to thank them for enabling me to work on this subject and their trust.

I want to share my gratitude with my jury members Assoc. Prof. Dr. Canan Sepil, Assist. Prof. Dr. Sakine Batun and Assoc. Prof. Dr. Mehmet Rüştü Taner for their valuable comments and suggestions to improve my thesis.

A special thanks to Dr. Mohammad Saleh Farham for conveying his experiences in problem structure and solution methods throughout the study and giving me all the support I needed to further my thesis.

I would like to thank Melis Boran for her support and friendship throughout my studies in METU.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

LIST OF ALGORITHMS

ALGORITHMS

LIST OF ABBREVIATIONS

2E-LRPTW	Two-echelon capacitated location routing problem with time windows
2E-OLRP	Two-echelon open location routing problem
2E-VRPTW	Two-echelon capacitated vehicle routing problem with time windows
ALNS	Adaptive large neighborhood search
BB	Branch-and-bound
BCRC	Best cost route crossover
BP	Branch-and-price algorithm
CDC	City distribution center
CLRP	Capacitated location-routing problem
CVRP	Capacitated vehicle routing problem
EA	Evolutionary algorithm
FLP	Facility location problem
FLPTW	Facility location problem with time windows
GA	Genetic algorithm
GRASP	Greedy randomized adaptive search procedure
ILS	Iterative local search
LP	Learning process
LRPTW	Location-routing problem with time windows
MDVRP	Multi-depot vehicle routing problem
MDVRPTW	Multi-depot vehicle routing problem with time windows
MIP	Mixed-integer linear programming
NSGA-II	Non-dominated sorting genetic algorithm- II

PR	Path relinking
TS	Tabu search algorithm
TW	Time window
VNS	Variable neighborhood search
VRPTW	Vehicle routing problem with time windows
SAA	Sample average approximation method

CHAPTER 1

INTRODUCTION

1.1 Motivation and Problem Definition

In today's world, transportation enables economic activities and transactions essential for human life, especially considering the increasing demand resulting from urbanization's rapid pace (United Nations, 2012 in [39]). Transportation networks have a complex structure since these networks bring different stakeholders at several decision levels. Although various stakeholders exist, the network's leading players are the producers, carriers, customers and government. Besides, while designing a chain, various objectives and outputs should be considered at strategic and operational levels for each player, which may be conflicting. Moreover, this structure, which requires long-term investment, can be affected by political, economic, and social changes at every level. Therefore, there is a need for systems that can quickly adapt to these changes and tools that can assist and develop the decision-making stages. A general perception for designing and planning a network in the freight transportation industry to improve economic efficiency and service quality is required [10].

Later, the needs of the actors in the system and their conflicting objectives have created adverse outcomes both economically and environmentally when there are no coordination and collaboration [15]. City logistics concept emerged to reduce the adverse effects of seeking individual interests. Here, the aim is to plan distribution effectively and efficiently, but it is necessary to design it by minimizing the undesirable effects on safety, the environment, and congestion. Although the freight distribution network allows companies to make profits, the negative effects, especially for people living in urban areas increase. For instance, using large trucks to satisfy the increasing demand or extending the fleet size causes congestion and environmental problems. Besides, the time interval of distribution is as important as the type and number of vehicles used. In this context, governments may force companies to make such plans by introducing specific regulations to distribute the freight accordingly. Consequently, city logistics aims to ensure that the increasing freight movement does not lower the quality of life in urban areas by designing and operating activities [33].

Advancements in e-commerce lead replacement of traditional shopping behavior with an online purchasing routine. Thus, convenience in online shopping increased the daily freight amount to be managed. In the conventional system, products are delivered to stores in large quantities, but now the relationship between the customer and the retailer increased the product's movement. A direct consumer-producer relationship may seem like a development, but considering the excessive purchasing urge becomes a challenge. As one can realize from growth in demand, a deficiency in the literature studies is not realistic to seek a solution assuming that physical facilities are not capacitated. Since the facility's size is a factor that increases operational and construction costs, modeling these networks under the assumption of without capacity is missing the essential constraints in the decision-making process.

Adding to these, online shopping and the adoption of just-in-time delivery from the manufacturers have put customer expectations first, and as a result, service quality has become a crucial performance indicator. Now a customer has more power in the system by organizing how and when to receive the product. Consequently, service reliability becomes a key indicator, and customers demanding to get service within the appropriate time interval causes the emergence of the hard time window structure. According to Crainic [10], a company should invest in points that will increase customer satisfaction and meet their service expectations while maximizing profits.

Considering the ability to adapt to continuous growth in demand and satisfy the expectations, we grasped the necessity of designing an adequate transportation chain. In line with this objective, logistics models exist to plan strategic, tactical, and operational levels in the literature. These models can have a single or multi-echelon structure. The city logistics concept underlines the necessity of using multi-layered logistic models to reduce congestion in urban areas. Unlike conventional approaches, these models build large facilities far from the city, open dynamic facilities to city centers, and distribute freight to customers with environmentally-friendly vehicles.

By considering all challenges and developments in the freight distribution networks, we propose a two-echelon LRP with hard time windows (2E-LRPTW) that combine strategic and tactical decisions to open facilities, assign customers, and create routes simultaneously. 2E-LRPTW is an NP-Hard problem that consists of FLP and VRP with hard time windows (VRPTW). Our model's primary facilities called City Distribution Centers (CDC), secondary facilities satellites, and customers are connected through a two-layered system. Freight comes to the CDC, then transportation to the satellites is performed by primary vehicles. After the consolidation processes are completed in satellites, the freight is transported to the customers with secondary vehicles. Each secondary vehicle leaves the satellite travels customers having hard time windows and returns to the satellite within working hours.

The complexity of the 2E-LRP increases by adding the facility capacities and imposing hard time window constraints to encounter customer needs. A few exact solutions approaches for 2E-LRP exists because of the computational complexity, and the problem sizes those exact procedures can solve within a reasonable time frame are small. Additionally, metaheuristic approaches exist to solve the problem, but none is a population-based algorithm. In order to fill the existing gap in the literature, we propose an EA with problem-specific operators that solves relatively large problem sets in a reasonable time with effective results. This proposed EA aims to generate offsprings by recombining the individuals selected randomly from the initial population and ensure evolution throughout generations.

1.2 Contributions and Novelties

This thesis aims to create an algorithm that produces solutions of good quality within a reasonable time motivated by the problems mentioned above. Moreover, a 2E-LRPTW is modeled and explained by emphasizing the constraints and complexities, considering it is a newly introduced problem. In this sense, our contributions can be listed as follows.

- This research is the second study with a distribution network model with facility and vehicle capacities at both echelons and customer hard time windows.
- To our knowledge, this thesis is the first study that proposing a population-based evolutionary algorithm inspired by the genetic algorithm for 2E-LRPTW, which gives solutions with high quality within a reasonable time limit for the large-size instances.
- Our operators are constructed to prevent infeasible solutions and solve problems simultaneously rather than decomposing problems at each echelon as the literature studies practiced.

A thesis carried out in line with these motivations and goals is structured as follows. In Chapter 2 we demonstrate the existing 2E-LRP studies with their structural properties and create a link between related problems such as 2E-VRP and LRP literature. The literature gap is emphasized by discussing the strengths and weaknesses of the models with constructed solution approaches. In Chapter 3 the constituted mathematical model is introduced with the necessary assumptions, sets, and decision variables. Additionally, by explaining the constraints, the necessity of metaheuristic approaches with difficulties are highlighted. The proposed solution methodology for a model with one CDC at the first echelon is explained in detail with evolutionary operators, charts, and pseudo-codes in Chapter 4. Later in Chapter 5, we reveal the necessary modifications to make the proposed algorithm suitable for solving two-echelon structures that decide both CDC and satellite locations with resulting routes. Chapter 6 presents benchmark instances and analysis and the EA results for both modified and original sets. The last chapter contains a summary of the completed study and the motivation behind it. The strengths and improvable aspects of the results obtained from the algorithm and the future research areas have been discussed.

CHAPTER 2

LITERATURE REVIEW

With the rapid growth in consumption of goods and services, economic and environmental concerns arose and, designing a proper distribution network gained importance. For these reasons, while designing logistic supply chains, environmental, humanitarian, and commercial activities are taken into account, and mathematical models are constructed accordingly. Our model 2E-LRPTW has directly intertwined these activities such as changing the objective, the purpose of the model can be changed, but the structure stays the same. Therefore, considering that all the areas and models are varied, we have considered the structural similarity of the literature models while reviewing.

Distribution network models are mainly classified as single or multi-echelon systems throughout the literature review. Single echelon systems consist of the facility location problem (FLP), vehicle routing problem (VRP), and location routing problem (LRP). First one, FLP deals with a strategic problem to determine where candidate facilities should be located to minimize the customers' total cost and assignment to the opened facilities. Secondly, VRPs work with the currently open facilities and determine customers' assignments then the resulting routes rather than constructing direct trips between facilities and customers as in FLPs. Lastly, LRPs combines strategic and tactical decisions to open facilities, assign customers, and create routes simultaneously. If we take these models one step further, we get multi-echelon versions, namely 2E-FLP, 2E-VRP, and 2E-LRP. These two-echelon systems are obtained by adding an intermediate level facility called satellites between the CDCs and customers. Now, the distribution in the first echelon is performed by huge trucks, while in the second echelon, environment-friendly smaller vehicles are used. Adding capacity restrictions to the facilities and time windows to customers will also increase the model's diversity and complexity. In this study, the studied 2E-LRPTW model is classified as a multi-echelon system consisting of two LRPs at both echelons. According to one other definition, 2E-LRP combines a 2E-FLP and a 2E-VRP in terms of the decisions and constraints.

All the mentioned distribution network models have variations in data definitions, timings, capacity definitions, and data uncertainty. Our model is constructed considering the complete literature of 2E-LRPs and presented the limitations and the structures of the existing models. We propose an evolutionary algorithm (EA) to solve 2E-LRPTW; therefore, we narrowed down the reviews of VRPs and LRPs studies having the (meta)heuristic solution approaches. Before going into the details of the 2E-LRP, the reader will see the (meta)heuristic algorithm studies in the VRP and LRP literature in Sections 2.1 and 2.2, respectively and then the complete literature of 2E-LRP (2.2.1).

2.1 Vehicle Routing Problem

This section focuses mainly on (meta)heuristic studies for models having multi-depot and time windows in VRPs, excluding exact solution methods. Cordeua et al. [8] constructed a tabu search (TS) algorithm to solve multi-depot VRP with time windows (MDVRPTW). Having simple neighborhood definitions, customers exchanged between one route to another, and penalty costs are added to the objective because of infeasible solutions allowed throughout the algorithm. A hybrid genetic algorithm (GA) for the VRPTW is studied by Berger and Barkaoui [4]. They tried to solve two GAs simultaneously having different objectives, namely minimizing total distance and constraint violation. Polacek et al. [29] proposed a variable neighborhood search (VNS) for MDVRPTW. In this study, the algorithm enhanced with exchange operators and a 3-opt algorithm as a local search operator. Then Ombuki and Berger [27] constructed a GA restricting capacity and route-length on individuals. They adapted the best cost route crossover (BCRC) as in their former research with some improvements. As mutation operators, inter and intra route improvement algorithms are used, and offsprings always replaced parents. Then, Karakatic et al. [19] demonstrated the genetic algorithm-based solutions to solve MDVRP. The work of the authors mentioned the limitations and advantages of solution representations of studies in the literature. Moreover, this study compared the algorithms according to their selection, crossover, and mutation operators and tested them. They pointed out that ordered crossover, tournament selection should be used in future GA studies. The reader may refer to the study to gain an understanding of GAs for MDVRP.

Hemmelmayr et al. [16] proposed an adaptive large neighborhood search heuristic (ALNS) for 2E-VRP and LRP. They allowed vehicle capacity, the number of available vehicles, and satellite capacity violations and managed to control them using penalty costs for such solutions to widen the search space. Solutions are first destroyed, then repaired, and the final solutions proceed into the local search phase. Wang et al. [37] introduced a 2E-VRP model having stochastic demands and proposed a genetic-based algorithm to solve it. Rather than using the objective function as a fitness score, a penalty coefficient and expected total excess quantity of vehicles are added to the objective function since authors allow infeasible solutions to relax the capacity constraints.

2.2 Location Routing Problem

FLPs are the models for strategic level decisions because they require considerable investments to locate a physical facility, so they are considered a long-term plan. On the contrary, VRPs are considered (mid) short-term plans and built at a tactical level. According to the literature studies, the distribution networks built up not conceiving these two problems together probably have a high total cost. Therefore, the need for LRPs arises, which integrates strategic and tactical decision levels by combining facility locating and routing decisions simultaneously.

Derbel et al. [14] proposed a GA, which includes their former iterative local search (ILS) as a local search procedure in the algorithm. In the algorithm, chromosomes consist of two vectors. One of them is an allocation vector, and the other one keeps the routes of depots. Apart from the classical one-point crossover, mutation operators aim to improve solutions by changing allocation and routes. They achieved to obtain improved results with a higher run time, but their vehicles are uncapacitated. Jarboui et al. [18] solve LRP with capacitated depots but uncapacitated vehicles using variable neighborhood search (VNS). Routes are represented with an array starting and ending with open depots. Neighborhood structures aiming to diversify route and location decisions. Lopes et al. [21] studied a GA with a variable chromosome length representing vehicle routes starting with the assigned facility. New offsprings replace the two worst individuals obtained by route copy crossover. Solutions are improved with local search procedures by changing facility configurations and routes. For more solution procedures in this area, please refer to surveys of [32] and [34].

2.2.1 The Two-Echelon Location-Routing Problem

As mentioned earlier, 2E-LRP requires a decision of facility opening, allocation, and routing at both echelons simultaneously. Considering the complexity of the problem, some of the studies focused on models having 1 CDC at first echelon having direct trips to open satellites and only location decision for satellites at the intermediate level and routing decisions at the second echelon although having models of 2E-LRP. The reader may realize that these models do not have a classical 2E-LRP structure, but these are the classical models' variants having fewer decision variables.

Jacobsen and Madsen [17] are the ones who first considered the multiple echelons in a location routing problem. They proposed three heuristic procedures for a newspaper distribution system in Denmark. Computational results demonstrated that these heuristics generate total costs above the current system. Following Madsen [22], tried to improve the heuristics and archived to have significant improvements in the current system. However, in both research pieces, only one real instance is tested, and results are only comparable with the current system.

Boccia et al. [5] proposed TS, which is introduced as an "iterative-nested approach" for a two-echelon capacitated LRP (2E-CLRP). Consequently, the problem decomposed into two separate LRPs, each consisting of a capacitated FLP (CFLP) and a MDVRP. Although some of the solutions cannot outperform the decomposition method, overall TS can be considered as an effective algorithm. After that, Crainic et al. [9] proposed three mixed-integer programming models for 2E-LRP and compared each model's flexibility.

Nguyen et al. [25] introduced three constructive greedy randomized heuristics as well as a hybrid metaheuristic called greedy randomized adaptive search procedure (GRASP) reinforced with the learning process (LP) and path relinking (PR). Each construction heuristic first constitutes a second-echelon solution then constructs first level routes accordingly. The developed GRASP is composed of one diversification and one intensification phase at each iteration. There existed no publicly available 2E-LRP instances up to that time; two new sets generated, containing only 1 CDC in the first echelon, so there is no location decision for the CDCs exist. Results indicated that GRASP-LP outperforms constructive heuristics in both instances. Nguyen et al. [24] proposed a multi-start iterated local search additionally having a tabu list and PR. Moreover, in this research, they introduced new instances constructed by Sterle having multiple candidate CDCs at the first echelon. Findings revealed that the algorithms in the new study outperform their former approach.

Contardo et al. [7] research consist of two major parts, proposing an exact and metaheuristic method by decomposing 2E-CLRP into two separate CLRPs. A new two-index vehicle flow formulation, which is strengthened by new valid inequalities, is solved using Branch and Cut (BC) method. They are the ones who first attempted to solve this class of problems exactly. Secondly, an ALNS procedure is created for 2E-CLRP, modifying the previous work of [16] for 2E-VRP and CLRP. Numerical results demonstrate that ALNS outperforms both [25] and [24] on each of the sets. However, only the best solutions were reported regardless of the parameter setting.

Schewengener et al. [35] proposed a VNS algorithm modifying their previous work for a location-routing problem. The developed VNS algorithm ensures diversification by shaking with seven neighborhood structures also, intensification by applying 2-opt* and 3-opt algorithms. Computational experiments revealed that although VNS can both outperform [25] and [24]. On the contrary, work of [7] outperforms VNS on the sets having multi-platforms.

Winkenbach et al. [39] introduced a two-stage iterative optimization model, which first solves the facility location subproblem and then the VRP and compares it with single-stage optimization 2E-CLRP. Rather than obtaining routing cost from the objective, they constructed a closed-form approximation for optimal routing cost that considers service time constraint and other economic measures.

Then, Breunig et al. [6] presented a hybrid metaheuristic consisting of local search with destroy and repair operators for a 2E-LRP. The metaheuristic indeed developed for a 2E-VRP. The procedure first destroys the partial solutions of second-level routes, then the second level solution is repaired and improved using a local search algorithm, and finally, the first level solution is regenerated. The robustness of the heuristic is tested using 2E-LRP having instances with 1 CDC. Computational experiments pointed out that their algorithm at least competitive as the VNS [35] method.

Wang et al. [38] demonstrated an approach combining customer clustering and modification of Non-dominated Sorting Genetic Algorithm- II [13] to solve a 2E-LRP having soft time windows. The model is constructed to minimize costs and maximize customer satisfaction, which is evaluated by vehicle punctuality. The proposed approach first estimates demand using exponential smoothing and segments customers using the k-means algorithm. This step followed by a distribution centers' location problem, and finally, M-NSGA-II generates routes on both echelons. Since none of the literature is suitable for comparison, the authors compared the algorithm's capability with Multi-Objective Genetic Algorithm (MOGA) and Multi-Objective Particle Swarm Optimization (MOPSO), and mostly the developed algorithm performs better than the other two.

Different than previous researchers, Pichka et al. [28] introduced three mixed-integer linear programs (MILPs) and a two-stage heuristic for a two-echelon open LRP (2E-OLRP). Both echelons' vehicle routes do not need to end at a satellite or a depot in this context. The hybrid heuristic is capable of finding optimal solutions for small-size and reasonable solutions for medium-sized instances. Nonetheless, [24] and [7] models outperform the hybrid heuristic. Although each of the authors studied 2E-LRP, the problem structure was not identical.

Amiri et al. [1] constructed a supply vessel planning model to a two-echelon fleet composition mix periodic LRP having time windows for onshore bases and offshore units. These time windows are only working hours. Using a Lagrangean decomposition method, the number and type of vehicles, routes, schedules, and allocations are decided at each echelon. However, only the results of a case study having small size instances are demonstrated. Darvish et al. [12] considered the flexibility in service time of customers and distribution network design by constructing a two-echelon network over a planning horizon. Each day, the supplier has to decide on open intermediate facilities, allocations, and routing using the enhanced parallel exact method (EPEM), a branch-and-bound method (BB). Another study comes from Mirhedayatian et al. [23], which includes synchronization into the 2E-LRP because, in transshipment points like satellites, there is limited storage and waiting times. In their mathematical model, customers have time windows as well as secondary facilities. Three stages as opening facilities, allocation, and routing are used to solve the problem. Measures like farness and closeness are introduced to narrow down the solution space. The best-known solutions were obtained for 26 out of 112 instances, and no feasible solution was obtained using the MILP formulation for large-sized instances. Mohamed et al. [3] study designs a two-echelon distribution network (2E-DDP) under uncertain and time-varying demand and opening costs. 2E-DDP formulated as a multi-stage stochastic program, and the solution methodology relies on a benders decomposition and sample average approximation methods (SAA).

The last study that we need to take into account is Farham's [15] doctoral thesis. His model is constructed as a classical 2E-LRP by adding capacity constraints at each echelon, also defining hard time windows to customers in the network. Up to this time, none of the studies considered opening, allocation, and routing decisions at both echelons having hard time windows. The author developed a branch-and-price (BP) algorithm with enhanced column generation techniques, which take all three decisions simultaneously. Considering the computational complexity, two math-heuristic approaches developed, namely top-to-bottom and bottom-to-top. The top-to-bottom approach is integrated with BP, while the bottom-to-top uses clustering algorithms.

Table 2.1 lists the solution methodologies with corresponding largest instances solved. The problem size in the Table 2.1 refers to the instance size in the order of the number of candidate CDC, candidate satellite, and customers. The reported values are the largest instance sizes that can be sold using the corresponding solution algorithm. Then, a comprehensive summary of the discussed 2E-LRP models regarding their structure is demonstrated in Table 2.2.

Table 2.1: Solution methods for the two-echelon location routing problems

Reference	Solution Algorithm	Problem Size
Jacobsen Madsen [17]	Problem specific heuristics	1-42-4510
Madsen [22]	Problem specific heuristics	1-42-4510
Boccia et al. [5]	TS	5-20-200
Nguyen et al. [25]	GRASP with learning process (LP) and path relinking (PR)	1-10-200
Nguyen et al. [24]	Multi-start ILS with PR and tabu list	5-20-200
Contardo et al. [7]	Branch-and-cut	1-10-50, 4-10-25
Contardo et al. [7]	ALNS	5-20-200
Schwengerer et al. [35]	VNS	5-20-200
Breunig et al. [6]	LNS-2E	1-10-200
Wang et al. [38]	M-NSGA-II	1-16-100
Pichka et al. [28]	Hybrid heuristic	1-10-200
Amiri et al. [1]	A lagrangean decomposition method	2-3-4
Darvish et al. [12]	Pure branch-and-bound, EPEM	1-3-60
Mirhedayatian et al. [23]	Decomposition based heuristic	1-5-70
Mohamed et al. [3]	Benders decomposition, SAA	4-8-100
Farham [15]	BP, bottom-to-top, top-to-bottom	3-5-100, 6-4-100

According to the Table 2.1 we can say that the largest instances solved using and exact method belongs to [7]; however, the size is still considerably small for the real-life cases, especially for the number of customers.

2E-LRP is known to be an NP-Hard problem that leads to exact solution methods only capable of giving results for the small to medium size instances. Having a model with two echelons increases the problem complexity. Because trying to take the opening, allocation, and routing decisions for two echelons simultaneously is hard to deal with. Therefore, researchers mainly focused on solving 2E-LRPs using heuristic solution approaches for large-size instances.

In Table 2.2, decisions at each echelon with facility capacity properties and whether a customer has a time window or not is demonstrated. Considering the Table 2.2, we can see that some of the studies have CDCs or satellites without capacities. Even for the capacitated models, since there is 1 CDC at the first echelon, the capacity is set to customers' total demand. As one can imagine, this is a strong assumption because facilities are physical constructions that have boundaries. In addition to the capacities, time windows can be considered one of the crucial components in today's distribution

Table 2.2: Model structures of the reviewed papers

Reference	Time Window		Second Echelon			First Echelon		
	Soft	Hard	Routing	Capacity (Satellite)	Location (Satellite)	Routing	Capacity (CDC)	Location (CDC)
Jacobsen Madsen [17]			✓	✓	✓	✓		
Madsen [22]			✓	✓	✓	✓		
Boccia et al. [5]			✓	✓	✓	✓	✓	✓
Nguyen et al. [25]			✓	✓	✓	✓		
Nguyen et al. [24]			✓	✓	✓	✓	✓	✓
Contardo et al. [7]			✓	✓	✓	✓	✓	✓
Schwengerer et al. [35]			✓	✓	✓	✓	✓	✓
Breunig et al. [6]			✓	✓	✓			
Wang et al. [38]	✓		✓	✓	✓	✓	✓	
Pichka et al. [28]			✓	✓	✓	✓		
Amiri et al. [1]			✓	✓	✓	✓	✓	✓
Darvish et al. [12]			✓	✓	✓			
Mirhedayatian et al. [23]		✓	✓	✓	✓			
Farham [15]		✓	✓	✓	✓		✓	✓

systems. Although there exist dozens of state-of-art articles on 2E-LRP, the problem of 2E-LRPTW is only mentioned in three studies. These studies cannot be compared with each other because of the other properties of their models. For instance, soft time window models are easier to deal with than hard time window models. Because, outside the time interval for the soft time window case, the solution is not infeasible such that by giving a penalty to solutions situation can be handled. However, it is impossible to provide service before and after the time interval in models with hard time windows. In that case, constraints are violated; thus, the solution is considered infeasible. Although [23] has hard time windows in the proposed model, there is no location decision in the first echelon, which reduces the problem complexity compared to the model of [15].

From the tables, we can conclude that there exists a gap for the models having hard time windows with facility capacities. Moreover, there needs to be a solution methodology that gives solutions with high quality within a reasonable time limit for the large-size instances. Although there exist heuristic procedures in the literature, none of these methods have benefited from the performance of population-based approaches for 2E-LRPTW. For all the reasons pointed out so far, we are proposing a model with hard time windows and an EA inspired by a GA for solving 2E-LRPTW.

CHAPTER 3

MATHEMATICAL MODEL

In this chapter, we provided the general setting of the 2E-LRPTW with the detailed description of the proposed formulation. Using a mathematical model generally is not the priority in algorithms consisting of heuristic or metaheuristic solution methods; introducing a model for the problem we have is crucial to understanding the solution method's challenges and constraints. Since all constraints shown in the mathematical model must be checked at every stage of the solution algorithm, we have to develop a solution methodology that overcomes the complexity of creating feasible solutions.

In our logistics framework CDCs, satellites, and customers are connected through a two-echelon system using two types of vehicle fleets having different characteristics. Two- echelon systems are introduced to solve pollution, congestion, and logistics problems, especially in large cities [15]. In these systems, customers' deliveries go through a two-step process rather than a direct one-step process, and there is no direct access from a CDC to a customer. Freight first comes to the CDC, a primary facility; after the consolidation processes are completed, it is transported to the satellites with first echelon vehicles. Every freight that reaches a satellite is then transported to the customers with second-echelon vehicles. Each vehicle leaves the satellite during working hours, travels customers on its route with hard-time windows, and returns to the corresponding satellite. An illustration of the system can be seen in Figure 3.1.

The provided mathematical model for the 2E-LRPTW is derived from the formulations proposed by Farham [15]. In 2E-LRPTW, the complete transportation network is assumed to be comprised of three sets of nodes. The network is linked with directional arcs among CDC nodes $i \in \mathcal{I}$ and satellite nodes $j \in \mathcal{J}$, and customer locations $k \in \mathcal{K}$. Transportation between the same type of facilities and two-way flows among CDCs and customers are not permitted.

The complete node-set is divided into two sets to represent each echelon separately. The first set comprises candidate CDC and satellite locations $m \in \mathcal{M}$ where $M = \mathcal{I} \cup \mathcal{J}$, and the second one consists of satellite and customer nodes $n \in \mathcal{N}$ where $N = \mathcal{J} \cup \mathcal{K}$. Also, first echelon arcs represented by $\mathcal{E}'$ while $\mathcal{E}''$ demonstrates second echelon arcs.

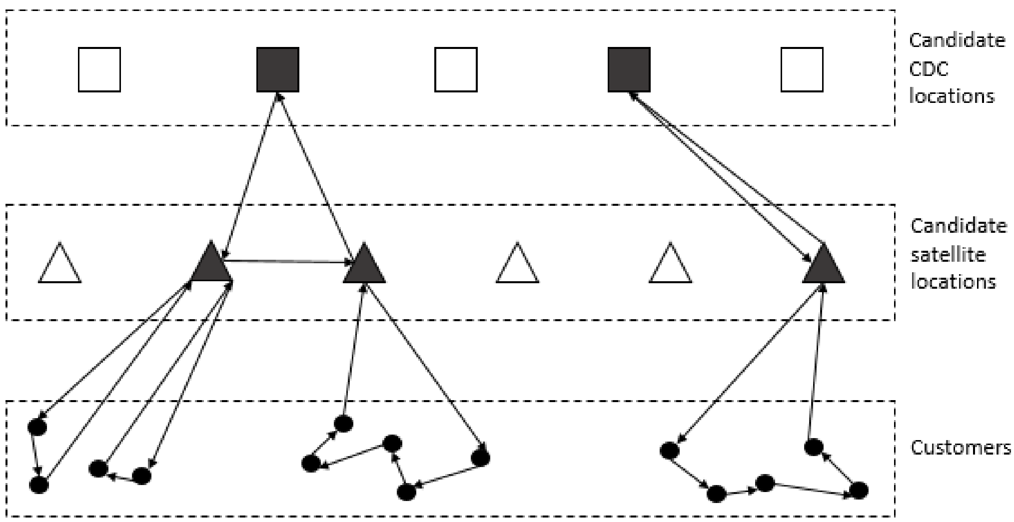

Figure 3.1: Two-echelon location routing model representation

Each customer k has a demand D_k, distinct time windows [A_k, B_k] and a non-negative service time which supplied from satellite j. In the models having customers with hard time windows, it is assumed that if the vehicle reaches the customers earlier than A_k orders are accepted, but the vehicle must wait until the customer's ready time is reached. However, orders are strictly rejected if the vehicle reaches the customers later than B_k, resulting in infeasibility in the system.

Primary and secondary facilities have an opening fixed cost F_m and capacity Q_m. Locating a primary facility is a long-term decision since large structures' construction requires huge investments and has no replacement flexibility. On the other hand, satellites are remote locations, like parking lots, in cities where no inventory operations are possible. Also, each satellite j is available during working hours [0, B_j] while no time window is defined for the CDCs.

The two types of homogeneous fleet of vehicles sustain transportation between the locations on both echelons. CDCs are favored to be constructed in the outer skirt of the urban areas due to requiring broad grounds. So, we can benefit from huge trucks called primary vehicles to serve satellites. Since it is challenging to use huge vehicles in city centers, small secondary vehicles serve customers in the second echelon. Each primary vehicle has a fixed usage cost F' and capacity Q' while each secondary vehicle has a fixed usage cost F'' and capacity Q''. Fixed costs are maintenance costs that must be paid to keep the vehicle ready for use and emerge for each route. We define time T_{mn} by including service time at location m and the time required to travel from m to n which costs C_{mn} units where $(m, n) \in \mathcal{E}' \cup \mathcal{E}''$. Lastly as a parameter, we have B_{mn} is a constant equal to the $max\{B_m + T_{mn} - A_n, 0\}$.

Table 3.1: Mathematical model parameters

Notation	**Description**
$\mathcal{I}$	Set candidate CDC nodes
$\mathcal{J}$	Set of candidate satellite nodes
$\mathcal{K}$	Set of customer nodes
$\mathcal{M}$	Set of first echelon nodes, $m \in \mathcal{I} \cup \mathcal{J}$
$\mathcal{N}$	Set of second echelon nodes, $n \in \mathcal{J} \cup \mathcal{K}$
$\mathcal{E}'$,$\mathcal{E}''$	Set of first and second echelon arcs
F_m	Fixed cost of opening facility $m \in \mathcal{M}$
Q_m	Capacity of $m \in \mathcal{M}$
F',F''	Fixed usage cost of primary and secondary vehicles
Q',Q''	Capacities of first and second echelon vehicles
D_k	Demand of customer $k \in \mathcal{K}$
$[A_n, B_n]$	Time window of node $n \in \mathcal{N}$
C_{mn}	Travelling cost of edge $(m, n) \in \mathcal{E}' \cup \mathcal{E}''$
T_{mn}	Travelling time between of edge $(m, n) \in \mathcal{E}' \cup \mathcal{E}''$ including the setup/service time at node m

3.1 Decision Variables

The decision variables required to construct the mathematical model should also be considered the decisions we have to make during each step of the proposed solution method. In order to take facility opening decisions, z_m be the binary decision variable indicating that facility m is opened or not. The binary decision variable r_{jk} represents the allocations in the second echelon by indicating if customer k is assigned to satellite j or not. The variable x_{jk} takes value 1 if a secondary vehicle travels on arc $(j,k) \in \mathcal{E}''$, 0 otherwise.

In the second echelon, a customer can only be served by one vehicle that originated its route from a single satellite. Nonetheless, in the first echelon, multi-sourcing is allowed implying that a satellite can be served more than one vehicle and multiple CDCs. Therefore, a vehicle index for primary vehicles $v \in \mathcal{V}$ is introduced to keep track of each primary vehicle's flows.

Table 3.2: Decision variables

Notation	**Description**
z_m	Facility $m \in \mathcal{M}$ is opened or not
r_{jk}	Whether customer $k \in \mathcal{K}$ is assigned to satellite $j \in \mathcal{J}$ or not
x_{jk}	Whether a secondary vehicle is traveled on arc $(j,k) \in \mathcal{E}''$ or not
y^v_{mn}	Whether a primary vehicle $v \in \mathcal{V}$ is traveled on arc $(m,n) \in \mathcal{E}'$ or not
w^v_{ij}	The weight of flow sent from CDC $i \in \mathcal{I}$ to satellite $j \in \mathcal{J}$ on vehicle $v \in \mathcal{V}$
q^v_j	Load on vehicle $v \in \mathcal{V}$ upon arriving at satellite $j \in \mathcal{J}$
q_k	Load on secondary vehicle upon arriving at customer $k \in \mathcal{K}$
t_n	The arrival time to node $n \in \mathcal{N}$

Let decision variable y^v_{mn} indicates whether a primary vehicle v traverses on arc $(m,n) \in \mathcal{E}'$. To be able to calculate the flow from CDC i to satellite j on primary vehicle v, variable w^v_{ij} is defined. Let a non-negative decision variable q^v_j demonstrates the load on the vehicle v upon arriving at satellite j while the load on the vehicle upon arriving at customer k is controlled by q_k. Lastly, a non-negative variable t_n indicates the arrival time to node $n \in \mathcal{N}$ in the second echelon.

3.2 The Two-Echelon Location-Routing Problem with Time Windows

The mathematical formulation for a two-echelon distribution network model, which we wish to reflect on real-life problems, is created under some assumptions. In the 2E-LRPTW model, it is assumed that the model's static behavior is homogeneous throughout the planning horizon. One of the essential features of the model is that customers have hard time windows. In other words, it is not possible to provide service to the customer outside the specified intervals, and it is not possible to obtain feasible solutions by adding penalty costs to the objective in case of a violation, as in problems with soft time windows. If service is not provided to the customer within the specified time window, that solution is infeasible. While creating the mathematical formulation of 2E-LRPTW, the following assumptions are also considered:

1. In the formulation, there is a single commodity, and the flow of that commodity starts and ends at corresponding facilities.

2. Both CDCs and satellites are capacitated, and it is not possible to violate the capacities.

3. The customer demands cannot be split in the second echelon; however, multiple vehicles can serve from multiple CDCs to satellites.

4. The loading/unloading duration of vehicles at facilities are not included.

5. The number of vehicles is not restricted in both echelons. Also, secondary vehicles' capacity is lower than the primary vehicles' capacity since primary vehicles generally serve the outskirts of the urban zones.

6. The cost of opening primary facilities is the strategic decisions requiring long-term investment, but the routing decisions are taken on the tactical level. In the objective function, we combine both long-term and short-term costs to obtain solutions giving minimum costs. Because studies demonstrated that strategic decisions influence tactical and operational levels that indicate a facility's location affects routings and transportation costs [9]. Therefore, we are focusing on the dependency between location and routing decisions simultaneously.

Under the mentioned assumptions and decision variables, the 2E-LRPTW is formulated as the mixed-integer linear programming (MIP).

$$\begin{aligned} \text{minimize} \sum_{m\in\mathcal{M}} F_m z_m + \sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} F' y_{ij}^v + \sum_{j\in\mathcal{J}}\sum_{k\in\mathcal{K}} F'' x_{jk} \\ + \sum_{v\in\mathcal{V}}\sum_{(m,n)\in\mathcal{E}'} C_{mn} y_{mn}^v + \sum_{(m,n)\in\mathcal{E}''} C_{mn} x_{mn} \end{aligned} \tag{3.1}$$

$$\text{subject to} \sum_{j\in\mathcal{J}} r_{jk} = 1 \qquad \forall k \in \mathcal{K} \tag{3.2}$$

$$\sum_{n\in\mathcal{N}} x_{nk} = 1 \qquad \forall k \in \mathcal{K} \tag{3.3}$$

$$\sum_{m\in\mathcal{N}} x_{nm} - \sum_{m\in\mathcal{N}} x_{mn} = 0 \qquad \forall n \in \mathcal{N} \tag{3.4}$$

$$x_{jk} \le r_{jk} \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{3.5}$$

$$r_{jk} + x_{kl} \le 1 + r_{jl} \qquad \forall j \in \mathcal{J}, k, l \in \mathcal{K} \tag{3.6}$$

$$q_k + D_k - q_l \le Q''(1 - x_{kl}) \qquad \forall k, l \in \mathcal{K} \tag{3.7}$$

$$D_k \le q_k \le Q'' \qquad \forall k \in \mathcal{K} \tag{3.8}$$

$$\sum_{k\in\mathcal{K}} D_k r_{jk} \le \sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{I}} w_{ij}^v \qquad \forall j \in \mathcal{J} \tag{3.9}$$

$$\sum_{v\in\mathcal{V}}\sum_{i\in\mathcal{I}} w_{ij}^v \le Q_j z_j \qquad \forall j \in \mathcal{J} \tag{3.10}$$

$$T_{jk} - t_k \le B_{jk}(1 - x_{jk}) \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{3.11}$$

$$t_k + T_{kn} - t_n \le B_{kn}(1 - x_{kn}) \qquad \forall k \in \mathcal{K}, n \in \mathcal{N} \tag{3.12}$$

$$A_n \le t_n \le B_n \qquad \forall n \in \mathcal{N} \tag{3.13}$$

$$\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{J}} y_{ij}^v \le 1 \qquad \forall v \in \mathcal{V} \tag{3.14}$$

$$\sum_{v\in\mathcal{V}}\sum_{j\in\mathcal{J}} w_{ij}^v \le Q_i z_i \qquad \forall i \in \mathcal{I} \tag{3.15}$$

$$\sum_{n\in\mathcal{M}} y_{mn}^v - \sum_{n\in\mathcal{M}} y_{nm}^v = 0 \qquad \forall m \in \mathcal{M}, v \in \mathcal{V} \tag{3.16}$$

$$\sum_{i\in\mathcal{I}} w_{ij}^v \le q_j^v \le Q' \qquad \forall j \in \mathcal{J}, v \in \mathcal{V} \tag{3.17}$$

$$q_j^v + \sum_{i\in\mathcal{I}} w_{ij}^v - q_l^v \le Q'\left(1 - y_{jl}^v\right) \qquad \forall v \in \mathcal{V}, j, l \in \mathcal{J} \tag{3.18}$$

$$0 \le w_{ij}^v \le Q' \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, v \in \mathcal{V} \tag{3.19}$$

$$z_m \in \{0, 1\} \qquad \forall m \in \mathcal{M} \tag{3.20}$$

$$x_{nm} \in \{0,1\} \qquad \forall (n,m) \in \varepsilon'' \tag{3.21}$$

$$r_{jk} \in \{0,1\} \qquad \forall j \in \mathcal{J}, k \in \mathcal{K} \tag{3.22}$$

$$y_{ij}^{v} \in \{0,1\} \qquad \forall i \in \mathcal{I}, j \in \mathcal{J}, v \in \mathcal{V} \tag{3.23}$$

The objective function (3.1) consists of three main components: total facility opening costs, vehicle usage costs, and routing costs in both echelons. Vehicle usage cost arises for each route, and the routing cost indicates the total distance traversed. Constraint (3.2) imposes that each customer can only be assigned to a single satellite and (3.3) does not allow split deliveries in the second echelon, ensuring that the freight is supplied to a customer by precisely one vehicle. Constraint (3.4) balances the incoming and outgoing flow to each second echelon node. According to constraints (3.5) and (3.6) an arc (j,k) can be traversed if a customer k is assigned to satellite j. Miller-Tucker-Zemlin constraint (3.7) is introduced to eliminate subtours and control the load of a secondary vehicle on its route. Constraint (3.8) forces secondary vehicle load to be greater than customer demands and at most equals to its capacity upon arriving at the customer. While (3.9) controls the flow coming into a satellite by ensuring the quantity should be at least the total customer demand it serves, constraint (3.10) makes sure that the flow cannot exceed an available satellite's capacity. Constraints (3.11) to (3.13) are hard time window constraints imposing that arrival time to each customer should be within pre-determined time limits.

Constraints (3.14)- (3.19) are responsible for maintaining feasibility in the first echelon. Constraint (3.14) ensures that each primary vehicle can be used at most once. Constraint (3.15) provides that the outgoing flow from an open CDC to an available satellite can be at most equals to CDC capacity. By (3.16), flow balance on the first echelon nodes is met. Constraint (3.17) prevents primary vehicle capacity violations and makes sure that flow on primary vehicles should be non-negative. Constraint (3.18) eliminates sub-tours as well as providing feasible loads on primary vehicles. Constraint (3.19) enables that the amount carried in the first echelon does not exceed the primary vehicle's capacity and is non-negative. Finally, (3.20) to (3.23) declare the domains of the decision variables.

This mathematical model is constructed as a three-index formulation since routing variables are controlled using three-indexed decision variables. This formulation is miscellaneous but more complicated compared to the mathematical models having one or two-index. Realize that the 2E-LRPTW is NP-Hard and reduces to a two-echelon FLP when there is a direct link between CDCs and customers. If the facility locations are given, the problem is transformed into a two-echelon VRPTW [9]. Additionally, the hard time window added complexity to the problem compared to the models with soft time windows. In models having a soft time window, solutions are not considered infeasible if service is not performed between customer time intervals. The violation is only reflected as a cost component in the objective, while in a hard time window setting, the solution is infeasible when the constraints are not satisfied.

The complexity leads to exact methods to solve only small-sized instances. Metaheuristic approaches can solve large size instances providing good solutions but, they are not different from the exact approaches when sustaining feasibility. Because the reader may realize that to construct a feasible solution, all constraints should be checked during each construction and evolution phase throughout the algorithm. To better understand adopting a solution procedure to a mathematical model with capacities and hard time windows, the following summary will be helpful. Note that our model decomposed into three stages, namely locating facilities, allocation, and routing phases.

1. When the number and locations of the facilities are determined, customers' allocations should be done checking the following: the facility capacity, customer's latest time window, and distance between the customer and the latest time window of the facility.
2. After the allocation process is completed, it is necessary to check the vehicle capacity, the time window feasibility to create routes for each satellite, and ensure that all customers are visited once.

It is difficult for a metaheuristic approach to produce many different solutions while checking so many constraints at each stage. The solution diversity of the algorithm is limited with each constraint checked. Beasley [2] reported that even when the existing solution methods perform well, improvements can be achieved by hybridizing the GA. That is why, in this study, we are hybridizing the GA with well-known construction and improvement heuristics to obtain an EA that combines both exploration and exploitation. To manage the complexity, we first proposed a solution framework to solve problems with given CDC locations by making them parameters and reducing two types of facility location decisions to one. Then we are extending the proposed algorithm by also deciding the locations for CDCs along with allocation and routing problems simultaneously without decomposing the decisions at each echelon.

CHAPTER 4

SOLUTION APPROACH FOR LRP WITH HARD TIME WINDOW PROBLEMS

The solution framework we propose can solve the 2E-LRPTW problem. In cases where open CDC locations are known, the problem is reduced to a single echelon, implying a location decision for only satellites, but allocation and routing decisions exist for both echelons. To keep narration comprehensible in Chapter 4, we describe our algorithm to solve single echelon structures. In the next chapter, the necessary modifications to simultaneously solve the two-echelon structures having location, allocation, and routing decisions are described in detail.

In this study, the proposed evolutionary algorithm (EA) follows GA's basic principles, which were first introduced by Holland (Holland 1975 in [2]). GA is a population-based metaheuristic which mimics the genetic processes of individuals. GA is also classified as a stochastic search algorithm since the algorithm's structure highly depends on the randomization and probabilistic parameters. In the basic GA, the population of individuals competes with each other in each generation to survive, and as Charles Darwin indicated that the fittest ones could survive to bring more offsprings.

To represent the real nature, GA uses a population of individuals; each individual represents a complete solution to the problem at hand. Each individual has a fitness score, which indicates the desired performance of the solution. As first comes to mind, the fitness score may be the objective function, but this is not always the case. Considering the nature of the problem or the constructed algorithm, the fitness function can be constructed to measure the required performance metric or combination [2]. Then fitted individuals can reproduce to create new offsprings that have taken some biological information from each parent. In reproduction, the goal is to pass on the right features to the next generation to evolve solutions.

Although there exist (meta) heuristic approaches that perform well in the literature [6, 35] GA can be considered one of the most robust metaheuristic techniques that can deal with large-sized instances within a reasonable time for even NP-Hard problems. According to [2], by hybridizing GA with other methods, improvements can be achieved even where the existing solution methods perform well.

Considering the nature of the 2E-LRPTW into account, the constructed EA deals with three problems in each operator: the facility opening problem, customer allocation, and routing decisions. In this section, the evolutionary algorithm is used to find a complete solution to the second echelon, given the optimal CDCs and considering direct transportation among CDC-satellite pairs.

Initial solutions are generated both randomly and using well-known construction heuristics to increase diversity and prevent slow-finishing. In these techniques, only the priority of customers in the allocation phase is differing. However, the route construction is done using the Push-Forward Insertion heuristic proposed by Solomon [36] in each of them. Each generated individual consists of a complete solution, and while calculating the fitness score, the corresponding first echelon cost is added. After creating the initial population, a mating pool is constructed using a binary tournament selection to select individuals participating in the reproduction phase. Individuals in the mating pool can reproduce and generate two offsprings or be directly copied as new offsprings for the next generation. Rather than using classical mutation operators, GA is hybridized with several local search techniques to exploit the solutions that have been obtained after the reproduction phase. Before going into the next generation, all of the individuals sorted in ascending order of their fitness score, and the first ω_1 can construct the next generation. This elitist approach enables to keep population size constant and expects to obtain good solutions from fitted individuals.

In this chapter proposed EA is designed to solve the second stage; therefore, the optimal CDCs at the first echelon is considered as given in the preliminary studies. At this stage, as location decisions, there is a decision to open and close satellites only. However, in both echelons, allocation and routing problems exist. In the following sub-sections, all the EA operators' details and used heuristics are explained. The reader can follow the main steps of solution algorithm from Figure 4.1.

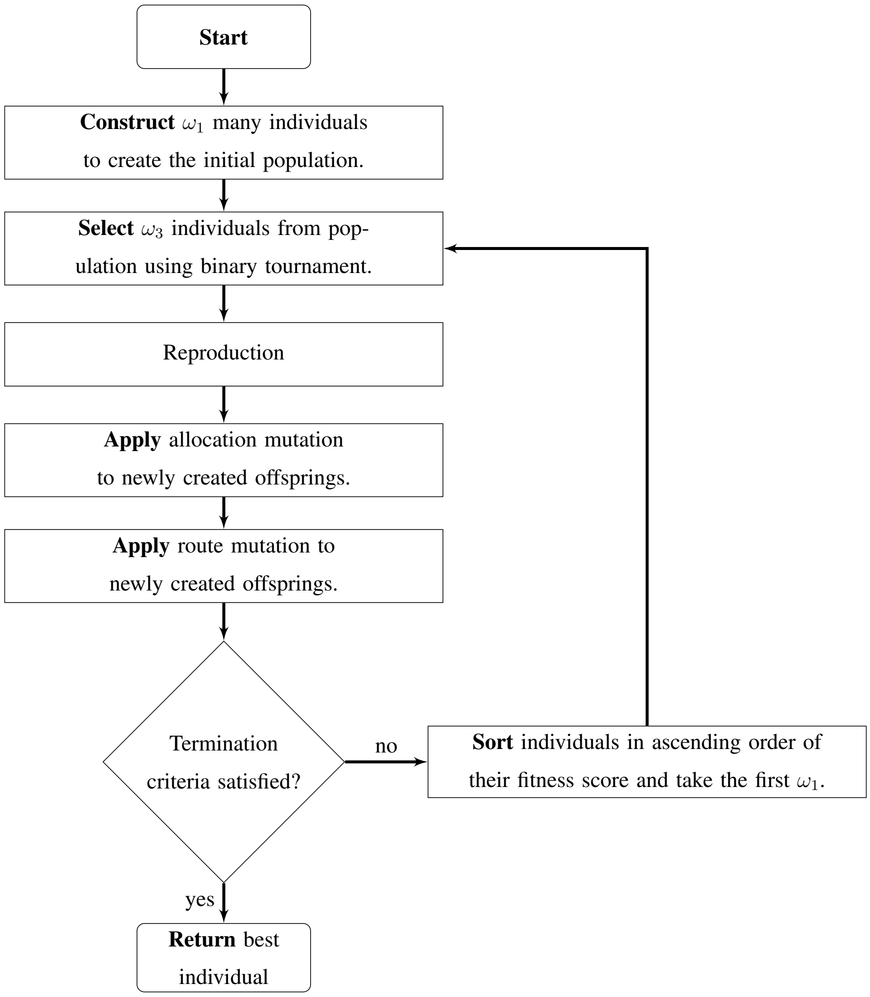

Figure 4.1: The main steps of the proposed evolutionary algorithm

4.1 Solution Representation

A solution to the problem includes many defining components: opened facilities at both echelons, allocation information of customers and satellites, and constructed routes of each satellite. However, in the classical GA approach, chromosomes are enough to hold the necessary information, and a set of chromosomes create an individual, which is also called a complete solution.

Remember from Chapter 2 that an LRP can be reduced to an MDVRP after determining which satellites to open if decisions are taken sequentially. Therefore, the related literature of MDVRP solved by GAs is informative about the approaches of our study. The survey of Karakatic and Podgoreleck [19] demonstrated the variations of genotype representations used in GA to solve the MDVRP. According to their study, classical approaches in routing problems without time windows, some of the studies tried to construct the individual with one chromosome filled with index numbers of the customers by including the facility indexes in an array [20, 26]. However, this approach is debatable since it is not clear that it is solved the operator handling problems throughout the algorithm. A straightforward approach is used in studies rather than representing the solution using one array, using multiple arrays (chromosomes), representing a corresponding facility tour [4, 31]. The crossover operation is a combination of routes, but this representation may visit some customers more than once.

Considering all the related representations and their drawbacks, similar chromosome representation is inspired by the literature, but the complete individual cannot be represented using a set of route arrays. Additional arrays are added to prevent infeasible solutions. Because in our algorithm, infeasible solutions are not accepted at the end of an operator. We aim to create feasible solutions throughout the EA since our problem has hard time windows and capacity restrictions in each echelon.

4.1.1 Chromosome

Chromosomes represent only feasible routes. A feasible route consists of stops, indicates no violation of time windows of facilities and customers, and vehicles' capacity. Chromosomes or namely routes indexed by r, and each route originated from a satellite j denoted by j_r.

1	7	11	20	12	8	9	**1**
1	10	13	17	18	**1**		

3	23	21	24	**3**	
3	14	16	19	15	**3**

Figure 4.2: Chromosome array

The representation of a chromosome can be seen in Figure 4.2. Each chromosome starts and ends with an open satellite j. Nodes between the satellites represent the visited customers in the given route. For each route, the travel time, total waiting time, and route cost is kept. Route cost is represented by c_{jr} for each route r originated from satellite j. Each of the routes can only be performed by one related vehicle, and a customer can be precisely in one tour. Vehicle load and time window constraints are always checked whenever a new route is constructed.

4.1.2 Individual

An individual represents a complete solution to the given problem, including all the necessary information about the open facilities, assignments, and resulting routes. One can say that an individual consists of a vector of chromosomes as well as the following information.

1. Each individual has a facility array with the length of open CDC and candidate satellites to demonstrate which facilities are open. Indexes start from 0 to $m-1$ and represent the corresponding facility, and 0/1 in the arrays indicates whether the facility open or not. Index zero represents the open CDC.

1	**1**	**0**	**1**	**0**
0	1	2	3	4

Figure 4.3: Facility array

2. Each individual has an allocation array that holds customer assignments. Each index demonstrates the corresponding customer starting from m to l, and each value in the cell is satellites to which they are assigned. For instance, Figure 4.4 shows that customer 8 is assigned to satellite 3.

Satellite	1	3	3	1	1	3	1	3	1	3	3	3	1
Index	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**

Figure 4.4: Allocation array

4.2 Initial Solutions

One of the essential elements to start the EA algorithm is to create initial solutions as large as the population size. These solutions may increase diversity on the way to the best solution or result in hitting a solution far from the optimal solution. In the literature, many techniques are followed to create initial solutions. The most common of these are to create the whole population randomly or some portion randomly while producing the remaining portion with heuristics that give good results and giving them to the population as seed.

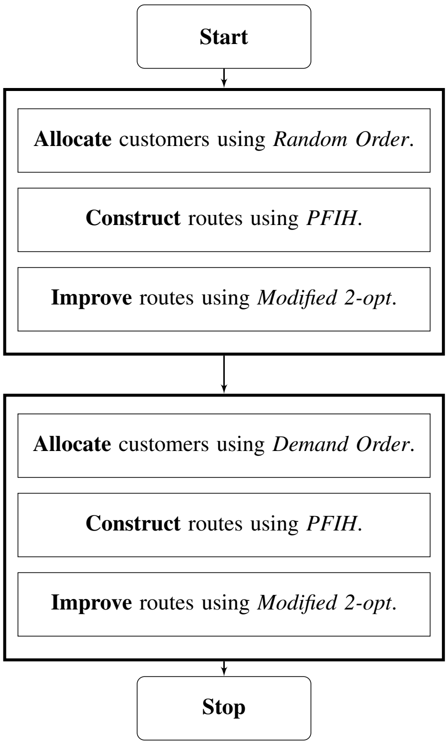

Figure 4.5: Initialization procedure

Due to the complexity of our problem, some solutions were randomly produced, while the rest were created with well-known construction heuristics. In a complex problem, results are generated to increase the variety, but seed solutions are added to sustain evolution throughout EA. The goal is to increase diversity and prevent slow finishing, which is a pervasive problem in EAs.

When generating a solution, keep in mind that there are three stages in this problem.

The creation process consists of opening satellites, assigning customers to the opened satellites considering time windows and facility capacities, and creating routes considering vehicle capacities and time windows. A hierarchical order is followed during the solution generation, or we can say that we tried to solve the problem by decomposing it into three stages.The complete initial solution generation algorithm can be followed by Figure 4.5.

4.2.1 Opening Satellites

The generation process always starts with the phase in which satellites are to be opened. In both creation techniques, namely *Create Random Order* and *Create Demand Order*, satellites opened randomly. Using a heuristic method may end up with a fixed satellite configuration, which is considered an undesired situation in a population-based metaheuristic. For this reason, satellites are opened randomly to represent a variety of satellite combinations in the solutions. The most crucial point here is that if the correct satellites are not opened, allocations and routes will be far from optimal. In other words, it is vital to generate solutions having open satellites in the optimal solution to reach correct allocations.

While making location decisions, another issue that needs to be decided is how many satellites should be opened to satisfy feasibility. As there are those in the literature who simultaneously [5], [35] solved this problem, it is possible to decide on the number and then which satellites to open.

In this evolutionary algorithm's initialization phase, the number of satellites to be opened, N_s, is also decided randomly. Although the aim is to obtain minimum cost by opening a minimum number of facilities on both echelons, the number can be higher than the required for exploration purposes. To provide feasibility, we calculated a lower bound (LB_s) to the number of satellites to satisfy the total customer demand. LB_s can be calculated by dividing the total customer demand by average satellite capacity and rounding up the obtained value. So, the number of satellites to be opened is a random number between the lower bound and the candidate satellites in the corresponding instance.

$$LB_s = \left\lceil \frac{\sum_k D_k}{\frac{\sum_{m \in J} Q_m}{|\mathcal{J}|}} \right\rceil \tag{4.1}$$

$$LB_s \leq N_s \leq |\mathcal{J}| \tag{4.2}$$

After obtaining the corresponding value, N_s many satellites opened randomly. The second echelon problem is reduced to MDVRPTW, which indicates that we need to deal with allocation and routing problems.

4.2.2 Allocation Phase

The two initial solution generation techniques differ from the allocation phases only. One of the essential stages in the algorithm is how customers are assigned to facilities because the primary source of diversity comes from the allocation phase. The allocation phase aims to create different allocation combinations even for the solutions having the same open satellites. Therefore, ω_2 many solutions are generated using *Create Random Order* while the remaining solutions generated by *Create Demand Order*.

- *Random Order:* Customers are selected randomly and assigned to the nearest satellite with sufficient available capacity and considering the distances between customers and the satellites to satisfy both parties' time window constraints.
- *Demand Order:* This allocation method is inspired by [5] and [35]. The method sorts customer demands in descending order, and starting from the customer having the highest demand, assigns the customer to the nearest satellite, preventing constraint violation. This method prioritizes customers with demands to efficiently use satellite capacities.

The reader may refer to the Algorithms 1 and 2 for the details of the allocation techniques. In order to be consistent while creating pseudo-codes, joint sets, parameters and indexes are referenced in the same way. The definition of sets that will facilitate tracking is as follows.

Algorithm 1: Random Order

Result: Allocations of an individual

Calculate LB_s and determine N_s randomly.

Shuffle customer orders.

for $k \leftarrow 1 : \ \forall k \in \phi_1$ ***to*** $|\mathcal{K}|$ **do**

  while $k \notin \theta_j$ **do**

    for $j \leftarrow 1 : \ \forall j \in \tau$ ***to*** N_s **do**

      Search for k the nearest open j having sufficient capacity.

      Add customer k to the allocation set of θ_j.

    if $k \notin \theta_j$ ***and*** $j = N_s$ ***and*** $N_s \leq |\mathcal{J}|$ **then**

      $N_s \leftarrow N_s + 1$

    else

      Initialize θ_j and N_s.

      $k \leftarrow 1$

      $j \leftarrow 1$

      break

The set θ_j contains the customers which allocated the satellite j while set τ holds random order of the candidate satellite locations. ϕ_1 and ϕ_2 represent the sets for the customers' random order and the customers according to their descending order of demands, respectively.

Let LB_s indicates the minimum number of satellites that needs to be open and N_s represents the number of satellites opened randomly. $Size_j^r$ be the length of route r originated from satellite j. r_1^{Best} introduce the minimum cost route r_1 obtained so far and r_1^{New} indicates that route r_1 is updated. C_{N1} is the routing cost of new r_1 and C^{Best} represents the minimum total routing cost at hand. Ind^{Best} demonstrates the individual having the minimum fitness score and Ind^{Orig} is the original individual we start the algorithm initially.

In *Random Order*, starting from the first customer from the shuffled order, we search the nearest open available satellite and then assign the customer to the corresponding satellite. If none of the open satellites are feasible and there can be another close satellite to be open, open it and try to assign the customer. However, none of the satellites are available for that particular customer, shuffle customer orders and initialize the parameters and start from the beginning. The difference is in *Demand Order*, we start from the customer having the highest demand for allocation and follow the demand order rather than randomization. In this algorithm, the condition to start over is not encountered since the capacities of the satellites are used efficiently.

Algorithm 2: Demand Order

Result: Allocations of an individual

Calculate LB_s and determine N_s randomly.

for $k \leftarrow 1 : \forall k \in \phi_2$ ***to*** $|\mathcal{K}|$ **do**

  while $k \notin \theta_j$ **do**

    for $j \leftarrow 1 :\ \forall k \in \tau$ ***to*** N_s **do**

      Search for k the nearest open j having sufficient capacity.

      Add customer k to the allocation set of θ_j.

    if $k \notin \theta_j$ ***and*** $j = N_s$ ***and*** $N_s \leq |\mathcal{J}|$ **then**

      $N_s \leftarrow N_s + 1$

4.2.3 Routing Phase

In the last phase of the initial solution construction, there exists a routing phase. This phase is applied after both allocation techniques. Our goal is to construct a set of minimum-cost vehicle routes that start and ends at an originating facility, satisfying customer demands and time windows.

- *Push Forward Insertion Heuristic:* Routes are generated using a fast and straightforward construction heuristic named Push Forward Insertion Heuristic (PFIH) proposed by Solomon [36]. This sequential route construction heuristic is effective and efficient in incorporating the time window constraints in the solution process. The first selected customer, seed, has the minimum latest time window B_k. The sequential customers are chosen one by one, considering insertion costs to minimize the total distance and time. The subsequent customers are located by investigating all feasible customer positioning. A new route is created if only if no more customer feasible insertions are possible.

- *Modified 2-opt:* Whenever a new route is constructed, the 2-opt heuristic which introduced by [11] is applied with modification. The nature of classical exchange heuristics like 2-opt is not desirable for the problems having time windows since they do not keep the orientation. On the contrary, the modified 2-opt takes customer orientation into account and selects the route with the highest waiting time between the two, having the same route cost. We chose the route with more waiting times to obtain a route suitable for modification in the future improvement stages. This heuristic provides inter route improvement until we hit a locally optimal solution.

4.3 Fitness Function

EA needs a performance measure to compare individuals in the current pool to sustain evolution for the following generations. Depending on the nature of the problems, a variety of fitness score definitions can be constituted. In our problem, the fitness function calculates the total cost of 2E-LRPTW. Currently, the algorithm only finds the second echelon solution and the first echelon solution assumed to be given in this stage. In order to sustain integrity, we considered not only the total cost of the second stage but also the total cost of the original problem as a fitness score. For each individual, the corresponding fitness score is represented by π.

Before calculating the fitness score, the algorithm has already been decided the allocated demand to each open satellite and the resulting trips from these allocations with corresponding route costs. When the CDCs are taken as given, the opening cost in the objective function and the total cost of the trips have become parameters. We are reducing the problem size and work with a slightly straightforward model. Using the allocated demands to satellites, direct trip costs from CDC to open satellites are calculated by finding the number of vehicles needed. We have facility opening costs and routing costs for both echelons, consisting of vehicle operating costs and the cost of trips. The fitness score is then calculated as the sum of these costs from these two echelons, which result in comparable solutions with [15].

4.4 Reproduction Phase

In GAs, the generation stage of new offsprings is named with the crossover operator because it is possible to obtain two new offsprings after the classical crossover process. Considering the complexity of our problem, it is not possible to obtain feasible offspring with classical crossover methods in the literature. Therefore, in this study, the crossover operator is used as a step of the reproduction procedure to increase the diversity of satellite configuration.

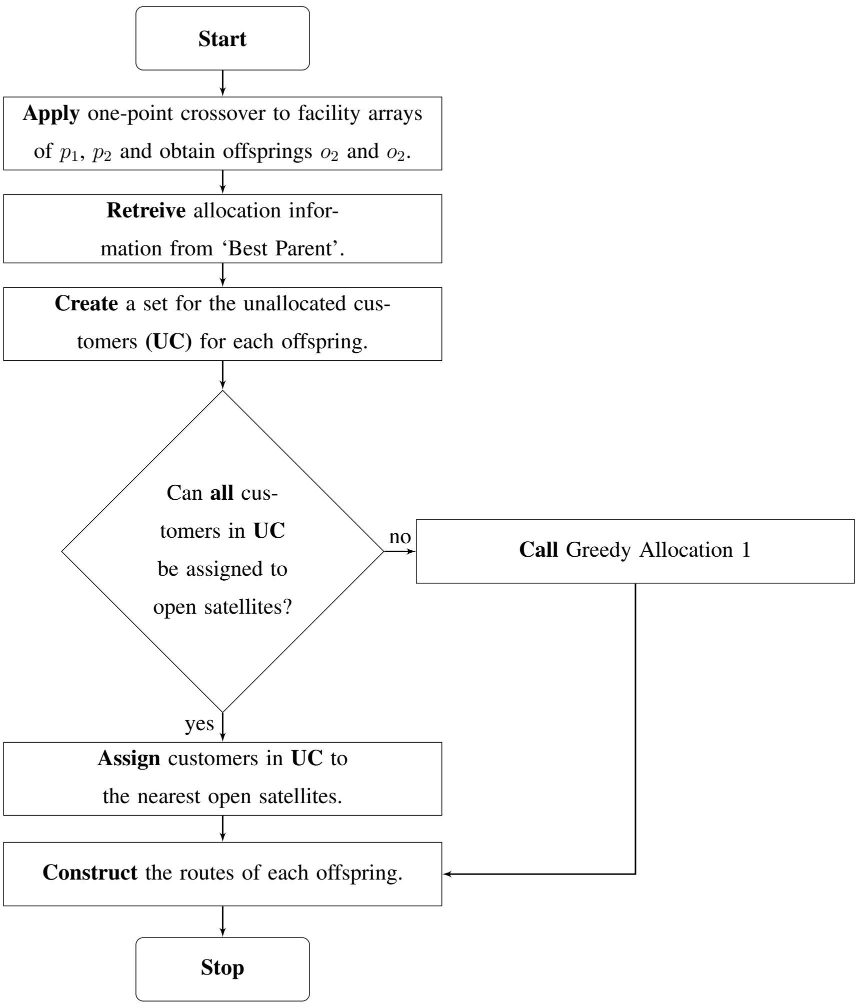

Figure 4.6: The main steps of the reproduction phase

The reproduction phase consists of three steps: a one-point crossover, allocation, and routing steps. In this phase we aim to achieve evolution by creating better individuals from two parents' genetic heritage who come together randomly from the mating pool. In this study, the reproduction phase determines individuals' recombination to produce new individuals called offsprings, $o \in O$. In this phase, parents $p \in P$ are selected randomly from a mating pool to reproduce. If reproduction is not applied, parents are duplicated as new offsprings. Unlike classical crossover approaches in the literature, this study combines classical one-point crossover with new techniques to generate a feasible offspring. Because well-known crossover operators for TSPs do not keep track of feasibility through each step. The authors who used these operators either accept an infeasible solution or try to repair them after crossover. This algorithm aims to produce feasible solutions at each step without using a repair operator because hard time windows increase complexity. As a result, the classical operators need to be adapted to the nature of the problem at hand.

4.4.1 One-point Crossover

One-point crossover is applied to the facility arrays of selected two parents to obtain two new facility arrays of offsprings. At this stage, the only thing expected is to create a new satellite configuration using both parents' information.

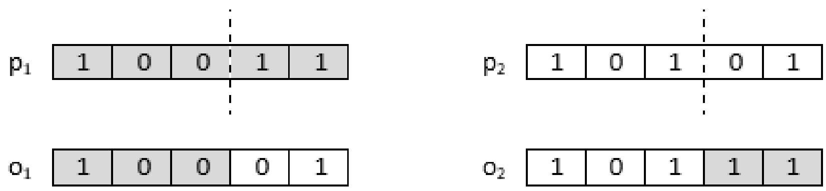

Figure 4.7: One-point Crossover

This stage is sufficiently straightforward. It cuts the parents' array from a random position to get two 'head' and two 'tail' segments. The tails are then swapped to obtain a full-length facility array, indicating which satellites are open for each offspring. The only important point here is that although the facility array retains CDC information, the cut point is selected randomly from a point after CDC end.

4.4.2 Allocation Phase for Offsprings

After the one-point crossover of facility arrays, we know which satellites are open for each offspring, and we can track satellite information that comes from which parent. Using the information in each parent's allocation array, customers assigned to open satellites are also known. If we try to copy open satellites' assignment information directly from both parents simultaneously, we will probably obtain an offspring having unassigned customers or customers assigned more than one satellite. To not repair such solutions, we must do the assignment step by step to get information from parents. After all, one of our aims in reproduction is to transfer the good information to the next generation.

The allocation stage consists of three parts. In the first stage, as we receive satellite information from the parents, we check which parent's allocation information along with the corresponding satellite information portion should be copied directly as in classical approaches. It is then checked whether remaining customers in the set **UC** can be allocated to the available satellites. If all **UC** customers cannot be assigned, we search for well-constructed allocations using *Greedy Allocation 1* and *Greedy Allocation 2* algorithms.

It has been already mentioned that retrieving allocation information from both parents directly probably ends up getting allocation duplication. Therefore, we need to perform the retrieving process sequentially. At the beginning, rather than deciding whether to get information directly from the 'head' or 'tail' parent, retrieve allocations of 'Best' parent for the corresponding satellites. Since we are trying to minimize the total cost using the algorithm, only the 'Best' parent's allocation information is retrieved at first considering the performance measure, $AvgDist_p$.

In other words, we know that each offspring has a 'head' and a 'tail' parent. In the one-point crossover stage, we separate each parent's facility arrays from the cut off point we randomly obtained. The left part from cut-off point p_1 and the right part from p_2 merge, and a facility array is formed for the first offspring. Accordingly, p_1 is the 'head' parent and p_2 'tail' parent as it can be seen from Figure 4.7. By this means, the transferred satellite information is available with the allocations along

Algorithm 4: Select Best Parent

Input: p_1,p_2, cut-off point

Result: Best parent

Initialize C^{p_1}, C^{p_2}, N_c^1 and N_c^2.

for *first parent's open satellites before cut-off point* **do**

  Sum route costs and vehicle usage costs, $Cost^{p_1}$.

  Sum all of the visited customers in these routes, N_c^1.

$AvgDist_1 = C^{p_1} \setminus N_c^1$.

for *second parent's open satellites after cut-off point* **do**

  Sum route costs and vehicle usage costs, C^{p_2}.

  Sum all of the visited customers in these routes, N_c^2.

$AvgDist_2 = C^{p_2} \setminus N_c^2$

Return the parent having minimum cost as 'Best'.

with the constructed routes. To understand which parent we can get direct information from, we compared the 'head' and 'tail' parents for each child. The one with the lowest cost per customer is selected as the 'Best' parent, and allocations of the 'Best' parent are directly copied to the corresponding offspring. Emphasizing that not all of the 'Best' parent allocation information is copied; only the transferred satellites' allocation information to offspring are given priority.

It should be noticed that the 'Best' parent can be different for each child because they take different information regions of the parents. This performance criterion considers the allocation of information coming from each parent one by one, for each parent, summation of costs of the routes owned by each parent's open satellites divided by the total number of customers visited.

The remaining customers are copied to a set called unallocated customers **UC** and proceed into the steps in the following order.

- *Check:* Applying control using this function, we aim to eliminate unnecessary steps to assign remaining customers to the open satellites. This function checks if all of the customers in set **UC** can be assigned to the nearest available satellites. If all of them can be assigned, assignments are completed, and

UC becomes empty, which ends the allocation phase. Otherwise, leave **UC** as original and go to *Greedy Allocation 1*.

Algorithm 5: Greedy Allocation 1

Result: Allocations of an offspring

for $k \leftarrow 1 : \ \forall k \in \omega_2$ **do**

  Let $F^{Best} = -1$ be the closest satellite to customer j in **UC**.

  Let $C^{Best} = \infty$

  for *all open satellite j* **do**

    if *satellite j is available **and** dist(j,k)* $\leq$*C*^{Best} **then**

      F^{Best} = j

      C^{Best} = dist(j,k)

    if $F^{Best} \neq -1$ **then**

      Add customer k to θ_j.

      Delete customer k from **UC**.

if ***UC*** $\neq \emptyset$ **then**

  Call *Greedy Allocation 2*

- *Greedy Allocation 1:* This algorithm applies *Demand Order* for allocation as in the initialization phase. The **UC** set's customers are sorted according to demand and assigned to the nearest satellite having sufficient capacity. If **UC** is not empty after allocations, proceed to *Greedy Allocation 2*; otherwise, stop.

- *Greedy Allocation 2:* If allocation processes did not complete in *Greedy Allocation 1*, this implies whether open satellites have no available capacity for the remaining customers or assignment cannot be done because of the hard time window constraints. In this case, a new satellite should be opened, but the problem arises of which one to open. For each closed satellite, the total distance of customers in the **UC** set is kept. The satellite having the minimum total distance to the unallocated customers is opened. This process continues until all customers in **UC** is assigned to an open satellite.

Algorithm 6: Greedy Allocation 2

Result: Allocations of an offspring

for *all close satellite j* **do**

  Sum the direct trips between satellite j and customers in **UC**.

Sort the total cost of trips in a descending order, **TC**.

while $\boldsymbol{UC} \neq \emptyset$ **do**

  for *each close satellite* $j \in$ ***TC*** **do**

    Open the satellite j and assign customers until the capacity is full.

    Delete allocated customers from **UC**.

4.4.3 Routing Phase for Offsprings

After the assignments are complete, the last step is to generate routes for each satellite. Since we get the allocation information directly from the 'Best' parent, we can directly copy its routes to the offspring. The 'Best' parent selection has already taken into account the routes to transfer good information to the next generations.

After copying the 'Best' parent routes, we navigate the customers who do not route using the *Create Route* as in the initialization stage with a little modification. Originally, seed customers are selected according to the earliest time window, but seed customers were randomly selected to increase diversity in this step.

4.5 Mutation Operators

An offspring obtained through the reproduction then proceed into the mutation phase. Also, an offspring can remain in the population without mutation according to the determined probability. As mutation operators, a set of improvement techniques have been applied to enhance the current solution. In the literature, there exist many algorithms combining classical improvement techniques with each other. The algorithms that are generally created for the problems in this structure consider two points: one is the routes, and the other is the allocation changes.

Solutions to our problem can be improved by providing a variety of route and allocation moves. While a move used to improve routes does not affect the satellite's allocated capacity, it is very likely that a move created for allocation will affect the total satellite demand, thus changing the first echelon solution. At this stage, we try to balance the density between exploration and exploitation. By changing the allocations, we aim to obtain a much different solution from the current solution we have, and we are trying to rearrange our routes to a less costly point with route improvements.

For this purpose, two mutation algorithms have been proposed. The proposed neighborhood structures in the EA are examined in two parts, namely allocation and route improvements. The first one selects customers belonging to different satellites for relocation; in other words, it takes a customer from a route and inserts a route belonging to a different satellite. In comparison, the second one tries to improve routes belonging to the same satellite by changing the customer locations across routes. The flow charts of these algorithms can be followed by Figure 4.8 and Figure 4.9 respectively. The detailed explanations will be in the following subsections along with the the pseudo-codes.

4.5.1 Improve Allocations

Allocation information in the solution is changed by taking a customer from a route and inserting it into another satellite route. Using this neighborhood structure, the demand assigned to each satellite is changed, so does the first echelon solution. The primary point in the problem can be considered as the allocation change. Therefore, we expect the new solutions obtained from this operator to be different from the incumbent solution. Existing operators in the literature make customer swaps between satellites or switch satellites on and off to modify allocation structure. Because the capacity constraints are relaxed in the algorithms, infeasible solutions progress over generations either with a penalty or repair.

In the experiments conducted, we have seen that it is impossible to perform classical inter-satellites swap operations with tight capacity facilities. In this case, even if two swapped customers provide feasibility, it is impossible to achieve either improvement or feasibility due to hard time windows. Since there is no hard time window structure in the existing studies, they can obtain different solutions by relaxing capacity constraints. However, since we are working not to create infeasible solutions and not allow such solutions, we modify the existing swap mutations according to the problem structure.

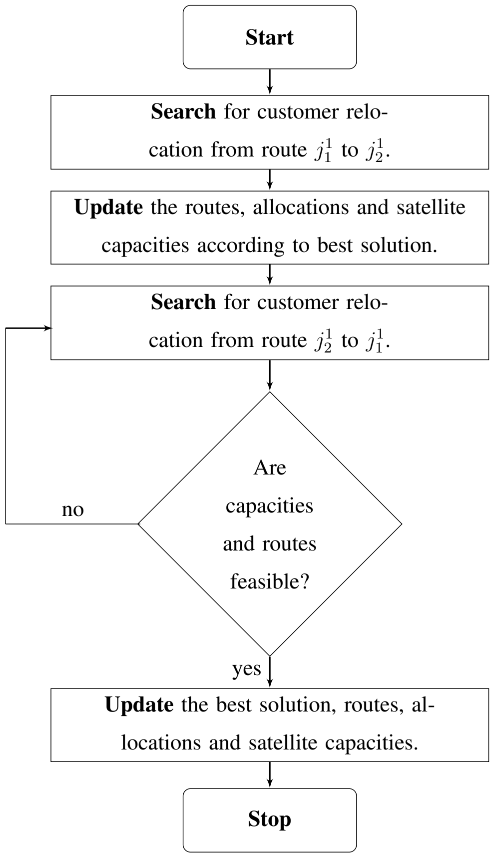

Figure 4.8: The main steps of Improve Allocation

In order for a solution to be feasible, a satellite capacity cannot be exceeded. However, sets having tight satellite capacities do not leave room for relocating a customer to a different satellite route. The *Improve Allocation* allows keeping an infeasible solution in the first stage.

The critical point here is that we only do this at one stage. At the end of this algorithm, it is not possible to allow or repair an infeasible solution. Since the operation is performed between two routes belonging to different satellites, let us say j_1^1 and j_2^1, only the routes' feasibility is checked when adding the customer from j_1^1 to j_2^1 in the first iteration. The incumbent solution is updated with the best feasible route found with the satellite capacities, but satellite capacities' feasibility is not checked.

Algorithm 7: Improve Allocation

Input: Offspring

Output: Mutated Offspring

Initialize Ind^{Orig} and Ind^{Best} with the current offspring.

for *each* r_1, r_2 *pair* ***belonging different*** *satellites* **do**

  Initialize r_1^{Best}, r_2^{Best}, and C^{Best}.

  for $p_1 \leftarrow 1 : p_1 \leq Size_j^{r1} - 1$ **do**

    for $p_2 \leftarrow 1 : p_2 \leq Size_j^{r2}$ **do**

      Erase customer from position r_1 and **insert** position in r_2.

      if *new routes are feasible and better than the current ones* **then**

        Update C^{Best} = C_{N1} + C_{N2}

        Update r_1^{Best} = r_1^{New}

        Update r_2^{Best} = r_2^{New}

  Initialize r_1^{Best}, r_2^{Best}, and C^{Best}.

  for $p_2 \leftarrow 1 : p_2 \leq Size_j^{r2} - 1$ **do**

    for $p_1 \leftarrow 1 : p_1 \leq Size_j^{r1}$ **do**

      Erase customer from position r_2 and **insert** position in r_1.

      if *satellite* ***capacities are feasible*** **then**

        if *new routes are feasible and better than the current ones* **then**

          Update C^{Best} = C_{N1} + C_{N2}

          Update r_1^{Best} = r_1^{New}

          Update r_2^{Best} = r_2^{New}

          $Node^{Best}$ = customer

    if $Node^{Best} \neq -1$ **then**

      Return r_1^{Best} and r_2^{Best}

      Update the allocations of the offspring.

      Update the satellite capacities of the offspring.

Update first echelon solution of Ind^{Best}.

Calculate fitness score of Ind^{Best}.

if Ind^{Best} *better than* Ind^{Orig} **then**

  Return Ind^{Best}

else

  Return Ind^{Orig}

In the second stage, customer relocation from new j_2^1 to j_1^1 is searched. The expectation here is that the relocation from j_2^1 to j_1^1 should balance the capacities resulting in relocation from the first stage. While searching, the best solution is updated if and only if the generated solution is feasible. This move is different from exchanging two customers belonging to different satellite routes; this is an iterative approach. After finding the best feasible route resulting from the second relocation, the incumbent is updated and accepted if feasible otherwise, original routes are returned.

Improve Allocation is mainly targeting the instances having tight facility capacities. Although it is possible to obtain different satellite configurations with various allocation schemes, it is impossible to create diversified solutions when capacities are tight. Therefore, we take two routes of offspring originated from different satellites and try to change the allocations of two customers. In the first best insertion, the satellite capacities are updated but the feasibility is not checked. However, in the second insertion best routes are only updated if the resulting allocation gives feasible capacities with feasible routes. After completing the changes among all of the routes of different satellites, a new individual is obtained and the resulting first echelon solution is computed with the corresponding fitness score. If the new individual is better, continue with the new one; otherwise, return the original individual.

4.5.2 Improve Route

Improve Route algorithm consists of five well-known iterative route improvement heuristics. These heuristics are mainly in the class of exchange heuristics but with some modified versions. Since most classical k-exchange heuristics do not preserve the routes' orientation, they are not suitable for the problems with hard time windows. The *Improve Route* takes two routes belonging to the same satellite and performs 2-opt* [30], Exchange-Edges, and Relocate Customer heuristics.

- *2-opt*:* The 2-opt* exchange heuristic is proposed by Potvin [30] to deal with problems having time windows. [30] study demonstrates that 2-opt* perform well to obtain inter-routes improvements. 2-opt* generates new solutions that are very different from the incumbent solution. In the classical 2-opt approach, the sequence is not affected by changing orientation, but when the customers have time windows, reversing some portion of a route is likely to produce an infeasible solution. This heuristic can be considered as a link exchange heuristic between two routes of an individual. The algorithm considers all possible feasible results between two routes and returns only the best solution decreasing

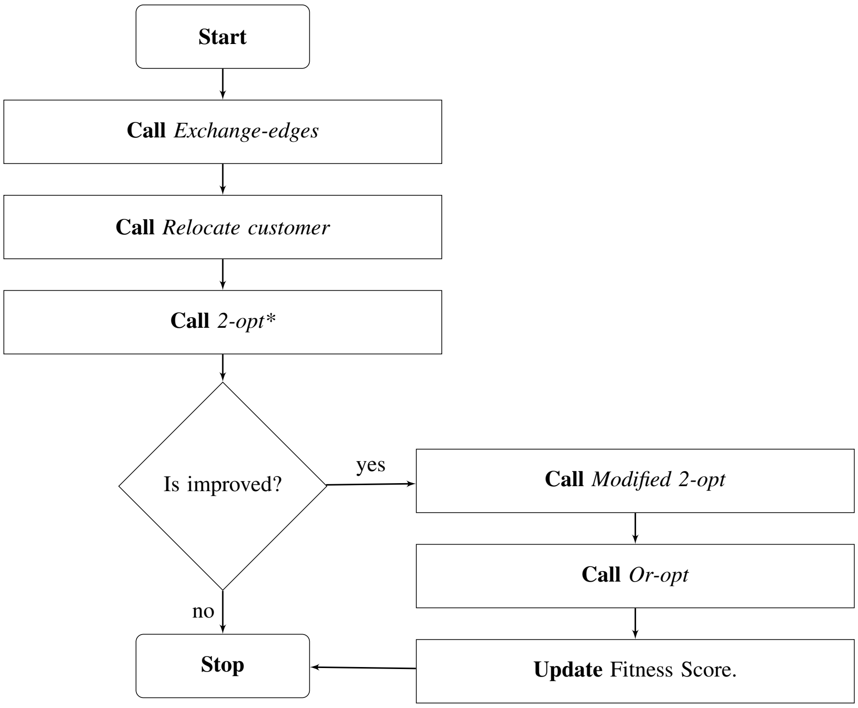

Figure 4.9: The main steps of Improve Route

the total route cost of the incumbent solution.

- *Exchange-edges:* Exchange two variable-length edges between two routes of the same satellite. There exists no selection of satellites or routes; exchange-edges is applied to all generated routes of each open satellite. The algorithm searches all pairs of feasible exchanges among two routes and applies only the best solution obtained.

- *Relocate customer:* One customer is selected from a route and inserted into a new position in another route connected to the same satellite. All customers in the routes and possible positions giving feasible solutions are taking into account, and only the move giving the least cost from the incumbent solution is applied.

If one of the heuristics above succeeds at finding a new best solution, then modified 2-opt and Or-opt algorithms are applied to the corresponding new routes.

- *Or-opt:* Or-opt exchange is a well-known node exchange heuristic proposed by Or (Or 1976 in [30]). This heuristic tries to improve the incumbent solution

Algorithm 8: Exchange-edges

Input: r_1,r_2

Output: r_1^{Best},r_2^{Best}

Take routes r_1, r_2 and initialize C^{Best} as their total cost.

for $n_1 \leftarrow 1 : n_1 \leq Size_j^{r1} - 1$ **do**
  for $p_1 \leftarrow 1 : p_1 \leq Size_j^{r1} - n_1$ **do**
    for $n_2 \leftarrow 1 : n_2 \leq Size_j^{r2} - 1$ **do**
      for $p_2 \leftarrow 1 : p_2 \leq Size_j^{r2} - n_2$ **do**
        edge1 = (n_1, n_1 + p_1)
        edge2 = (n_2, n_2 + p_2)
        Exchange edge1 and edge2 between r_1 and r_2.
        if *new routes are feasible and better than the current ones* **then**
          Update C^{Best} = C_{N1} + C_{N2}
          Update r_1^{Best} = r_1^{New}
          Update r_2^{Best} = r_2^{New}

Return r_1^{Best} and r_2^{Best}

by changing the customer's position in the current tour. In Or-opt, not only one customer is a candidate; also, the sequences of more than one customer is taking into account. This algorithm slightly modifies the incumbent solutions with finer refinements.

After explaining all the operators used in the algorithm in detail, we can follow the main stages of the Algorithm 10. α_{max} indicates the maximum number of generations this algorithm can proceed, and Δ represents the convergence criteria value.

Initialize the parameters and construct the initial population as described in Section 4.2 before proceeding into the main loop. In the main loop, select ω_3 many individuals from the initial population using Binary Tournament. Binary Tournament selection randomly selects two individuals and compares their fitness scores. The fittest individual is added to the mating pool set (**MP**) and the other one returns to the

Algorithm 9: Relocate customer

Input: r_1,r_2

Output: r_1^{Best},r_2^{Best}

Take routes r_1, r_2 and initialize r_1^{Best},r_2^{Best}, C^{Best} as their total cost.

for $p_1 \leftarrow 1 : p_1 \leq Size_j^{r1} - 1$ **do**

  for $p_2 \leftarrow 1 : p_2 \leq Size_j^{r2}$ **do**

    Erase customer from position r_1 and **insert** position in r_2.

    if *new routes are feasible and better than the current ones* **then**

      Update C^{Best} = C_{N1} + C_{N2}

      Update r_1^{Best} = r_1^{New}

      Update r_2^{Best} = r_2^{New}

for $p_2 \leftarrow 1 : p_2 \leq Size_j^{r2} - 1$ **do**

  for $p_1 \leftarrow 1 : p_1 \leq Size_j^{r1}$ **do**

    Erase customer from position r_2 and **insert** position in r_1.

    if *new routes are feasible and better than the current ones* **then**

      Update C^{Best} = C_{N1} + C_{N2}

      Update r_1^{Best} = r_1^{New}

      Update r_2^{Best} = r_2^{New}

Return r_1^{Best} and r_2^{Best}

initial set. After completing creating a mating pool, selected individuals will proceed into the reproduction phase. If a randomly generated value $random$ between 0-1, is smaller than the predetermined ω_4, selected two parents from **MP** generate two offsprings by recombination described in Section 4.4; otherwise, parents directly copied as offsprings into the set **o**. When each pair of parents in **MP** produced offsprings, the mutation will be applied if $random$ is smaller than the mutation probability ω_5. In this phase, mutated offsprings are only accepted if they are better than the original ones. Otherwise, we keep the original offspring. Then we merge the offsprings with parents and obtain one population set, **P**. Sort all individuals in **P** according to their fitness score and keep the best ω_1 for the next generations initial population. Here, achieve to keep population size constant by using an elitist approach. Then

Algorithm 10: The proposed evolutionary algorithm

Input: Problem parameters

Output: Best Individual

Generate initial population **P**.

Initialize α = 0, $\Delta = \infty$

while $\alpha \leq \alpha_{max}$ ***and*** $\Delta \geq 0.01$ **do**

  Insert ω_3 individual using Binary Tournament to mating pool (**MP**).

  for *each consecutive pair of parents in* ***MP*** **do**

    if $random \leq \omega_4$ **then**

      Call Reproduction (p_1,p_2) and obtain o_1,o_2

    else

      Copy p_1 and p_2 directly offspring set **O**.

  for *each offspring in* ***O*** **do**

    if $random \leq \omega_5$ **then**

      Call *Improve Allocation (o_1,o_2).*

      Call *Improve Route (o_1,o_2).*

  Insert set **O** to set **P**.

  Sort the individuals in **P** in descending order Fitness Score.

  Calculate $\Delta = (Fitness_{Worst} - Fitness_{Best})/Fitness_{Worst}$

  Resize **P** to ω_1.

  α = α + 1

check whether our population is convergence by calculating the distance between the worst customer and the best customer. Update Δ and α. If the best individual deviates smaller than 0.01 percent from the worst individual or the algorithm reached the maximum number of generations, the EA stops. Otherwise, take the ω_1 many individuals as current population and start over.

CHAPTER 5

REVISITING THE EVOLUTIONARY ALGORITHM FOR 2E-LRPTW

This chapter explains how the constructed EA for a single-echelon problem can be adapted to the two-echelon structure step by step. Since the proposed algorithm is constructed with a holistic perspective, we can obtain an algorithm that determines the locations of the facilities at both echelons, allocations and solves the routing problems by adding small modifications. The proposed evolutionary algorithm's pseudo-code is introduced in the previous chapter; the necessary modifications need to be applied to initialization and reproduction phases. There are no structures that need to change in other steps. A few properties should be added to individuals' characteristics to represent the complete solution. Note that there is a routing decision on both echelons since we have a classical 2E-LRPTW structure.

The essential point to be emphasized in this chapter is that the CDC and satellite opening decisions are taken simultaneously. As the reader may remember from the literature studies, the 2E-LRP is generally solved by decomposition approaches due to its complexity. In other words, the second echelon is solved first, then accepted as parameters to solve the first echelon or the other way around. However, the proposed EA decides locations of facilities simultaneously in the reproduction phase by transferring the facility information over the 'Best' parent to the offspring.

5.1 Individual

Since an individual demonstrates a complete solution to the problem, some properties should be modified while solving the two-echelon problem. Chromosomes have the same structure and now represent the routes that originated from each open facility, not only satellites. After the allocation of open satellites to CDCs is completed, routes can be created, and chromosomes represent the resulting first echelon routes. Consequently, by checking each individual's chromosomes, we can obtain which facilities are open and the routes with their costs.

The facility array of an individual is precisely the same. In the former example, only node 0 represents the open CDC, but now we have an array with the length of total candidate CDCs and satellites, and 0/1 in the arrays indicates whether the corresponding facility open or not.

In the previous chapter, the allocation array holds the customer assignments; now, the allocation array holds both customer and satellite assignments. So, the allocation array has the length of the total number of candidate satellites and customers. In the corresponding satellite section, the indexes represent the satellites and the values indicate the CDCs they assigned. Since not all satellites will be opened, the corresponding assignment value is -1 for the close satellites.

Let us consider an example where we have 2 candidate CDCs 3 satellites and 10 customers. Assume that the second CDC and satellites 1 and 3 are open. Realize that since indexes start from 0, CDC 1 indicates the second CDC. From the Figure 5.1, open satellites are assigned to the open CDC and customers are assigned to open satellites.

Allocation	1	-1	1	2	2	4	2	4	4	4	2	4	2
Index	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**

Figure 5.1: Allocation array

5.2 Initialization Phase

Recall that we have two initialization procedures, namely *Create Random Order* and *Create Demand Order* and at the beginning of each construction, there was a satellite opening phase. The satellite opening phase turned into the facility opening phase to obtain a complete solution.

5.2.1 Opening Facilities

The satellite opening phase is identical for both construction techniques. Keep everything the same while opening satellites and add a step for CDC location decisions to convert this phase as a whole facility location decision before proceeding into *Random Order* and *Demand Order*. Since the CDCs are extensive facilities and designed as a long-term investment, the number of CDCs are less in number than satellites in the benchmark sets. Thus, the decision to open which CDC is taken randomly to increase diversity.

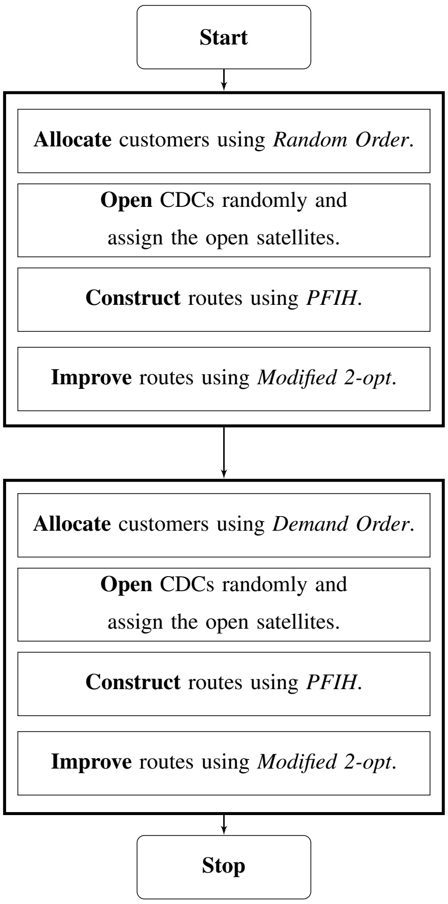

Figure 5.2: Initialization procedure

When opening satellites, we decide on the number first and then the satellites to be opened sequentially. However, the cost of opening a CDC is relatively high, so the number of open CDCs will be determined as a result of the assignments, as [5] and [35] approached in their studies. In other words, when a CDC is opened, assign the open satellites randomly until the CDC capacity is full. If the satellite used capacities are not met, continue by opening a new CDC randomly; otherwise, stop. After the assignments of satellites complete, we can obtain the number and the locations of open CDCs.

The allocation and routing phases in the initialization procedure for customers remain unchanged for the second echelon. The CDC assignments and the cost of routes are not incorporated into these two phases. CDC location decision and assignments of satellites performed after the allocation phase for customers end before the routing phase, followed by the Figure 5.2. There is a sequential approach in the initialization phase; two-echelon decisions are not taken simultaneously.

As a fitness function, we have already added the cost of opening the CDCs and direct trips in the first echelon to the cost we obtained from the second echelon to make our solutions comparable with the original work. As shown in Figure 5.2, the assignments stage is followed by the route construction and improvements phases. In route construction, we used to generate routes only for the second echelon, but now we generate routes for the first echelon. The heuristic works with the corresponding facility's allocation information, and we obtain each route's cost as well. Therefore, there is no difference between constructing a route for the second or first echelon. As a result, the fitness score is the summation of first and second echelon routing, facility opening, and vehicle usage costs.

5.3 Reproduction Phase

The initial population is created in the initialization phase, and then ω_3 many individuals are selected for **MP**. The parents in the MP will proceed into the reproduction phase. The reproduction phase consists of one point crossover, allocation and routing phases. There is no modification in the routing phase since the routing phase constructs routes originated from open facilities.

5.3.1 One-point Crossover

By applying a one-point crossover to parents' facility arrays, we aim to obtain different satellite configurations. The goal did not change while solving two-echelon problems because the number of CDCs is relatively low compared to the satellites, and it is possible to obtain different CDC configurations in the initialization phase by randomly choosing the locations.

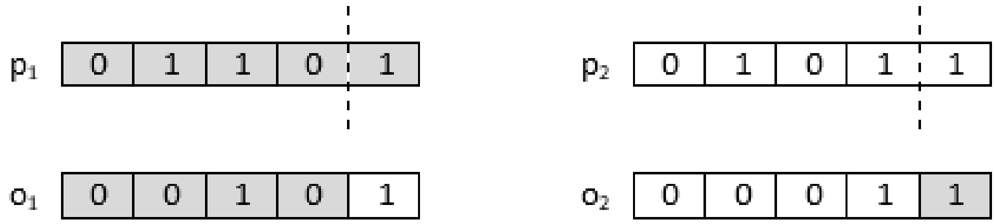

Figure 5.3: One-point crossover

The classical one-point crossover is applied to facility arrays and obtain a full-length facility array for each offspring indicating which satellites are open as in Figure 5.3 for an instance having 2 candidate CDC and 3 satellite locations. The cut-off point is again decided randomly from a point after satellite indexes start. We are not considering the entire facility array because we might have cut points useful for CDC configuration changes but stuck with the same open satellites. Therefore, we only change open/close satellites by applying one-point crossover and CDCs transferred as closed to offsprings. The open CDCs are determined in the allocation phase when the 'Best' parent is procured.

5.3.2 Allocation Phase for Offsprings

In the first stage of the allocation phase, the 'Best' parent's allocation information is favored for each offspring. For the details and reasoning behind it, please refer to Chapter 4. The CDC location decision follows the same approach. In other words, CDCs assigned to satellites in the transferred section are transferred to offspring by the 'Best' parent. However, realize that open satellite information comes from both parents. After taking satellite allocation information from the 'Best' parent, the remaining open satellites are also assigned to the open CDCs, and the used capacity of these facilities should be updated along with the allocation information of satellites.

In the second echelon, the unallocated customers after the 'Best' parent stage are copied to **UC** and proceed into the steps *Check*, *Greedy Allocation 1* and *Greedy Allocation 2*. The problem does not necessarily go into *Greedy Allocation 2* because if it goes, we can conclude that there exist customers in **UC** after *Greedy Allocation 1* and the existing satellites cannot satisfy the allocation constraints. For the two-echelon structure, after completing allocations of the customers, the CDC used capacities should be updated. Since in *Greedy Allocation 1* we deal with the open satellites, there is no need for an allocation step for satellites. Again update the CDC used capacities when the satellite used capacities are changed. Unfortunately, if we need to open a new satellite in the *Greedy Allocation 2*, we first try to assign the new satellite to existing CDCs; if we cannot assign the satellite because of the capacity restrictions, open a new CDC randomly.

In the reproduction stage, the routing phase, *Create Route*, creates routes using the customer allocation information to the satellites and taking the hard time windows into account. Also, the routing phase creates routes using satellite allocation information to the CDCs and primary vehicle capacities. The mutation algorithms are not changed because of shifting the two-echelon structure. There are no mutation algorithms for targeting CDC changes in the modified version. The main structure of the Algorithm 10 will be followed after adding the alterations.

CHAPTER 6

COMPUTATIONAL STUDY

In this chapter, we provide an extensive study of the numerical results and our inferences. We apply the proposed EA to two types of 2E-LRPTW instances, namely Set 1 and Set 2, generated in the doctoral thesis of Farham [15]. The details and the structure of the used test instances are introduced in Section 6.1. Lastly, in Section 6.2 we present the computational results of the EA for both modified sets and the original sets along with necessary performance measures.

6.1 The Benchmark Problem Sets

The 2E-LRP benchmark instances in the literature are generally generated by placing a CDC in the first echelon of the 2E-VRP benchmark sets. In this case, there is no facility opening decision in the first echelon; thus, the target distribution network structure cannot be obtained using the existing benchmark sets. Even the existing sets having more than one candidate CDC, either primary facilities have no capacity restrictions, or vehicles have capacities. The sets with hard time windows, facility and vehicle capacities first emerged in Farham's doctoral thesis [15] by appropriately modifying existing benchmark instances. In this study, we performed computational experiments on these two sets.

Set 1 is generated by modifying Solomon's (1987) benchmark sets to include candidate locations for CDC and satellite nodes. The instances vary in many features such as fleet size, vehicle capacity, and customers' spatial and temporal distribution. The test instances are named after the customer distributions with three categories. The instances starting with C refer to the clustered type of customers, while in randomly generated sets (R), all customers are located randomly on the plane. RC is a category of instances having a mixture of random and clustered customers. The instances are also categorized according to the time window properties. In the first category, we observe tight time windows with low vehicle capacity; however, we have longer time intervals and vehicles with more capacities in the second category. In the instances, the candidate CDC locations' size equal to 2, candidate satellite nodes vary between 2 to 4 and containing 15 to 30 customer nodes.

The second group of instances, Set 2, are again obtained by modifying existing benchmark instances originally generated for 2E-VRPTW. Initially, instances have candidate locations for CDC and satellite points, but they cannot represent physical boundaries such as capacities and economic issues as opening costs. Consequently, to represent our logistics model properties, capacity and opening costs to facilities at both echelons are assigned. The instances are grouped namely a,b,c and d considering customer time windows and demands. Although Group-a and Group-b have similar tight time windows, Group-b has more diversity in demand distributions. The same demand distributions appear in Group-c and Group-d, but Group-d has tighter time windows than Group-c. The main features of each group are: $i \in \{2, 3, 6\}$ candidate CDC points, $j \in \{3, 4, 5\}$ candidate satellite locations and $k \in \{15, 30, 50, 100\}$ customers.

6.2 Evaluation of the Results

The proposed evolutionary algorithm for both single and two-echelon models is coded in C++ compiled with Visual Studio 19 on a computer with Intel® Core™ i7 3.10 GHz processors 16GB memory under the 64-bit Microsoft Windows 10 operating system.

Recall the Algorithm 10 in Chapter 4 that the proposed method requires parameters. While developing the algorithm, a parameter set used to see the results or check whether there were errors. Rather than working on small representative sets, we tested the heuristics on all sets, namely Set 1 and Set 2. In this way, we found the parameter set that results in the best overall performance in all instances, not in a created

Table 6.1: Parameters for the proposed algorithm

Parameter	**Description**	**Value**
ω_1	Population size	100
ω_2	Number of individuals created by Random Create	70
ω_3	Mating pool size	90
ω_4	Probability of crossover	0.9
ω_5	Probability of mutation	0.9
ω_6	Number of generations that EA evolves	500
ω_7	Number of replications	20

test environment. Since we have been working with all instances from the beginning, no additional tuning was required. The parameters we have obtained in preliminary experiments are shown in Table 6.1. Our performance criterion was an average deviation from the optimal (best-known) solution while determining the parameter set.

6.2.1 Single-echelon Results

In this section, we represent the computational results of the proposed algorithm to solve the 2E-LRPTW problem with given open CDC locations; in other words, solutions for the single echelon structure. While evaluating the performance of the algorithm, the following measures are considered: Total Cost indicates the total cost of the best solution that our algorithm in 20 replications, Gap demonstrates the percent deviation of the best solution we have from the optimal (best-known) solution. S_{EA} and S_{BKS} indicate the satellites open in our algorithm's best solution and the satellites open in the optimal (best-known) solution because preliminary experiments showed us that the primary source of the deviation comes from the incorrect facility configurations. CPU time in seconds is reported as T_s.

In each result table, the instances are grouped into three columns according to the number of candidate satellites, $|J|$, to distinguish the proposed EA's performance depending on complexity. According to the number of customers, we consider instances having 15-30 customers as small-sized, 50 customers as medium-sized, and 100 as large-sized. All Set 1 instances are small-sized so, we created three separate tables considering the customer distributions namely C, R, and RC. However, tables are reported for each customer size, $|K|$, for Set 2 instances since variety exists. The average CPU times for BP solutions proposed by Farham [15] are reported in parenthesis near T_s in each group average to state that the proposed EA outperforms the BP results in computational time.

Table 6.2: Results for Clustered (C) instances

		$\lvert J\rvert = 2$					$\lvert J\rvert = 3$					$\lvert J\rvert = 4$				
$\lvert K\rvert$	Instance	Total Cost	Gap	S_{EA}	S_{BKS}	T_s	Total Cost	Gap	S_{EA}	S_{BKS}	T_s	Total Cost	Gap	S_{EA}	S_{BKS}	T_s
	C101	1454.60	0.00	1 2	1 2	0.05	1400.50	0.00	1 2	1 2	0.10	1399.80	0.00	1 2	1 2	0.06
	C102	1485.07	2.27	1 2	1 2	4.75	1397.92	0.12	1 2	1 2	0.21	1400.64	0.46	1 2	1 2	0.18
	C103	1455.11	0.21	1 2	1 2	0.04	1397.92	0.12	1 2	1 2	0.15	1400.64	0.46	1 2	1 2	7.58
	C104	1455.11	0.35	1 2	1 2	0.09	1391.10	0.00	1 2	1 2	0.36	1393.88	0.14	1 2	1 2	0.18
15	C105	1454.60	0.00	1 2	1 2	0.03	1400.50	0.00	1 2	1 2	0.09	1399.80	0.00	1 2	1 2	0.05
	C106	1454.60	0.00	1 2	1 2	0.05	1400.50	0.00	1 2	1 2	0.10	1399.80	0.00	1 2	1 2	0.05
	C107	1455.11	0.16	1 2	1 2	0.03	1401.26	0.24	1 2	1 2	0.07	1400.64	0.25	1 2	1 2	7.52
	C108	1455.11	0.21	1 2	1 2	0.03	1401.26	0.38	1 2	1 2	0.07	1400.64	0.39	1 2	1 2	0.03
	C109	1455.11	0.30	1 2	1 2	0.03	1401.26	0.38	1 2	1 2	15.06	1397.47	0.35	1 2	1 2	0.08
	Average		**0.39**			**0.63 (29.20)**		**0.14**			**2.01 (101.60)**		**0.23**			**1.96 (197.30)**
	C101	1541.40	0.00	1 2	1 2	0.74	1479.02	0.38	1 2	1 2	0.40	1462.26	0.51	1 2	1 2	0.55
	C102	1509.10	0.00	1 2	1 2	1.05	1453.87	0.91	1 2	1 2	0.53	1457.37	0.99	1 2	1 2	15.74
	C103	1515.00	0.39	1 2	1 2	0.57	1439.00	0.00	1 2	1 2	15.45	1445.57	0.01	1 2	1 2	0.57
	C104	**1508.16**	**-0.29**	1 2	1 2	0.56	1433.20	0.00	1 2	1 2	0.42	**1442.28**	**-0.08**	1 2	1 2	0.53
20	C105	1527.63	0.55	1 2	1 2	0.73	1455.00	0.00	1 2	1 2	0.48	1458.84	0.47	1 2	1 2	0.49
	C106	1541.40	0.00	1 2	1 2	0.18	1474.05	0.36	1 2	1 2	0.65	1462.26	0.71	1 2	1 2	0.48
	C107	1533.12	0.91	1 2	1 2	0.60	1455.00	0.00	1 2	1 2	0.40	1457.62	0.39	1 2	1 2	0.71
	C108	1527.29	0.53	1 2	1 2	1.04	1455.76	0.70	1 2	1 2	0.37	1447.40	0.00	1 2	1 2	0.54
	C109	1518.14	1.06	1 2	1 2	0.81	1435.20	0.00	1 2	1 2	0.37	1448.80	0.10	1 2	1 2	0.42
	Average		**0.35**			**0.69 (3331.30)**		**0.26**			**2.33 (3436.10)**		**0.34**			**2.43 (4501.90)**
	C101	1599.84	1.20	1 2	1 2	1.05	1486.96	2.42	1 2	1 2	1.16	1456.22	0.26	1 2	1 2	2.21
	C102	1599.84	1.23	1 2	1 2	1.43	1489.92	2.70	1 2	1 2	0.83	1452.45	0.21	1 2	1 2	3.17
	C103	1601.89	1.33	1 2	1 2	2.99	1481.03	1.41	1 2	1 2	0.87	1457.27	0.54	1 2	1 2	1.76
	C104	1596.45	1.42	1 2	1 2	2.26	1481.03	2.31	1 2	1 2	1.06	1449.40	0.00	1 2	1 2	1.94
25	C105	1610.11	2.28	1 2	1 2	1.81	1487.39	2.45	1 2	1 2	0.31	1459.38	0.47	1 2	1 2	1.19
	C106	1599.84	1.20	1 2	1 2	0.36	1490.99	2.70	1 2	1 2	0.31	1492.14	2.73	1 2	1 2	1.02
	C107	1590.70	1.05	1 2	1 2	1.89	1487.39	2.72	1 2	1 2	0.32	1455.67	0.22	1 2	1 2	1.35
	C108	1592.25	1.15	1 2	1 2	2.71	1485.99	2.62	1 2	1 2	0.35	1455.57	0.30	1 2	1 2	1.39
	C109	1605.60	1.99	1 2	1 2	4.85	1481.76	2.33	1 2	1 2	0.65	1452.74	0.22	1 2	1 2	1.56
	Average		**1.43**			**2.29 (3815.00)**		**2.41**			**0.59 (5231.70)**		**0.55**			**1.67 (4892.5)**
	C101	1632.25	2.13	1 2	1 2	2.71	1505.68	2.66	1 2	1 2	23.61	1468.30	0.00	1 2	1 2	4.88
	C102	1615.09	1.08	1 2	1 2	5.57	1500.27	2.36	1 2	1 2	18.35	1466.20	0.00	1 2	1 2	3.81
	C103	1612.69	0.61	1 2	1 2	7.36	1494.71	0.46	1 2	1 2	3.58	**1468.12**	**-1.49**	1 2	1 2	4.38
	C104	1606.38	0.13	1 2	1 2	28.32	1488.77	0.23	1 2	1 2	2.28	**1466.57**	**-0.06**	1 2	1 2	5.60
30	C105	1610.85	1.27	1 2	1 2	4.88	1505.68	2.66	1 2	1 2	8.55	1468.30	0.00	1 2	1 2	2.25
	C106	1639.01	2.55	1 2	1 2	4.06	1506.28	2.70	1 2	1 2	8.54	1471.24	0.20	1 2	1 2	2.20
	C107	1608.94	1.15	1 2	1 2	3.44	1502.61	2.49	1 2	1 2	16.08	1468.30	0.00	1 2	1 2	2.33
	C108	1628.38	2.37	1 2	1 2	3.24	1502.61	2.50	1 2	1 2	39.03	1470.69	0.22	1 2	1 2	2.37
	C109	1608.94	1.15	1 2	1 2	5.63	1494.98	1.81	1 2	1 2	47.33	1471.24	0.26	1 2	1 2	4.11
	Average		**1.38**			**7.81 (6067.30)**		**1.99**			**17.97 (6658.30)**		**-0.10**			**3.38 (7376.10)**
	Grand Average		**0.89**			**2.68**		**1.20**			**5.78**		**0.26**			**2.32**

From Table 6.2, as expected, when the number of customers is increased the deviation from the optimal solution increases. We were able to obtain a total of 24 optimal results in instances with clustered structure. Besides, we reported improvements on the upper bounds of 4 best-know solutions. The maximum average deviation is below 2.5% and the maximum deviation among all of the clustered sets is 2.73%, which is considerably reasonable.

Table 6.3: Results for Random (R) instances

		$\lvert J\rvert = 2$					$\lvert J\rvert = 3$					$\lvert J\rvert = 4$				
$\lvert K\rvert$	**Instance**	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s
15	R101	1347.50	0.00	1 2	1 2	2.29	1347.50	0.00	1 3	1 3	3.09	1326.39	0.06	3 4	3 4	1.38
	R102	1313.90	0.00	1 2	1 2	2.75	1291.93	0.63	1 2	1 2	3.34	1282.71	0.06	2 3	2 3	2.42
	R103	1311.51	0.06	1 2	1 2	5.23	1291.93	0.63	1 2	1 2	1.90	1282.71	0.06	2 3	2 3	25.73
	R104	1282.20	0.00	1 2	1 2	4.21	1276.12	0.06	1 2	1 2	2.42	1273.80	0.00	2 3	2 3	9.94
	R105	1336.10	0.00	1 2	1 2	2.36	1326.90	0.07	1 2	1 2	10.19	1326.39	0.06	3 4	3 4	2.15
	R106	1297.23	0.44	1 2	1 2	0.05	1299.99	1.56	1 2	1 2	2.48	1278.23	0.06	2 3	2 3	3.21
	R107	1297.23	0.44	1 2	1 2	5.37	1299.99	1.56	1 2	1 2	2.34	1278.23	0.06	2 3	2 3	3.51
	R108	1252.20	0.00	1 2	1 2	2.30	1252.20	0.00	1 3	1 3	2.43	1252.20	0.00	1 4	1 4	2.29
	R109	1299.65	0.06	1 2	1 2	3.15	1309.89	0.85	1 2	1 2	3.75	1291.10	0.00	3 4	3 4	2.98
	R110	1285.36	0.96	1 2	1 2	2.10	1273.84	0.06	1 3	1 3	33.49	1277.44	0.34	3 4	1 4	1.75
	R111	1287.40	0.00	1 2	1 2	3.29	1299.97	1.56	1 2	1 2	3.08	1274.72	0.06	2 3	2 3	2.09
	R112	1268.85	0.18	1 2	1 2	1.67	1268.85	0.18	1 3	1 3	1.93	1267.44	0.07	3 4	1 4	9.27
	Average		**0.18**			**2.54 (19.10)**		**0.60**			**7.46 (35.50)**		**0.07**			**3.40 (70.90)**
20	R101	1464.11	0.07	1 2	1 2	27.04	1446.33	0.29	1 2	2 3	3.12	1432.67	0.10	3 4	3 4	2.97
	R102	1416.55	0.06	1 2	1 2	10.79	1373.19	0.09	1 2	1 2	3.06	1376.01	0.10	2 3	2 3	4.68
	R103	1353.64	0.06	1 2	1 2	254.86	1336.04	0.09	1 2	1 2	4.66	1318.99	0.08	2 3	2 3	6.22
	R104	1337.00	0.00	1 2	1 2	4.85	1336.04	1.89	1 2	1 2	6.27	1310.82	0.09	2 3	2 3	4.40
	R105	1399.69	0.06	1 2	1 2	4.58	1393.55	0.51	1 2	1 2	4.78	1383.13	0.07	3 4	3 4	10.43
	R106	1384.15	0.27	1 2	1 2	0.18	1360.02	1.04	1 2	1 2	6.43	1354.60	0.43	2 3	2 3	11.67
	R107	1340.65	0.06	1 2	1 2	8.00	1329.19	0.48	1 2	1 2	4.25	1312.62	0.25	2 3	2 3	13.01
	R108	1331.84	0.65	1 2	1 2	3.67	1323.61	1.68	1 2	1 2	4.37	1297.52	0.08	2 3	2 3	4.12
	R109	1361.91	0.23	1 2	1 2	5.24	1355.38	0.24	1 2	1 2	3.97	1354.46	0.11	2 3	2 3	3.78
	R110	1331.76	0.06	1 2	1 2	2.94	1324.12	0.08	1 2	1 2	3.65	1329.22	0.08	3 4	3 4	2.29
	R111	1337.13	0.06	1 2	1 2	5.74	1336.04	0.70	1 2	1 2	4.03	1318.99	0.08	2 3	2 3	4.40
	R112	1309.60	0.06	1 2	1 2	4.21	1320.56	1.24	2 3	1 2	3.39	1308.24	0.07	3 4	3 4	2.28
	Average		**0.14**			**4.32 (859.60)**		**0.70**			**4.36 (1682.40)**		**0.13**			**6.50 (600.10)**
25	R101	1587.69	0.07	1 2	1 2	20.70	1590.97	1.53	2 3	1 2	0.92	1568.82	0.18	3 4	3 4	5.60
	R102	1517.24	0.09	1 2	1 2	11.36	1503.14	1.13	1 2	1 2	27.50	1496.14	0.09	2 3	2 3	12.87
	R103	1462.33	0.77	1 2	1 2	11.13	1441.86	0.85	1 2	1 2	1.93	1445.79	**-0.38**	3 4	1 4	16.82
	R104	1430.97	1.17	1 2	1 2	13.33	1431.24	1.88	2 3	1 2	2.35	1419.93	**-0.23**	2 4	3 4	8.58
	R105	1508.60	0.57	1 2	1 2	7.89	1497.72	0.30	1 2	1 2	6.35	1485.03	0.09	3 4	3 4	8.37
	R106	1445.41	0.07	1 2	1 2	0.36	1444.68	0.10	1 2	1 2	8.60	1445.41	0.07	1 4	1 4	10.15
	R107	1399.60	0.07	1 2	1 2	6.84	1427.63	1.71	1 2	1 2	7.38	1403.03	**-0.79**	3 4	3 4	8.04
	R108	1389.86	0.29	1 2	1 2	482.81	1390.41	**-0.06**	1 3	1 3	38.41	1386.71	0.06	1 4	1 4	8.35
	R109	1439.88	0.08	1 2	1 2	19.69	1465.47	3.28	1 2	1 2	7.92	1431.39	0.11	2 3	2 3	13.48
	R110	1425.44	0.24	1 2	1 2	8.67	1454.09	2.26	1 3	1 3	13.38	1421.43	0.09	2 3	2 3	14.74
	R111	1414.40	0.06	1 2	1 2	8.55	1420.19	1.27	1 2	1 2	6.57	1417.69	0.29	3 4	1 4	19.14
	R112	1376.46	0.12	1 2	1 2	6.80	1375.46	**-0.31**	1 3	1 3	9.78	1388.26	0.98	3 4	1 4	44.75
	Average		**0.30**			**67.70 (7899.00)**		**1.16**			**12.30 (9496.10)**		**0.05**			**15.88 (8554.90)**
30	R101	1619.59	0.10	1 2	1 2	76.42	1611.18	0.50	2 3	1 2	16.89	1610.93	0.12	3 4	3 4	12.01
	R102	1544.65	0.60	1 2	1 2	867.63	1528.81	0.94	2 3	1 2	1.62	1535.17	0.11	3 4	3 4	18.48
	R103	1459.73	0.67	1 2	1 2	9.54	1472.23	1.81	2 3	1 2	3.77	1451.51	0.10	1 4	1 4	308.06
	R104	1408.49	0.10	1 2	1 2	531.76	1410.73	**-0.76**	1 2	1 2	4.77	1408.49	**-1.01**	1 4	3 4	536.06
	R105	1550.48	0.26	1 2	1 2	167.81	1550.48	0.26	1 3	1 3	129.66	1540.96	0.11	3 4	3 4	15.67
	R106	1497.85	1.17	1 2	1 2	125.80	1504.68	1.66	1 2	1 2	13.01	1495.29	1.16	1 3	2 3	30.88
	R107	1430.33	0.39	1 2	1 2	25.65	1430.33	0.39	1 3	1 3	22.70	1427.66	**-0.08**	1 4	1 4	411.18
	R108	1392.12	**-1.02**	1 2	1 2	37.23	1385.20	**-0.37**	1 3	1 3	25.56	1392.12	**-1.25**	1 4	1 4	50.82
	R109	1462.19	0.38	1 2	1 2	36.99	1458.36	0.12	1 3	1 3	50.08	1458.36	0.12	1 4	1 4	42.61
	R110	1451.36	0.45	1 2	1 2	25.19	1446.15	0.09	1 3	1 3	13.81	1449.10	0.30	1 4	1 4	65.84
	R111	1437.10	0.23	1 2	1 2	8.78	1435.33	**-0.01**	1 3	1 3	22.45	1435.33	0.10	1 4	1 4	43.01
	R112	1386.56	0.10	1 2	1 2	13.48	1433.33	3.47	2 3	1 3	10.33	1386.56	0.10	1 4	1 4	16.15
	Average		**0.29**			**55.12 (3703.30)**		**0.68**			**35.95 (6820.60)**		**-0.01**			**84.52 (7386.10)**
	Grand Average		**0.22**			**58.10**		**0.78**			**12.08**		**0.06**			**37.86**

Instance properties are highly influential on the solutions obtained. For the exact solution approaches or heuristics, it is easier to solve the clustered instances since results are less sensitive to the small distances among customers. However, our algorithm has no diversity problem while creating routes or opening facilities; thus, randomly generated instances can be solved contentedly. We can verify this inference from the Table 6.3. If we compare the average deviations obtained when the number of candidate satellites is two in clustered and random instances, it is clear that the proposed EA gives overall better results in random sets. Especially in the initialization and the reproduction phases, randomness enabled us to capture aspects that heuristics could not capture.

Table 6.4: Results for Random-Clustered (RC) instances

		|J| = 2					|J| = 3					|J| = 4				
|K|	**Instance**	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s
15	RC101	1720.25	0.73	1 2	1 2	0.25	1524.53	0.06	1 2	1 2	0.19	1553.80	0.00	1 2	1 2	15.23
	RC102	1605.50	0.00	1 2	1 2	0.23	1501.48	0.44	1 2	1 2	15.23	1534.86	0.32	1 2	1 2	7.79
	RC103	1641.61	2.25	1 2	1 2	0.32	1501.48	0.44	1 2	1 2	15.31	1530.74	0.05	1 2	1 2	15.29
	RC104	1627.04	1.75	1 2	1 2	0.72	1493.50	0.32	1 2	1 2	0.18	1524.28	0.04	1 2	1 2	7.77
	RC105	1649.81	0.19	1 2	1 2	0.41	1533.31	0.19	1 2	1 2	0.18	1578.03	0.17	1 2	1 2	7.85
	RC106	1639.63	2.02	1 2	1 2	0.05	1502.50	0.00	1 2	1 2	0.29	1538.55	0.07	1 2	1 2	7.67
	RC107	1594.90	0.00	1 2	1 2	8.23	1484.10	0.04	1 2	1 2	0.18	1518.40	0.00	1 2	1 2	0.16
	RC108	1588.20	0.00	1 2	1 2	0.34	1483.81	0.12	1 2	1 2	7.74	1514.70	0.00	1 2	1 2	0.18
	Average		**0.87**			**1.32 (56.50)**		**0.20**			**4.91 (419.60)**		**0.08**			**7.74 (237.00)**
20	RC101	1822.19	5.06	1 2	1 2	29.67	1615.48	0.38	1 2	1 2	0.72	1609.40	0.00	2 3	2 3	0.24
	RC102	1716.91	0.40	1 2	1 2	24.51	1580.10	0.00	1 2	1 2	0.92	1586.30	0.00	2 3	2 3	0.74
	RC103	1714.90	0.88	1 2	1 2	37.10	1577.00	0.00	1 2	1 2	0.77	1587.71	0.09	2 3	2 3	0.25
	RC104	1696.79	0.35	1 2	1 2	41.76	1572.29	0.09	1 2	1 2	0.59	1579.97	0.23	2 3	2 3	0.33
	RC105	1808.70	4.49	1 2	1 2	0.55	1607.66	0.38	1 2	1 2	1.24	1610.09	0.07	2 3	2 3	1.01
	RC106	1778.78	4.24	1 2	1 2	0.18	1595.97	0.52	1 2	1 2	0.65	1592.06	0.07	2 3	2 3	0.19
	RC107	1723.23	2.58	1 2	1 2	0.60	1567.30	0.00	1 2	1 2	0.70	1570.10	0.00	2 3	2 3	0.78
	RC108	1701.90	1.59	1 2	1 2	36.43	1566.77	0.09	1 2	1 2	0.74	1566.90	0.00	2 3	2 3	0.82
	Average		**2.45**			**21.35 (340.60)**		**0.18**			**0.79 (3835.50)**		**0.06**			**0.55 (3073.80)**
25	RC101	1939.95	5.97	1 2	1 2	0.93	1726.97	3.37	1 2	1 2	0.75	1650.98	0.07	2 3	2 3	1.29
	RC102	1830.76	4.96	1 2	1 2	0.78	1623.20	0.52	1 2	1 2	1.30	1621.49	0.07	2 3	2 3	1.33
	RC103	1744.03	0.68	1 2	1 2	1.55	1622.73	1.13	1 2	1 2	1.62	1609.28	0.09	2 3	2 3	8.88
	RC104	1725.15	1.10	1 2	1 2	63.98	1602.50	1.62	1 2	1 2	2.10	1581.00	0.17	2 3	2 3	1.95
	RC105	1867.52	6.27	1 2	1 2	1.37	1683.60	3.53	1 2	1 2	1.05	1630.03	0.08	2 3	2 3	9.17
	RC106	1819.86	4.65	1 2	1 2	0.36	1661.75	3.21	1 2	1 2	0.88	1612.60	0.00	2 3	2 3	15.97
	RC107	1778.05	4.54	1 2	1 2	1.13	1628.16	3.23	1 2	1 2	8.55	1581.50	0.00	2 3	2 3	31.42
	RC108	1756.86	3.46	1 2	1 2	1.77	1622.72	3.02	1 2	1 2	9.02	1576.80	0.00	2 3	2 3	16.02
	Average		**3.95**			**8.98 (4735.10)**		**2.45**			**3.16 (3522.70)**		**0.06**			**10.75 (4875.10)**
30	RC101	1979.77	1.91	1 2	1 2	0.97	1947.89	2.59	1 2	1 2	2.00	1885.88	0.07	2 3	2 3	10.76
	RC102	1933.80	4.21	1 2	1 2	2.25	1848.26	0.11	1 2	1 2	2.38	1829.86	0.44	3 4	3 4	3.34
	RC103	1859.47	3.70	1 2	1 2	4.64	1741.50	0.68	1 2	1 2	2.55	1731.26	0.07	2 3	2 3	3.37
	RC104	1853.14	4.74	1 2	1 2	2.19	1683.49	0.49	1 2	1 2	2.57	1705.54	0.22	2 3	2 3	5.43
	RC105	1925.96	4.71	1 2	1 2	2.06	1799.41	2.13	1 2	1 2	2.30	1764.90	0.07	2 3	2 3	79.01
	RC106	1872.23	3.71	1 2	1 2	1.47	1769.14	1.78	1 2	1 2	2.11	1743.86	0.23	2 3	2 3	17.02
	RC107	1833.35	3.83	1 2	1 2	2.33	1699.90	0.00	1 2	1 2	3.98	1699.40	0.00	2 3	2 3	16.97
	RC108	1763.00	0.00	1 2	1 2	1.84	1723.19	2.32	1 2	1 2	2.22	1694.70	0.00	2 3	2 3	17.65
	Average		**3.35**			**2.22 (7752.30)**		**1.26**			**2.51 (5470.60)**		**0.14**			**19.19 (8403.60)**
	Grand Average		**2.65**			**8.47**		**1.02**			**2.84**		**0.08**			**9.56**

In R sets, we can only report 12 optimal solutions in sets having 15 and 20 customers, but average deviations are low. We are also reporting new best-known solutions by improving 7 upper bounds in sets with 25 customers, 5 in sets with 30 customers. The maximum deviation is 3.47%, although the reported gap greater than the maximum of clustered sets; overall, we obtain smaller percentage deviations in random instances.

One of the limitations comes from the capacities of the satellites. Considering both hard time windows and satellite capacities while allocating, allocations might differ from the optimal solution resulting in higher deviations. Therefore, when the number of candidate satellites is small, and the satellites have tight capacities, the average deviation increases. According to Table 6.4, we can produce 21 optimal solutions as we can in clustered sets, but in sets with 2 candidate satellites, the deviation is significantly higher than in other instances. While there are 2 satellites in RC sets, deviation from best-known is more, but deviation decreases noticeably as the number of satellites increases. This is because the proposed algorithm has substantial exploration property in non-tight environments.

Table 6.5: Set 2 results for instances having 15 customers

	$\|J\| = 3$					$\|J\| = 4$					$\|J\| = 5$				
Instance	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s
15 a1	1652.27	0.05	1 2	1 2	6.25	1639.51	0.00	2 3	2 3	7.02	1648.70	0.00	1 3	1 3	20.14
15 a2	1641.38	0.00	2 3	2 3	0.29	1658.03	0.00	1 4	1 4	0.88	1690.83	0.00	3 4	3 4	0.45
15 a3	1615.53	0.00	1 2	1 2	0.37	1605.91	0.13	1 2	1 2	13.82	1585.63	0.00	4 5	4 5	13.52
15 a4	1673.19	0.75	1 3	1 3	0.34	1531.44	0.00	2 3	2 3	33.53	1598.90	0.00	1 4	1 4	25.01
15 a5	1654.34	0.00	1 2	1 2	0.45	1526.17	0.00	1 2	1 2	20.30	1551.18	0.00	3 5	3 5	24.65
Average		**0.16**			**1.54 (3.06)**		**0.03**			**15.11 (9.37)**		**0.00**			**16.75 (11.24)**
15 b1	1660.36	0.00	1 2	1 2	6.87	1643.16	0.00	2 3	2 3	20.21	1619.59	0.00	3 4	3 4	26.87
15 b2	1660.24	0.00	2 3	2 3	20.16	1684.67	0.00	2 3	2 3	0.64	1672.19	0.00	1 4	1 4	7.14
15 b3	1675.17	0.00	1 2	1 2	0.46	1663.94	0.99	3 4	3 4	7.14	1587.83	0.00	3 4	3 4	20.32
15 b4	1646.91	0.00	2 3	2 3	0.31	1592.01	0.00	2 3	2 3	7.38	1520.49	0.01	3 4	3 4	66.47
15 b5	1638.46	0.77	1 2	2 3	0.58	1548.37	0.00	1 2	1 2	0.43	1617.81	0.00	3 5	3 5	40.21
Average		**0.15**			**5.68 (8.19)**		**0.20**			**7.16 (9.33)**		**0.00**			**32.30 (24.64)**
15 c1	1639.04	0.00	1 2	1 2	6.89	1645.20	0.00	2 3	2 3	6.96	1550.92	0.00	1 3	1 3	20.38
15 c2	1619.16	0.00	2 3	2 3	0.47	1626.22	0.00	1 4	1 4	0.31	1587.90	0.00	1 4	1 4	23.07
15 c3	1639.39	0.00	1 2	1 2	0.74	1637.67	0.20	1 2	1 2	0.51	1581.68	0.06	3 4	4 5	19.75
15 c4	1636.43	0.00	2 3	2 3	0.28	1583.55	0.00	2 3	2 3	77.00	1590.89	0.04	3 4	3 4	65.63
15 c5	1552.93	0.00	2 3	2 3	8.23	1520.02	0.00	1 2	1 2	0.42	1598.56	0.00	3 5	3 5	19.68
Average		**0.00**			**3.32 (6.27)**		**0.04**			**17.04 (12.19)**		**0.02**			**29.70 (20.65)**
15 d1	1649.17	0.00	1 2	1 2	13.57	1633.29	0.00	2 3	2 3	6.96	1641.84	0.00	1 3	1 3	66.21
15 d2	1588.20	0.00	2 3	2 3	33.30	1631.59	0.00	1 4	1 4	6.93	1609.51	0.00	4 5	4 5	39.94
15 d3	1622.36	0.00	1 2	1 2	0.38	1654.34	0.00	3 4	3 4	7.09	1592.27	0.00	4 5	4 5	66.08
15 d4	1649.35	0.00	1 3	1 3	0.57	1598.46	0.00	2 3	2 3	92.52	1588.45	0.00	3 4	3 4	20.40
15 d5	1636.30	0.00	1 2	1 2	20.07	1520.02	0.00	1 2	1 2	6.95	1603.16	0.00	3 5	3 5	33.22
Average		**0.00**			**13.58 (2.50)**		**0.00**			**24.09 (8.25)**		**0.00**			**45.17 (10.14)**
Grand Average		**0.08**			**6.03**		**0.07**			**15.85**		**0.01**			**30.96**

For the Set 2 instances having 15 customers, the proposed EA can able to find the optimal solutions for 51 instances among 60. Only in two instances, the location decisions for candidate satellites are incorrect, and even if the location decisions are accurate, there exist small gaps from the optimal. Table 6.5 shows that the maximum deviation is 0.99%, and the averages for each group of candidate satellites are below

0.1%.

Table 6.6 demonstrates that when the number of customers increased, the time does not necessarily increase. It is expected that the solution time will increase when the size increases; however, each generation can be solved within a reasonable time, but the number of the generation that the algorithm evolves determines the total time. Because when solutions do not converge, the algorithm proceeds until the maximum number of generations. We reported 27 optimal solutions and the solutions deviate a maximum of 0.92% from the optimal.

Table 6.6: Set 2 results for instances having 30 customers

	|J| = 3					|J| = 4					|J| = 5				
Instance	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s
30 a1	2122.91	0.03	1 3	1 3	54.76	2171.02	0.14	1 2	1 2	3.64	2184.14	0.21	3 4	3 4	5.93
30 a2	2128.50	0.00	2 3	2 3	2.48	2155.14	0.00	2 3	2 3	4.41	2170.85	0.00	2 3	2 3	3.42
30 a3	2207.19	0.25	1 3	1 3	8.22	2128.92	0.92	3 4	3 4	3.50	2116.41	0.00	2 3	2 3	16.36
30 a4	2136.08	0.00	1 2	1 2	3.07	2114.73	0.02	2 3	2 3	10.15	2140.17	0.49	4 5	4 5	4.75
30 a5	2215.68	0.10	2 3	2 3	4.31	2131.11	0.00	1 2	1 2	3.72	2076.96	0.04	4 5	4 5	9.45
Average		**0.08**			**14.57 (22.76)**		**0.22**			**5.08 (49.68)**		**0.15**			**7.98 (56.07)**
30 b1	2122.20	0.15	1 3	1 3	8.69	2200.55	0.08	1 2	1 2	2.74	2189.02	0.00	3 4	3 4	4.45
30 b2	2018.18	0.69	2 3	2 3	3.14	2180.56	0.00	3 4	3 4	4.59	2191.32	0.00	3 4	3 4	4.93
30 b3	2206.22	0.06	1 3	1 3	24.23	2161.27	0.11	2 3	2 3	2.00	2120.01	0.00	2 3	2 3	11.06
30 b4	2166.90	0.36	1 2	1 2	3.23	2131.86	0.48	2 3	2 3	4.47	2142.20	0.74	4 5	4 5	9.33
30 b5	1909.84	0.81	2 3	2 3	3.23	2168.05	0.24	1 2	1 2	4.00	2151.00	0.00	4 5	4 5	3.44
Average		**0.41**			**8.50 (88.62)**		**0.18**			**3.56 (110.28)**		**0.15**			**6.64 (63.36)**
30 c1	1802.38	0.16	1 3	1 3	4.90	2125.79	0.24	1 2	1 2	3.70	2209.80	0.00	3 4	3 4	3.58
30 c2	2098.61	0.00	2 3	2 3	2.83	2137.77	0.00	3 4	3 4	2.39	2117.71	0.06	3 4	3 4	3.00
30 c3	2149.85	0.00	1 3	1 3	10.98	2114.25	0.00	2 3	2 3	2.56	2154.62	0.00	2 3	2 3	3.65
30 c4	2128.25	0.00	1 2	1 2	2.07	2087.60	0.12	2 3	2 3	3.25	2111.80	0.91	4 5	4 5	10.44
30 c5	2179.26	0.00	2 3	2 3	3.19	1926.64	0.02	2 4	2 4	59.38	1800.27	0.26	4 5	4 5	15.41
Average		**0.03**			**4.79 (51.47)**		**0.08**			**14.25 (114.05)**		**0.25**			**7.22 (128.59)**
30 d1	1809.62	0.13	1 3	1 3	9.94	2175.11	0.44	1 2	1 2	4.17	2227.50	0.00	3 4	3 4	2.89
30 d2	2108.97	0.00	2 3	2 3	4.22	2148.66	0.02	3 4	3 4	2.13	2186.27	0.00	3 4	3 4	4.32
30 d3	2166.99	0.19	1 3	1 3	17.72	2144.36	0.70	3 4	3 4	2.92	2117.07	0.00	2 3	2 3	3.01
30 d4	2137.98	0.00	1 2	1 2	3.64	2105.15	0.00	2 3	2 3	2.22	2119.03	0.16	4 5	4 5	2.11
30 d5	2197.54	0.00	2 3	2 3	3.15	2197.54	0.00	1 2	1 2	3.31	2105.16	0.12	4 5	4 5	4.30
Average		**0.06**			**7.73 (10.08)**		**0.23**			**2.95 (47.97)**		**0.06**			**3.33 (46.77)**
Grand Average		**0.15**			**8.90**		**0.18**			**6.46**		**0.15**			**6.29**

According to Table 6.7, the algorithm finds only optimal solutions for four instances. Nevertheless, the EA can improve the existing upper bounds for two instances that are not reported as optimally by [15]. Although the instances' size increases more than three times considering the 15 customer instances, the average deviations are still below 0.5%, and the maximum deviation equal to 1.61%. Only one satellite configuration is not accurate among 60 instances. Therefore, the reader may realize that the source of the variation comes from the allocations.

Table 6.7: Set 2 results for instances having 50 customers

	|J| = 3					|J| = 4					|J| = 5				
Instance	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s
50 a1	2661.61	0.83	1 2	1 2	14.06	2588.05	0.26	2 4	2 4	17.06	2592.44	0.17	1 4	1 4	17.09
50 a2	2701.87	0.32	1 3	1 3	162.42	2596.65	0.00	1 3	1 3	13.92	2610.74	0.30	1 4	1 4	21.02
50 a3	2609.07	0.35	2 3	2 3	13.49	2594.07	0.57	1 2	1 2	15.03	2400.81	0.05	2 3	2 3	11.77
50 a4	2564.75	0.16	1 2	1 2	15.91	2490.97	0.00	1 4	1 4	168.46	2498.22	0.06	4 5	4 5	9.69
50 a5	2549.61	0.05	1 3	1 3	166.72	2590.28	0.27	1 4	1 4	166.58	2512.72	0.07	3 4	3 4	9.56
Average		**0.34**			**74.52 (284.37)**		**0.22**			**76.21 (656.99)**		**0.13**			**13.83 (274.09)**
50 b1	2624.49	0.51	1 2	1 2	12.04	2575.33	0.30	1 4	1 4	158.16	**2575.08**	**-0.11**	1 4	1 4	24.97
50 b2	2590.38	0.44	1 3	1 3	170.87	2693.02	1.32	1 3	1 3	17.96	2642.45	0.66	1 4	1 4	21.94
50 b3	2648.41	0.70	2 3	2 3	15.47	2611.27	0.56	1 2	1 2	17.73	2529.24	0.67	2 4	2 4	17.41
50 b4	2588.46	0.30	1 2	1 2	10.68	2515.60	0.07	1 4	1 4	161.05	2523.95	0.14	4 5	4 5	12.21
50 b5	2635.04	1.61	1 3	1 3	165.37	2633.17	0.61	2 4	2 4	28.66	2540.52	0.43	3 4	3 4	9.34
Average		**0.71**			**74.88 (524.78)**		**0.57**			**76.71 (1047.52)**		**0.36**			**17.17 (3279.02)**
50 c1	2556.81	0.45	1 2	1 2	13.35	2524.71	0.49	1 4	1 4	172.32	2522.13	0.17	2 3	2 3	11.81
50c2	2606.22	0.64	1 2	1 2	21.00	2518.44	0.01	1 3	1 3	16.80	2569.54	0.17	1 4	1 4	40.97
50 c3	2530.33	0.26	2 3	2 3	14.20	2581.67	0.08	3 4	3 4	16.19	**2477.87**	**-0.02**	2 4	2 4	29.20
50 c4	2521.33	0.20	1 2	1 2	12.03	2416.15	0.13	1 4	1 4	179.80	2572.21	0.40	4 5	4 5	11.93
50 c5	2562.63	0.45	1 3	1 3	181.18	2742.51	0.00	1 2	1 2	148.89	2573.91	0.27	1 5	1 5	172.06
Average		**0.40**			**48.35 (1870.09)**		**0.14**			**106.80 (2397.18)**		**0.20**			**53.19 (5792.74)**
50 d1	2595.19	0.20	2 3	2 3	12.51	2530.31	0.29	1 4	1 4	170.54	2559.37	0.12	1 4	1 4	18.50
50 d2	2703.74	0.88	1 2	1 2	15.67	2570.48	0.09	1 3	1 3	39.39	2593.44	0.26	1 4	1 4	14.05
50 d3	2596.77	0.18	2 3	2 3	15.53	2648.46	0.41	3 4	3 4	12.45	2492.33	0.01	2 3	2 3	14.31
50 d4	2564.49	0.04	1 2	1 2	21.72	2499.08	0.32	1 4	1 4	171.29	2525.75	0.38	4 5	4 5	7.23
50 d5	2629.76	0.00	1 3	1 3	158.28	2603.56	0.41	3 4	2 3	20.14	2572.57	0.48	3 4	3 4	17.08
Average		**0.26**			**44.74 (210.69)**		**0.31**			**82.76 (359.94)**		**0.25**			**14.23 (872.32)**
Grand Average		**0.43**			**60.62**		**0.31**			**85.62**		**0.23**			**24.61**

Table 6.8: Set 2 results for instances having 100 customers

	|J|= 3					|J|= 4					|J|= 5				
Instance	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s	**Total Cost**	**Gap**	S_{EA}	S_{BKS}	T_s
100 a1	4130.59	0.98	2 3	2 3	42.56	4015.59	0.59	1 3	3 4	41.54	4141.82	**-1.11**	2 5	2 5	102.91
100 a2	4267.73	1.25	1 3	1 3	886.40	3919.25	**-1.06**	1 4	1 4	1014.65	3940.96	1.25	1 4	1 4	48.31
100 a3	3981.38	1.55	1 3	1 3	934.45	4232.14	1.11	3 4	3 4	69.49	3925.86	1.20	3 5	3 5	57.98
100 a4	4265.02	1.13	1 2	1 2	57.69	3911.45	**-0.06**	2 4	1 4	69.68	4057.67	2.14	1 2	1 2	32.42
100 a5	4107.38	1.98	2 3	1 2	33.23	4129.08	1.22	1 4	1 4	916.85	3978.31	0.25	2 3	2 3	24.80
Average		**1.38**			**390.87 (210.69)**		**0.31**			**82.76 (359.94)**		**0.25**			**14.23 (872.32)**
100 b1	3944.25	**-0.29**	2 3	2 3	61.50	3928.35	**-0.15**	1 3	1 2 4	53.39	4174.13	0.45	3 5	3 5	46.50
100 b2	4174.42	1.53	1 3	1 3	954.85	4098.18	2.87	1 4	1 4	934.10	3949.42	0.97	1 4	1 4	58.67
100 b3	4079.13	3.43	1 3	1 3	966.89	4239.75	0.04	2 3	1 3	24.31	3977.95	**-3.67**	3 5	1 2	34.72
100 b4	4203.46	1.20	1 2	1 2 3	40.40	4249.57	2.28	2 4	1 4	55.36	4165.89	3.46	1 2	2 3	32.43
100 b5	4146.44	2.51	2 3	1 2	38.45	4203.50	1.52	2 4	1 4	78.63	4176.06	**-2.34**	2 3	1 3	20.06
Average		**1.68**			**412.42 (12114.35)**		**1.31**			**229.16 (14273.36)**		**-0.23**			**38.48 (14256.46)**
100 c1	4101.25	**-2.17**	2 3	2 3	30.43	3970.29	**-0.84**	1 3	1 3	50.76	4120.17	**-0.95**	2 5	1 4	55.27
100 c2	4065.40	1.07	1 3	1 3	541.78	4093.30	**-0.23**	2 4	1 4	48.11	3897.77	**-1.47**	1 4	1 3	70.58
100 c3	3991.68	1.77	1 3	1 3	510.96	3930.79	**-2.03**	3 4	1 2 4	31.89	4018.17	**-0.74**	1 3	1 2	34.91
100 c4	4131.19	**-1.24**	1 2	1 3	28.55	3998.54	**-2.59**	2 4	2 4	66.01	4004.91	1.19	1 2	2 3	39.53
100 c5	4085.72	1.39	1 2	1 2	64.87	4005.64	**-1.37**	1 4	1 4	524.98	3991.51	**-1.81**	2 3	3 5	39.63
Average		**0.16**			**235.32 (14352.24)**		**-1.41**			**144.35 (14361.00)**		**-0.75**			**47.98 (14322.84)**
100 d1	4131.48	1.92	2 3	1 3	37.32	4067.60	0.85	1 3	3 4	43.08	4138.45	0.81	3 5	3 5	39.26
100 d2	4265.87	1.60	1 3	1 3	470.85	4040.12	2.03	2 4	1 4	59.33	3922.86	0.93	1 4	1 4	56.78
100 d3	3961.86	2.59	1 3	1 3	528.13	3990.15	1.74	1 3	1 3 4	60.42	3962.59	**-1.41**	3 5	3 5	30.79
100 d4	4182.26	1.85	1 2	1 3	28.59	4138.97	1.09	2 4	1 4	52.55	3989.65	1.99	1 2	1 2	19.49
100 d5	4134.11	2.32	2 3	1 2	39.95	4128.84	1.21	1 4	1 4	491.36	4032.26	0.55	2 3	2 3	25.28
Average		**2.06**			**220.97 (8626.84)**		**1.38**			**141.35 (10600.14)**		**0.58**			**34.32 (13357.63)**
Grand Average		**1.32**			**314.89**		**0.41**			**234.32**		**0.09**			**43.51**

When the number of customers increases, wrong assignments of customers result in greater deviations from the optimal solution. Because wrong allocations also directly affect the routes and the routes might end up completely different from optimal solutions.

Table 6.8 demonstrates that the average gap is 1.32%, 0.41%, and 0.09% for instances with 3,4 and 5 candidate satellites respectively. The average deviation from the best-known solution decreases when the number of satellites increases. In order to obtain optimal (best-known) solutions, the algorithm should generate a diversified population. Nonetheless, it seems the EA cannot escape the trap for small environments having multiple local optimums. However, with the larger sets having more alternatives, using randomization EA reaches better solution space points. Although we cannot report an optimal solution for these sets, the total costs of 18 instances are improved by a maximum of 3.67%.

6.2.2 Two-echelon Results

In this section, the computational results of the modified EA to solve the 2E-LRPTW original instances are demonstrated. In order to make our results comparable, in the first echelon cost of direct trips considered rather than creating routes because there are no reported results in the literature for the models having routes at both echelons.

Recall that the original Set 1 instances only have 2 candidate CDC locations in each of the sets. However, Set 2 instances have number of CDC points $i \in \{2, 3, 6\}$ and satellite points vary between 3 to 5 having $k \in \{15, 30, 50, 100\}$ customers. The tables' structure is the same as the previous section for Set 1 results, but we introduced the CDC points, $|I|$, and the candidate satellite points on the tables for Set 2 results. Also, S_{EA} and S_{BKS} are replaced with F_{EA} and F_{BKS} indicating which facilities are opened in the metaheuristic solution and the best-known solution, respectively, to represent not only satellites but also the open CDCs.

Since there are always 2 candidate CDC points in these sets, we can evaluate how the performance changes as the number of satellites increases from Table 6.9. The optimal solutions for 10 sets are reported and the average gap is a maximum 2.83%, and the maximum deviation in these sets is 3.64%. When the number of customers increases and the number of candidate satellites equal to two, deviation increases more than other scenarios. Also, as the number of satellites increases, the gap is expected to grow, while EA reports lower average deviations.

Table 6.9: Results for original Clustered (C) instances

		|J| = 2					|J| = 3					|J| = 4				
|K|	**Instance**	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s
	C101	1454.60	0.00	0 2 3	0 2 3	3.85	1400.50	0.00	1 2 3	1 2 3	7.20	1400.64	0.06	1 2 3	1 2 3	3.80
	C102	1484.09	2.20	0 2 3	0 2 3	79.25	1401.92	0.40	1 2 3	1 2 3	9.80	1395.00	0.06	1 2 3	1 2 3	11.50
	C103	1455.11	0.21	0 2 3	0 2 3	3.65	1397.92	0.12	1 2 3	1 2 3	9.90	1395.45	0.09	1 2 3	1 2 3	6.90
	C104	1458.95	0.61	0 2 3	0 2 3	4.15	1391.74	0.05	1 2 3	1 2 3	14.90	1393.88	0.14	1 2 3	1 2 3	8.00
15	C105	1454.60	0.00	0 2 3	0 2 3	3.80	1400.50	0.00	1 2 3	1 2 3	7.55	1400.64	0.06	1 2 3	1 2 3	3.80
	C106	1454.60	0.00	0 2 3	0 2 3	3.60	1400.50	0.00	1 2 3	1 2 3	8.10	1400.64	0.06	1 2 3	1 2 3	4.10
	C107	1455.11	0.16	0 2 3	0 2 3	3.70	1401.26	0.24	1 2 3	1 2 3	6.90	1400.64	0.25	1 2 3	1 2 3	3.90
	C108	1455.11	0.21	0 2 3	0 2 3	3.60	1401.26	0.38	1 2 3	1 2 3	7.70	1400.64	0.39	1 2 3	1 2 3	3.90
	C109	1455.11	0.30	0 2 3	0 2 3	3.20	1401.26	0.38	1 2 3	1 2 3	7.30	1395.23	0.19	1 2 3	1 2 3	4.50
	Average		**0.41**			**12.09 (29.20)**		**0.17**			**8.82 (101.60)**		**0.14**			**5.60 (197.30)**
	C101	1541.40	0.00	0 2 3	0 2 3	8.50	1479.02	0.38	1 2 3	1 2 3	6.75	1472.44	1.21	1 2 3	1 2 3	9.05
	C102	1509.61	0.03	0 2 3	0 2 3	87.85	1452.94	0.85	1 2 3	1 2 3	9.60	1459.43	1.13	1 2 3	1 2 3	10.70
	C103	1512.27	0.21	0 2 3	0 2 3	65.65	1440.27	0.09	1 2 3	1 2 3	11.65	1445.57	0.01	1 2 3	1 2 3	9.80
	C104	1521.01	0.56	0 2 3	0 2 3	13.30	1434.17	0.07	1 2 3	1 2 3	11.15	1442.28	**-0.08**	1 2 3	1 2 3	9.80
20	C105	1527.63	0.55	0 2 3	0 2 3	7.80	1455.00	0.00	1 2 3	1 2 3	7.85	1461.10	0.63	1 2 3	1 2 3	10.20
	C106	1541.40	0.00	0 2 3	0 2 3	8.40	1479.02	0.70	1 2 3	1 2 3	8.60	1473.32	1.47	1 2 3	1 2 3	9.20
	C107	1541.30	1.45	0 2 3	0 2 3	8.40	1462.04	0.48	1 2 3	1 2 3	6.60	1456.59	0.32	1 2 3	1 2 3	9.10
	C108	1543.95	1.62	0 2 3	0 2 3	8.00	1459.67	0.97	1 2 3	1 2 3	8.60	1461.86	1.00	1 2 3	1 2 3	9.80
	C109	1518.14	1.06	0 2 3	0 2 3	15.70	1435.90	0.05	1 2 3	1 2 3	8.80	1448.80	0.10	1 2 3	1 2 3	10.00
	Average		**0.61**			**24.84 (3331.30)**		**0.40**			**8.84 (3436.10)**		**0.64**			**9.74 (4501.90)**
	C101	1628.47	3.02	0 2 3	0 2 3	60.45	1490.99	2.70	1 2 3	1 2 3	2.00	1479.28	1.84	1 2 3	1 2 3	9.80
	C102	1615.40	2.21	0 2 3	0 2 3	16.10	1489.92	2.70	1 2 3	1 2 3	5.10	1475.34	1.79	1 2 3	1 2 3	11.70
	C103	1618.41	2.38	0 2 3	0 2 3	13.15	1481.03	1.41	1 2 3	1 2 3	8.40	1481.94	2.25	1 2 3	1 2 3	11.90
	C104	1609.61	2.26	0 2 3	0 2 3	22.60	1481.03	2.31	1 2 3	1 2 3	8.20	1471.13	1.50	1 2 3	1 2 3	11.60
25	C105	1626.81	3.34	0 2 3	0 2 3	11.05	1487.39	2.45	1 2 3	1 2 3	2.10	1462.41	0.68	1 2 3	1 2 3	10.30
	C106	1635.41	3.45	0 2 3	0 2 3	9.50	1490.99	2.70	1 2 3	1 2 3	2.00	1475.54	1.59	1 2 3	1 2 3	8.90
	C107	1625.88	3.28	0 2 3	0 2 3	13.10	1487.39	2.72	1 2 3	1 2 3	2.00	1473.62	1.45	1 2 3	1 2 3	10.00
	C108	1631.44	3.64	0 2 3	0 2 3	14.80	1485.99	2.62	1 2 3	1 2 3	2.30	1456.22	0.35	1 2 3	1 2 3	11.80
	C109	1603.49	1.86	0 2 3	0 2 3	13.40	1481.76	2.33	1 2 3	1 2 3	3.80	1479.94	2.10	1 2 3	1 2 3	10.90
	Average		**2.83**			**19.35 (3815.00)**		**2.44**			**3.99 (5231.70)**		**1.51**			**10.77 (4892.50)**
	C101	1632.25	2.13	0 2 3	0 2 3	15.95	1505.68	2.66	1 2 3	1 2 3	5.75	1474.37	0.41	1 2 3	1 2 3	10.45
	C102	1615.09	1.08	0 2 3	0 2 3	20.45	1504.60	2.65	1 2 3	1 2 3	9.35	1468.70	0.17	1 2 3	1 2 3	11.70
	C103	1606.49	0.22	0 2 3	0 2 3	71.00	1494.71	0.46	1 2 3	1 2 3	13.20	1468.70	**-1.46**	1 2 3	1 2 3	15.70
	C104	1611.77	0.47	0 2 3	0 2 3	116.20	1488.77	0.23	1 2 3	1 2 3	8.25	1467.40	0.00	1 2 3	1 2 3	18.65
30	C105	1615.88	1.58	0 2 3	0 2 3	133.30	1505.68	2.66	1 2 3	1 2 3	6.20	1489.55	1.45	1 2 3	1 2 3	11.85
	C106	1640.50	2.65	0 2 3	0 2 3	13.80	1506.28	2.70	1 2 3	1 2 3	5.40	1474.37	0.41	1 2 3	1 2 3	10.10
	C107	1619.09	1.78	0 2 3	0 2 3	13.20	1502.61	2.49	1 2 3	1 2 3	6.00	1474.37	0.41	1 2 3	1 2 3	9.60
	C108	1626.89	2.28	0 2 3	0 2 3	14.10	1502.61	2.50	1 2 3	1 2 3	6.30	1472.34	0.34	1 2 3	1 2 3	14.50
	C109	1610.98	1.27	0 2 3	0 2 3	13.50	1494.98	1.81	1 2 3	1 2 3	6.70	1473.37	0.41	1 2 3	1 2 3	11.40
	Average		**1.50**			**45.72 (6067.30)**		**2.02**			**7.46 (6658.30)**		**0.24**			**12.66 (7376.10)**
	Grand Average		**1.34**			**25.50**		**1.26**			**7.28**		**0.63**			**9.69**

According to Table 6.10, we get better results in R sets than C sets when the number of candidate satellites is equal to 2. Especially in instances with 25 and 30 customers, the performance is better in randomized sets. Even though random sets are challenging to solve, EA can overcome the complexities of the instances successfully. As the number of customers in the C set increases, the average deviation from the best increases rapidly, while the number of customers in R sets doubles, the gap does not increase twice. Although the facilities are opened incorrectly in 24 sets, deviations are low indicating that the routes and allocations are well built. In addition to 15 optimal solutions are reported, and the total cost of 10 best-known solutions is improved.

Table 6.10: Results for original Random (R) instances

		|J| = 2					|J| = 3					|J| = 4				
|K|	**Instance**	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s
	R101	1347.50	0.00	0 2 3	0 2 3	0.31	1347.50	0.00	0 2 4	0 2 4	0.29	1326.39	0.06	0 4 5	0 4 5	0.29
	R102	1313.90	0.00	0 2 3	0 2 3	0.28	1291.93	0.63	1 2 3	1 2 3	0.46	1282.71	0.06	1 3 4	1 3 4	12.79
	R103	1312.27	0.12	0 2 3	0 2 3	0.29	1291.93	0.63	1 2 3	1 2 3	0.43	1282.71	0.06	1 3 4	1 3 4	0.64
	R104	1282.20	0.00	0 2 3	0 2 3	0.38	1276.12	0.06	1 2 3	1 2 3	0.51	1273.80	0.00	1 3 4	1 3 4	6.75
	R105	1336.10	0.00	0 2 3	0 2 3	20.47	1326.90	0.07	1 2 3	1 2 3	0.48	1325.60	0.00	0 4 5	0 4 5	0.31
15	R106	1297.23	0.44	0 2 3	0 2 3	0.05	1292.26	0.96	0 2 4	1 2 3	0.78	1277.50	0.00	1 3 4	1 3 4	0.72
	R107	1297.23	0.44	0 2 3	0 2 3	0.36	1289.88	0.77	1 2 3	1 2 3	0.40	1278.23	0.06	1 3 4	1 3 4	0.62
	R108	1252.20	0.00	0 2 3	0 2 3	0.34	1252.20	0.00	0 2 4	0 2 4	0.33	1252.20	0.00	0 2 5	0 2 5	0.58
	R109	1299.65	0.06	0 2 3	0 2 3	0.26	1299.65	0.07	0 2 4	1 2 3	0.63	1291.10	0.00	0 4 5	0 4 5	0.41
	R110	1285.36	0.96	0 2 3	0 2 3	0.16	1279.72	0.52	1 2 3	0 2 4	40.55	1273.84	0.06	0 2 5	0 2 5	0.53
	R111	1287.40	0.00	0 2 3	0 2 3	0.34	1290.48	0.82	0 2 4	1 2 3	0.36	1274.72	0.06	1 3 4	1 3 4	0.41
	R112	1268.85	0.18	0 2 3	0 2 3	0.23	1268.85	0.18	0 2 4	0 2 4	0.27	1266.60	0.00	0 2 5	0 2 5	0.43
	Average		**0.18**			**1.96 (19.10)**		**0.39**			**3.79 (35.50)**		**0.03**			**2.04 (70.90)**
	R101	1464.11	0.07	0 2 3	0 2 3	3.04	1443.32	0.08	1 3 4	1 3 4	1.46	1432.67	0.10	0 4 5	0 4 5	1.26
	R102	1416.55	0.06	0 2 3	0 2 3	1.10	1373.19	0.09	1 2 3	1 2 3	0.61	1376.01	0.10	1 3 4	1 3 4	13.01
	R103	1353.64	0.06	0 2 3	0 2 3	36.66	1336.04	0.09	1 2 3	1 2 3	0.92	1318.99	0.08	1 3 4	1 3 4	1.30
	R104	1337.00	0.00	0 2 3	0 2 3	1.39	1334.27	1.76	1 2 3	1 2 3	1.34	1310.82	0.09	1 3 4	1 3 4	1.38
	R105	1399.69	0.06	0 2 3	0 2 3	0.54	1393.55	0.51	1 2 3	1 2 3	0.88	1383.13	0.07	0 4 5	0 4 5	0.82
20	R106	1384.15	0.27	0 2 3	0 2 3	0.18	1347.11	0.08	1 2 3	1 2 3	0.73	1354.60	0.43	1 3 4	1 3 4	1.11
	R107	1340.65	0.06	0 2 3	0 2 3	43.25	1344.80	1.66	1 2 3	1 2 3	1.01	1310.49	0.08	1 3 4	1 3 4	1.62
	R108	1331.84	0.65	0 2 3	0 2 3	0.62	1323.61	1.68	1 2 3	1 2 3	0.93	1297.52	0.08	1 3 4	1 3 4	1.44
	R109	1361.63	0.21	0 2 3	0 2 3	1.10	1355.38	0.24	1 2 3	1 2 3	0.93	1354.12	0.08	0 4 5	1 3 4	0.99
	R110	1331.76	0.06	0 2 3	0 2 3	0.61	1324.12	0.08	1 2 3	1 2 3	0.78	1329.22	0.08	0 4 5	0 4 5	0.66
	R111	1337.13	0.06	0 2 3	0 2 3	1.08	1337.13	0.79	0 2 4	1 2 3	1.30	1318.99	0.08	1 3 4	1 3 4	1.20
	R112	1309.60	0.06	0 2 3	0 2 3	0.71	1309.60	0.40	0 2 4	1 2 3	0.88	1308.24	0.07	0 4 5	0 4 5	1.29
	Average		**0.13**			**7.52 (859.60)**		**0.62**			**0.98 (1682.40)**		**0.11**			**2.17 (600.10)**
	R101	1587.69	0.07	0 2 3	0 2 3	2.09	1590.97	1.53	0 3 4	1 2 3	0.86	1585.53	1.25	1 3 4	0 4 5	1.14
	R102	1517.24	0.09	0 2 3	0 2 3	1.28	1524.99	2.60	0 3 4	1 2 3	1.58	1496.14	0.09	1 3 4	1 3 4	2.06
	R103	1466.24	1.04	0 2 3	0 2 3	1.23	1450.76	1.47	0 2 4	1 2 3	2.72	1429.62	**-1.49**	1 3 4	0 2 5	8.20
	R104	1415.43	0.07	0 2 3	0 2 3	2.95	1417.15	0.88	0 2 4	1 2 3	83.12	1420.02	**-0.22**	0 4 5	0 4 5	8.22
	R105	1508.60	0.57	0 2 3	0 2 3	0.89	1516.25	1.54	1 2 3	1 2 3	1.47	1485.03	0.09	0 4 5	0 4 5	7.60
25	R106	1445.41	0.07	0 2 3	0 2 3	0.36	1468.40	1.74	1 2 3	1 2 3	2.87	1467.14	1.57	0 4 5	0 2 5	10.38
	R107	1399.60	0.07	0 2 3	0 2 3	1.20	1404.78	0.08	1 2 3	1 2 3	7.96	1409.07	**-0.36**	1 3 4	0 4 5	3.28
	R108	1389.86	0.29	0 2 3	0 2 3	7.02	1390.41	**-0.06**	0 2 4	0 2 4	89.29	1390.57	0.34	1 3 4	0 2 5	1.92
	R109	1446.27	0.53	0 2 3	0 2 3	66.03	1451.38	2.29	1 3 4	1 2 3	1.84	1431.39	0.11	1 3 4	1 3 4	7.39
	R110	1425.44	0.24	0 2 3	0 2 3	1.09	1423.94	0.14	0 2 4	0 2 4	2.14	1421.40	0.09	0 4 5	1 3 4	8.01
	R111	1414.40	0.06	0 2 3	0 2 3	1.98	1416.22	0.99	0 2 4	1 2 3	70.54	1414.40	0.06	1 3 4	0 2 5	1.47
	R112	1381.53	0.49	0 2 3	0 2 3	0.89	1407.22	1.99	0 2 4	0 2 4	3.98	1388.26	0.98	0 4 5	0 2 5	33.35
	Average		**0.30**			**7.25 (7899.00)**		**1.27**			**22.36 (9496.10)**		**0.21**			**7.75 (8554.90)**
	R101	1619.59	0.10	0 2 3	0 2 3	10.05	1608.79	0.35	0 3 4	1 2 3	21.93	1610.93	0.12	0 4 5	0 4 5	3.52
	R102	1544.65	0.60	0 2 3	0 2 3	13.06	1528.81	0.94	0 3 4	1 2 3	9.61	1535.17	0.11	0 4 5	0 4 5	3.56
	R103	1459.73	0.67	0 2 3	0 2 3	3.31	1451.51	0.38	0 2 4	1 2 3	11.25	1451.51	0.10	0 2 5	0 2 5	4.77
	R104	1408.49	0.10	0 2 3	0 2 3	167.56	1408.49	**-0.92**	0 2 4	1 2 3	164.31	1408.49	**-1.01**	0 2 5	0 4 5	194.34
	R105	1579.96	2.17	0 2 3	0 2 3	4.26	1550.48	0.26	0 2 4	0 2 4	16.16	1540.96	0.11	0 4 5	0 4 5	8.36
30	R106	1497.85	1.17	0 2 3	0 2 3	29.90	1497.85	1.20	0 2 4	1 2 3	27.93	1480.42	0.15	0 4 5	1 3 4	4.34
	R107	1430.33	0.39	0 2 3	0 2 3	4.16	1430.33	0.39	0 2 4	0 2 4	11.97	1430.33	0.11	0 2 5	0 2 5	29.98
	R108	1392.12	**-1.02**	0 2 3	0 2 3	170.22	1385.70	**-0.33**	0 2 4	0 2 4	15.00	1392.12	**-1.25**	0 2 5	0 2 5	187.04
	R109	1462.19	0.38	0 2 3	0 2 3	11.86	1458.36	0.12	0 2 4	0 2 4	10.13	1458.36	0.12	0 2 5	0 2 5	12.67
	R110	1452.68	0.55	0 2 3	0 2 3	130.20	1446.15	0.09	0 2 4	0 2 4	5.64	1452.68	0.55	0 2 5	0 2 5	175.47
	R111	1437.10	0.23	0 2 3	0 2 3	3.22	1435.33	**-0.01**	0 2 4	0 2 4	5.95	1437.10	0.22	0 2 5	0 2 5	10.64
	R112	1386.56	0.10	0 2 3	0 2 3	2.74	1386.56	0.10	0 2 4	0 2 4	6.03	1386.56	0.10	0 2 5	0 2 5	13.48
	Average		**0.45**			**45.88 (3703.30)**		**0.21**			**25.49 (6820.60)**		**-0.05**			**54.01 (7386.10)**
	Grand Average		**0.27**			**15.65**		**0.62**			**13.16**		**0.07**			**16.49**

When the number of candidate satellites is two, RC instances can be called the instances where EA shows the lowest performance from the Table 6.11. While the number of customers was 25 and 30, solutions deviate a maximum of 9.38 percent from the best-known solution. Apart from that, as the number of candidate satellites increased, the average gap decreased, and the algorithm obtained the optimal solutions for 13 instances. We could not get the correct facility configuration in 6 sets, but the satellites opened are correct; only the CDCs are faulty. The fact that opening the wrong CDC when there are two candidate locations may indicate that the EA cannot obtain enough diversified solutions in tight sets.

Table 6.11: Results for original Random-Clustered (RC) instances

		|J| = 2					|J| = 3					|J| = 4				
|K|	Instance	Total Cost	Gap	F_{EA}	F_{BKS}	T_s	Total Cost	Gap	F_{EA}	F_{BKS}	T_s	Total Cost	Gap	F_{EA}	F_{BKS}	T_s
15	RC101	1720.53	0.75	0 2 3	0 2 3	0.30	1524.53	0.06	1 2 3	1 2 3	0.15	1554.79	0.06	1 2 3	1 2 3	0.17
	RC102	1606.42	0.06	0 2 3	0 2 3	0.28	1501.48	0.44	1 2 3	1 2 3	0.13	1534.86	0.32	1 2 3	1 2 3	6.29
	RC103	1606.42	0.06	0 2 3	0 2 3	0.29	1501.48	0.44	1 2 3	1 2 3	0.17	1534.86	0.32	1 2 3	1 2 3	18.55
	RC104	1627.04	1.75	0 2 3	0 2 3	0.63	1493.50	0.32	1 2 3	1 2 3	6.33	1523.60	0.00	1 2 3	1 2 3	0.24
	RC105	1649.81	0.19	0 2 3	0 2 3	0.47	1533.31	0.19	1 2 3	1 2 3	0.15	1577.81	0.16	1 2 3	1 2 3	6.32
	RC106	1639.63	2.02	0 2 3	0 2 3	0.05	1503.42	0.06	1 2 3	1 2 3	6.33	1538.55	0.07	1 2 3	1 2 3	0.14
	RC107	1594.90	0.00	0 2 3	0 2 3	7.05	1483.50	0.00	1 2 3	1 2 3	0.17	1518.40	0.00	1 2 3	1 2 3	12.34
	RC108	1588.20	0.00	0 2 3	0 2 3	0.28	1483.81	0.12	1 2 3	1 2 3	0.16	1518.62	0.26	1 2 3	1 2 3	0.25
	Average		**0.60**			**1.17 (56.50)**		**0.20**			**1.70 (419.60)**		**0.15**			**5.54 (237.00)**
20	RC101	1822.19	5.06	0 2 3	0 2 3	25.75	1622.66	0.82	1 2 3	1 2 3	0.81	1610.44	0.06	1 3 4	1 3 4	0.35
	RC102	1754.63	2.60	0 2 3	0 2 3	2.10	1581.05	0.06	1 2 3	1 2 3	0.89	1587.27	0.06	1 3 4	1 3 4	0.65
	RC103	1736.16	2.13	0 2 3	0 2 3	0.76	1577.99	0.06	1 2 3	1 2 3	1.04	1587.41	0.07	1 3 4	1 3 4	0.74
	RC104	1710.88	1.18	0 2 3	0 2 3	41.82	1572.29	0.09	1 2 3	1 2 3	0.78	1579.97	0.23	1 3 4	1 3 4	0.35
	RC105	1808.70	4.49	0 2 3	0 2 3	0.67	1613.02	0.71	1 2 3	1 2 3	0.74	1610.09	0.07	1 3 4	1 3 4	1.09
	RC106	1713.71	0.43	0 2 3	0 2 3	0.18	1596.86	0.58	1 2 3	1 2 3	0.58	1592.06	0.07	1 3 4	1 3 4	0.24
	RC107	1722.49	2.54	0 2 3	0 2 3	0.67	1567.30	0.00	1 2 3	1 2 3	0.51	1571.04	0.06	1 3 4	1 3 4	0.81
	RC108	1720.78	2.71	0 2 3	0 2 3	0.68	1565.30	0.00	1 2 3	1 2 3	0.78	1566.90	0.00	1 3 4	1 3 4	0.65
	Average		**2.64**			**9.08 (340.60)**		**0.29**			**0.77 (3835.50)**		**0.08**			**0.61 (3073.80)**
25	RC101	1946.38	6.32	0 2 3	0 2 3	1.76	1726.97	3.37	1 2 3	1 2 3	6.70	1650.98	0.07	1 3 4	1 3 4	7.21
	RC102	1865.02	6.93	0 2 3	0 2 3	1.18	1635.37	1.27	1 2 3	1 2 3	1.12	1621.49	0.07	1 3 4	1 3 4	7.48
	RC103	1795.48	3.65	1 2 3	0 2 3	1.83	1609.84	0.33	1 2 3	1 2 3	1.66	1609.28	0.09	1 3 4	1 3 4	1.66
	RC104	1773.50	3.93	1 2 3	0 2 3	60.36	1593.23	1.04	1 2 3	1 2 3	1.34	1582.07	0.24	1 3 4	1 3 4	1.72
	RC105	1922.24	9.38	1 2 3	0 2 3	1.43	1715.12	5.47	1 2 3	1 2 3	1.54	1630.03	0.08	1 3 4	1 3 4	7.76
	RC106	1868.21	7.43	1 2 3	0 2 3	0.36	1680.09	4.35	1 2 3	1 2 3	1.61	1613.62	0.06	1 3 4	1 3 4	7.04
	RC107	1826.40	7.38	1 2 3	0 2 3	1.62	1628.16	3.23	1 2 3	1 2 3	1.08	1581.50	0.00	1 3 4	1 3 4	13.41
	RC108	1756.86	3.46	0 2 3	0 2 3	1.71	1577.98	0.18	1 2 3	1 2 3	14.26	1576.80	0.00	1 3 4	1 3 4	31.57
	Average		**6.06**			**8.78 (4735.10)**		**2.40**			**3.66 (3522.70)**		**0.08**			**9.73 (4875.10)**
30	RC101	1981.99	2.03	0 2 3	0 2 3	1.07	1952.66	2.84	1 2 3	1 2 3	1.89	1885.88	0.07	1 3 4	1 3 4	9.06
	RC102	1934.00	4.23	0 2 3	0 2 3	2.32	1857.15	0.59	1 2 3	1 2 3	2.58	1829.66	0.43	0 4 5	0 4 5	3.07
	RC103	1877.07	4.68	0 2 3	0 2 3	98.36	1746.42	0.96	1 2 3	1 2 3	2.91	1731.26	0.07	1 3 4	1 3 4	8.83
	RC104	1822.65	3.02	0 2 3	0 2 3	3.72	1686.38	0.66	1 2 3	1 2 3	14.98	1701.80	0.00	1 2 4	1 2 4	3.05
	RC105	1925.96	4.71	0 2 3	0 2 3	2.15	1799.41	2.13	1 2 3	1 2 3	3.26	1764.90	0.07	1 3 4	1 3 4	3.50
	RC106	1872.23	3.71	0 2 3	0 2 3	1.14	1739.98	0.10	1 2 3	1 2 3	2.45	1743.86	0.23	1 3 4	1 3 4	2.00
	RC107	1833.35	3.83	0 2 3	0 2 3	1.53	1720.60	1.22	1 2 3	1 2 3	2.29	1699.40	0.00	1 3 4	1 3 4	2.55
	RC108	1812.10	2.79	1 2 3	0 2 3	1.79	1684.92	0.04	1 2 3	1 2 3	2.89	1694.70	0.00	1 3 4	1 3 4	7.86
	Average		**3.62**			**14.01 (7752.30)**		**1.07**			**4.16 (5470.60)**		**0.11**			**4.99 (8403.60)**
	Grand Average		**3.23**			**8.26**		**0.99**			**2.57**		**0.10**			**5.22**

Unfortunately, we cannot directly compare the number of facilities in the tables created for each customer size for Set 2 instances because candidate CDC and satellite pairs are 2-3, 6-4, and 3-5. Nevertheless, we can directly see the increase in complexity due to combinations of these facilities. In 60 sets having 15 customers (Table 6.12), facility configuration of only 7 sets is found wrong, which caused deviation to be high for those sets. While we can obtain the optimal solutions for 46 instances, we have obtained the least average deviation when candidate satellite points are 5.

Table 6.12: Set 2 results for original instances having 15 customers

	|I| = 2, |J| = 3					|I| = 6, |J| = 4					|I| = 3, |J| = 5				
Instance	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s
15 a1	1652.27	0.05	0 2 3	0 2 3	165.75	1639.51	0.00	3 7 8	3 7 8	0.64	1648.70	0.00	1 3 5	1 3 5	0.46
15 a2	1641.38	0.00	0 3 4	0 3 4	0.37	1661.83	0.23	3 7 8	5 6 9	0.54	1690.83	0.00	1 5 6	1 5 6	0.64
15 a3	1615.53	0.00	0 2 3	0 2 3	0.33	1613.59	0.61	3 8 9	1 6 7	12.71	1585.63	0.00	2 6 7	2 6 7	262.89
15 a4	1660.76	0.00	0 2 4	0 2 4	1.08	1531.44	0.00	3 7 8	3 7 8	0.57	1598.90	0.00	1 3 6	1 3 6	233.13
15 a5	1654.34	0.00	1 2 3	1 2 3	0.33	1526.17	0.00	1 6 7	1 6 7	18.76	1551.18	0.00	2 5 7	2 5 7	6.76
Average		**0.01**			**33.57 (3.06)**		**0.17**			**6.64 (9.37)**		**0.00**			**100.68 (11.24)**
15 b1	1660.36	0.00	0 2 3	0 2 3	183.33	1660.98	1.08	3 7 8	3 7 8	18.81	1619.59	0.00	1 5 6	1 5 6	7.18
15 b2	1660.82	0.03	1 3 4	0 3 4	0.45	1684.67	0.00	3 7 8	3 7 8	6.60	1672.19	0.00	2 3 6	2 3 6	43.08
15 b3	1675.17	0.00	0 2 3	0 2 3	0.37	1659.04	0.69	4 7 9	3 8 9	0.61	1587.83	0.00	2 5 6	2 5 6	49.31
15 b4	1646.91	0.00	1 3 4	1 3 4	0.37	1592.01	0.00	3 7 8	3 7 8	1.00	1520.98	0.04	1 5 6	1 5 6	37.30
15 b5	1638.46	0.77	1 2 3	1 3 4	0.41	1548.37	0.00	1 6 7	1 6 7	0.38	1617.81	0.00	2 5 7	2 5 7	0.72
Average		**0.16**			**36.99 (8.19)**		**0.36**			**5.48 (9.33)**		**0.01**			**27.52 (24.64)**
15 c1	1639.04	0.00	0 2 3	0 2 3	0.32	1645.20	0.00	3 7 8	3 7 8	0.56	1550.92	0.00	1 3 5	1 3 5	18.87
15 c2	1619.16	0.00	0 3 4	0 3 4	0.37	1626.22	0.00	5 6 9	5 6 9	16.89	1591.30	0.21	2 3 7	2 3 6	49.33
15 c3	1647.18	0.48	0 2 3	0 2 3	0.27	1640.59	0.38	2 7 8	1 6 7	0.81	1580.80	0.00	2 6 7	2 6 7	153.74
15 c4	1636.43	0.00	1 3 4	1 3 4	0.39	1583.55	0.00	3 7 8	3 7 8	0.54	1590.89	0.04	1 5 6	1 5 6	13.00
15 c5	1552.93	0.00	1 3 4	1 3 4	937.40	1520.02	0.00	1 6 7	1 6 7	12.60	1598.56	0.00	2 5 7	2 5 7	18.88
Average		**0.10**			**187.75 (6.27)**		**0.08**			**6.28 (12.19)**		**0.05**			**50.77 (20.65)**
15 d1	1649.17	0.00	0 2 3	0 2 3	12.80	1633.29	0.00	3 7 8	3 7 8	0.47	1641.84	0.00	1 3 5	1 3 5	202.90
15 d2	1588.20	0.00	0 3 4	0 3 4	0.93	1631.59	0.00	5 6 9	5 6 9	6.72	1615.98	0.40	2 6 7	2 6 7	19.02
15 d3	1622.36	0.00	0 2 3	0 2 3	0.35	1654.34	0.00	3 8 9	3 8 9	0.54	1592.27	0.00	2 6 7	2 6 7	96.08
15 d4	1649.35	0.00	0 2 4	0 2 4	1.15	1598.46	0.00	3 7 8	3 7 8	0.50	1593.36	0.31	1 5 6	1 5 6	6.67
15 d5	1636.30	0.00	1 2 3	1 2 3	258.07	1520.02	0.00	1 6 7	1 6 7	18.70	1603.16	0.00	2 5 7	2 5 7	6.65
Average		**0.00**			**54.66 (2.50)**		**0.00**			**5.39 (8.25)**		**0.14**			**66.26 (10.14)**
Grand Average		**0.07**			**78.24**		**0.15**			**5.95**		**0.05**			**61.33**

Table 6.13 shows that sets with 30 customers, the EA achieved 19 times the optimal solution and the worst gap achieved for a set is 1.48%. Increasing or decreasing the problem size in terms of facilities has not significantly increased the gap since the averages are close to each other. The facility configuration of 8 sets are found incorrectly, which caused the increase in deviation, and we obtained the maximum deviation because we opened the facilities incorrectly.

One of this study's main objectives is to obtain optimal or near-optimal results in a reasonable time limit for large sets. Because successful exact solution methods for small sets are already available in the literature, these methods cannot obtain an optimal solution for large-sized instances. Usually, the exact methods terminate because the time limit is reached. It would be more reasonable to comment on the success of the algorithm for the sets having 50 or 100 customers.

Table 6.13: Set 2 results for original instances having 30 customers

	|I| = 2, |J| = 3					|I| = 6, |J| = 4					|I| = 3, |J| = 5				
Instance	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s
30 a1	2122.91	0.03	1 2 4	1 2 4	76.30	2179.31	0.52	1 6 7	1 6 7	3.86	2182.52	0.14	0 3 7	1 5 6	5.04
30 a2	2131.49	0.14	1 3 4	1 3 4	3.15	2156.95	0.08	2 7 8	2 7 8	6.67	2170.85	0.00	0 4 5	0 4 5	3.28
30 a3	2224.39	1.03	1 2 4	1 2 4	20.78	2123.42	0.66	2 7 8	4 8 9	4.72	2116.41	0.00	1 4 5	1 4 5	44.11
30 a4	2142.20	0.29	0 2 3	0 2 3	3.22	2115.38	0.05	2 7 8	2 7 8	6.66	2132.19	0.11	1 5 6	2 6 7	3.09
30 a5	2213.37	0.00	1 3 4	1 3 4	3.09	2131.11	0.00	1 6 7	1 6 7	2.57	2076.96	0.04	2 6 7	2 6 7	4.26
Average		**0.30**			**21.31 (22.76)**		**0.26**			**4.89 (49.68)**		**0.06**			**11.95 (56.07)**
30 b1	2122.20	0.15	1 2 4	1 2 4	9.23	2207.84	0.41	1 6 7	1 6 7	4.02	2216.22	1.24	1 5 6	1 5 6	2.68
30 b2	2018.18	0.69	1 3 4	1 3 4	3.27	2190.98	0.48	2 7 8	4 8 9	3.31	2194.01	0.12	1 5 6	1 5 6	5.03
30 b3	2214.19	0.42	1 2 4	1 2 4	6.29	2161.29	0.11	2 7 8	2 7 8	3.61	2120.01	0.00	1 4 5	1 4 5	21.20
30 b4	2179.95	0.96	0 2 3	0 2 3	1.90	2131.86	0.48	2 7 8	2 7 8	10.82	2157.80	1.48	1 5 6	2 6 7	36.34
30 b5	1910.46	0.85	0 3 4	1 3 4	6.33	2168.05	0.24	1 6 7	1 6 7	4.51	2155.34	0.20	2 6 7	2 6 7	4.18
Average		**0.61**			**5.40 (88.62)**		**0.35**			**5.25 (110.28)**		**0.61**			**13.88 (63.36)**
30 c1	1799.56	0.00	1 2 4	1 2 4	9.86	2128.65	0.38	1 6 7	1 6 7	4.46	2209.80	0.00	1 5 6	1 5 6	3.55
30 c2	2098.61	0.00	1 3 4	1 3 4	3.85	2137.77	0.00	4 8 9	4 8 9	3.85	2117.71	0.06	1 5 6	1 5 6	4.67
30 c3	2149.85	0.00	1 2 4	1 2 4	8.36	2114.25	0.00	2 7 8	2 7 8	4.99	2154.62	0.00	1 4 5	1 4 5	2.44
30 c4	2128.30	0.00	0 2 3	0 2 3	2.15	2087.60	0.12	2 7 8	2 7 8	2.65	2110.28	0.84	2 6 7	2 6 7	14.63
30 c5	2179.26	0.00	1 3 4	1 3 4	2.82	1926.64	0.02	5 7 9	5 7 9	4.78	1797.05	0.08	2 6 7	2 6 7	6.39
Average		**0.00**			**5.41 (51.47)**		**0.10**			**4.15 (114.05)**		**0.19**			**6.34 (128.59)**
30 d1	1809.62	0.13	1 2 4	1 2 4	7.53	2175.58	0.46	1 6 7	1 6 7	5.42	2227.50	0.00	1 5 6	1 5 6	2.67
30 d2	2108.97	0.00	1 3 4	1 3 4	3.27	2155.12	0.33	4 8 9	4 8 9	80.66	2193.80	0.35	1 5 6	1 5 6	7.00
30 d3	2174.01	0.52	1 2 4	1 2 4	10.55	2138.62	0.43	2 7 8	4 8 9	3.00	2121.40	0.20	1 4 5	1 4 5	8.99
30 d4	2137.98	0.00	0 2 3	0 2 3	1.78	2105.15	0.00	2 7 8	2 7 8	5.46	2126.29	0.50	2 6 7	2 6 7	8.68
30 d5	2199.51	0.09	0 3 4	1 3 4	2.74	2197.54	0.00	1 6 7	1 6 7	40.81	2102.72	0.00	2 6 7	2 6 7	5.21
Average		**0.15**			**5.17 (10.08)**		**0.24**			**27.07 (47.97)**		**0.21**			**6.51 (46.77)**
Grand Average		**0.26**			**9.32**		**0.24**			**10.34**		**0.27**			**9.67**

Table 6.14: Set 2 results for original instances having 50 customers

	|I| = 2, |J| = 3					|I| = 6, |J| = 4					|I| = 3, |J| = 5				
Instance	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s
50 a1	2641.49	0.07	0 2 3	0 2 3	23.84	2588.62	0.28	5 7 9	5 7 9	20.36	2597.86	0.38	2 3 6	2 3 6	14.49
50 a2	2708.80	0.58	0 2 4	0 2 4	225.12	2596.65	0.00	5 6 8	5 6 8	12.96	2610.86	0.31	1 3 6	1 3 6	16.76
50 a3	2611.83	0.46	1 3 4	1 3 4	8.04	2591.65	0.48	2 6 7	2 6 7	10.77	2401.34	0.07	1 4 5	1 4 5	16.07
50 a4	2565.10	0.18	0 2 3	0 2 3	10.47	2496.21	0.21	5 6 9	5 6 9	256.90	2497.90	0.04	2 6 7	2 6 7	26.90
50 a5	2572.68	0.95	0 2 3	1 2 4	10.30	2591.48	0.32	2 6 7	5 6 9	241.84	2521.81	0.43	1 5 6	1 5 6	17.83
Average		**0.45**			**55.56 (284.37)**		**0.26**			**108.26 (656.99)**		**0.25**			**18.41 (274.09)**
50 b1	2641.58	1.17	0 2 3	0 2 3	10.01	2582.60	0.59	5 6 9	5 6 9	224.57	2577.31	**-0.02**	2 3 6	2 3 6	24.91
50 b2	2590.38	0.44	0 2 4	0 2 4	239.78	2686.55	1.08	5 6 8	5 6 8	19.31	2668.50	1.65	0 3 6	1 3 6	25.62
50 b3	2651.36	0.82	1 3 4	1 3 4	12.80	2611.27	0.56	2 6 7	2 6 7	10.71	2528.28	0.63	1 4 6	1 4 6	22.19
50 b4	2589.53	0.34	0 2 3	0 2 3	8.98	2517.19	0.14	5 6 9	5 6 9	77.22	2523.95	0.14	2 6 7	2 6 7	11.60
50 b5	2631.93	1.49	0 2 3	1 2 4	10.23	2625.55	0.32	2 6 7	3 7 9	47.83	2569.06	1.55	1 5 7	1 5 6	18.74
Average		**0.85**			**56.36 (524.78)**		**0.54**			**75.93 (1047.52)**		**0.79**			**20.61 (3279.02)**
50 c1	2560.79	0.60	0 2 3	0 2 3	9.32	2526.86	0.58	5 6 9	5 6 9	240.75	2522.93	0.20	1 4 5	1 4 5	12.58
50c2	2611.15	0.83	0 2 4	1 2 3	251.42	2518.44	0.01	5 6 8	5 6 8	11.76	2579.28	0.55	0 3 6	1 3 6	22.90
50 c3	2530.33	0.26	1 3 4	1 3 4	11.03	2588.34	0.34	1 6 8	5 8 9	12.95	2480.69	0.09	1 4 6	1 4 6	19.56
50 c4	2521.91	0.22	0 2 3	0 2 3	9.02	2424.16	0.46	5 6 9	5 6 9	267.52	2592.14	1.18	2 6 7	2 6 7	21.07
50 c5	2560.52	0.36	1 2 4	1 2 4	273.47	2751.60	0.33	0 6 8	2 6 7	396.66	2587.51	0.80	1 5 7	2 3 7	11.68
Average		**0.45**			**110.85 (1870.09)**		**0.34**			**185.93 (2397.18)**		**0.56**			**17.56 (5792.74)**
50 d1	2623.18	1.28	1 3 4	1 3 4	31.89	2532.33	0.37	5 6 9	5 6 9	243.20	2562.80	0.25	2 3 6	2 3 6	21.33
50 d2	2699.18	0.71	0 2 4	1 2 3	223.59	2570.20	0.08	5 6 8	5 6 8	27.71	2593.96	0.28	1 3 6	1 3 6	13.06
50 d3	2595.95	0.14	1 3 4	1 3 4	14.84	2656.33	0.71	1 6 8	5 8 9	14.94	2497.37	0.21	1 4 5	1 4 5	11.89
50 d4	2573.86	0.41	0 2 3	0 2 3	7.44	2499.68	0.35	5 6 9	5 6 9	241.40	2530.86	0.58	2 6 7	2 6 7	13.82
50 d5	2648.20	0.70	1 2 4	1 2 4	221.07	2615.29	0.86	3 7 8	3 7 8	34.70	2577.90	0.68	1 5 7	1 5 6	11.55
Average		**0.65**			**99.76 (210.68)**		**0.48**			**112.39 (359.94)**		**0.40**			**14.33 (872.32)**
Grand Average		**0.60**			**80.63**		**0.40**			**120.70**		**0.50**			**17.73**

We report a better total cost for a set for which optimal results cannot be achieved even though we only obtain an optimal result once for mid-sized instances in Table 6.14. Besides, the average gaps are below 1%, and even the maximum deviation is less than 2%. Considering the first three tables, one can see that percent deviations have increased. The important thing here is how much it has increased since the gap we reported for mid-sized instances indicates good performance.

Table 6.15: Set 2 results for original instances having 100 customers

	|I| = 2, |J| = 3					**|I| = 6, |J| = 4**					**|I| = 3, |J| = 5**				
Instance	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s	**Total Cost**	**Gap**	F_{EA}	F_{BKS}	T_s
100 a1	4119.06	0.70	1 3 4	1 3 4	56.72	4009.87	0.45	4 6 8	4 8 9	48.80	4129.46	**-1.40**	0 4 7	0 4 7	102.008
100 a2	4252.81	0.90	0 2 4	0 2 4	940.36	3920.73	**-1.03**	5 6 9	5 6 9	1008.28	3947.19	1.41	0 3 6	0 3 6	66.392
100 a3	3969.28	1.25	1 2 4	1 2 4	953.43	4236.65	1.22	3 6 8	3 8 9	112.05	3932.29	1.37	0 3 5	2 5 7	57.194
100 a4	4265.79	1.15	1 2 3	1 2 3	50.83	3891.47	**-0.57**	0 7 9	0 6 9	85.26	4068.29	2.41	0 3 4	0 3 4	42.753
100 a5	4104.63	1.91	0 3 4	0 2 3	51.19	4133.07	1.31	5 6 9	5 6 9	977.95	3968.88	0.01	1 4 5	1 4 5	35.012
Average		**1.18**			**410.51 (7213.22)**		**0.28**			**446.47 (13093.50)**		**0.76**			**60.67 (11510.37)**
100 b1	3937.78	**-0.45**	1 3 4	1 3 4	45.93	3932.28	**-0.05**	4 6 8	4 6 7 9	51.01	4199.37	1.06	2 5 7	2 5 7	38.885
100 b2	4257.19	3.54	1 2 3 4	0 2 4	514.00	4010.54	0.67	5 6 9	5 6 9	1001.83	3966.02	1.39	2 3 6	0 3 6	107.609
100 b3	4062.12	3.00	1 2 4	1 2 4	957.31	4207.36	**-0.72**	3 6 8	2 6 8	83.82	3996.05	**-3.24**	2 5 7	2 5 7	46.798
100 b4	4185.41	0.76	1 2 3	1 2 3 4	51.03	4262.56	2.59	5 7 9	0 6 9	48.69	4166.72	3.48	0 3 4	0 3 4	33.243
100 b5	4096.86	1.29	0 2 3	0 2 3	102.27	4201.07	1.46	3 7 8	5 6 9	56.37	4147.47	**-3.01**	1 4 5	1 4 5	61.202
Average		**1.61**			**334.11 (12114.35)**		**0.79**			**248.34 (14273.36)**		**-0.06**			**57.55 (14256.46)**
100 c1	4095.42	**-2.31**	1 3 4	1 3 4	37.99	3957.32	**-1.16**	4 6 8	4 6 8	68.13	4117.89	**-1.00**	0 4 7	0 3 5	98.38
100 c2	4063.48	1.02	0 2 4	0 2 4	1028.80	4077.44	**-0.62**	5 6 9	5 6 9	1008.90	3897.82	**-1.47**	0 3 6	0 3 6	101.425
100 c3	3985.15	1.60	1 2 4	1 2 4	974.28	3931.27	**-2.02**	3 8 9	3 6 8 9	41.46	4006.82	**-1.02**	2 5 7	0 3 5	34.123
100 c4	4131.61	**-1.23**	1 2 3	1 3 4	42.85	4013.79	**-2.22**	5 7 9	5 7 9	70.94	4024.78	1.69	0 3 4	0 3 4	40.941
100 c5	4064.80	0.87	0 2 3	0 2 3	61.89	4016.47	**-1.10**	3 7 8	5 6 9	59.76	3988.20	**-1.89**	1 4 5	1 4 5	47.298
Average		**-0.01**			**429.16 (14352.24)**		**-1.42**			**249.84 (14361.00)**		**-0.74**			**64.43 (14322.84)**
100 d1	4007.87	**-1.13**	1 3 4	0 2 4	43.42	4081.35	1.19	4 6 8	4 8 9	66.32	4136.54	0.76	1 5 7	1 5 7	58.993
100 d2	4270.99	1.72	0 2 4	0 2 4	946.55	4004.21	1.13	5 6 9	5 6 9	968.87	3917.35	0.79	0 3 6	0 3 6	91.952
100 d3	3938.98	1.99	1 2 4	1 2 4	996.82	4005.73	2.13	3 6 8	3 6 8 9	82.16	3964.47	**-1.36**	2 5 7	2 5 7	56.332
100 d4	4152.06	1.11	1 2 3	1 2 4	36.22	4134.62	0.99	5 7 9	0 6 9	52.88	3978.57	1.71	1 4 6	0 3 4	61.088
100 d5	4120.58	1.98	0 3 4	0 2 3	40.92	4131.44	1.27	5 6 9	5 6 9	950.47	4043.00	0.82	1 4 5	1 4 5	40.744
Average		**1.14**			**412.79 (8626.84)**		**1.34**			**424.14 (10600.14)**		**0.54**			**61.82 (13357.63)**
Grand Average		**0.98**			**396.64**		**0.25**			**342.20**		**0.13**			**61.1186**

In this EA, which works fast enough, the solution time is mostly affected by whether the population converges or not. The nature of the data in the set as well as the size, affects convergence. For instance, a shorter time window indicates that many candidate solutions can become infeasible, resulting in less diversity in the population due to the tight constraint. In contrast, a larger time window means that more feasible solutions are possible, and a variety of possibilities increase the number of different solutions that prolong the duration of convergence. Therefore, some of the instances in Table 6.15 duration is too long because the population does not achieve stopping criteria; the algorithm is continued until the maximum number of generations.

For the 17 instances specified as optimal with 100 customers, the maximum gap 3% given by EA and the facility configuration of 7 instances are wrong. It is not convenient to compare the facilities for the sets specified as best known because different facilities may have been opened in optimal answers. According to Table 6.15, we updated the best-known ones by obtaining lower total costs for 21 instances. Additionally, we observed that as instance size increased, the average deviation decreased.

This EA, which is more successful in Set 2 instances, produces very fast results and can obtain the optimal solution. Although the EA cannot achieve the optimal solution for larger sets, it can improve the best-known solution and shows that differences between the best-known and our solutions are too small. One reason to obtain lower success in Set 1 is that those sets are not created by considering the 2E-LRP structure. Nonetheless, Set 2 instances are generated to represent the unique structure of the introduced problem.

CHAPTER 7

CONCLUSION AND FUTURE WORKS

The rapid growth of the population resulted in more demands for products and more freight movement. In order to overcome the adverse consequences of the increased freight movements, an effective and efficient design of logistics network models is required. Therefore, the two-echelon location routing problem with hard time windows is studied in this thesis. The 2E-LRPTW is a problem that deals with strategic and tactical level decisions that concern stakeholders with different objectives of the system.

The 2E-LRPTW problem decides which facilities to open at both echelons, namely CDC and satellite location decisions and resulting routes from each open facility. In our distribution network design CDCs, satellites, and customers are connected through a two-layered system using two types of vehicle fleets. Customers in the system have hard time windows indicating that the service must be completed within the given time intervals, or the system is not feasible. Capacity restrictions are imposed on facilities and vehicles at both echelons.

The two-layered network modeled as a three-index MIP inspired the work of [15] and we proposed an evolutionary algorithm based on a genetic algorithm considering complexities while adopting a solution procedure to a mathematical model with capacities and hard time windows. The proposed EA does not allow infeasible solutions and tries to achieve results with good quality by evolving over generations with problem-specific operators. While designing the algorithm, we first worked on models with a 2E-LRPTW structure having open CDCs in the first echelon. Consequently, there are only satellite location decisions in the second echelon. Fewer decision variables reduce the complexity of the problem but still have hard time windows and facility capacities.

The EA generates the initial population combining randomness and heuristics, obtaining better offsprings from the selected individuals applying reproduction over generations. In order to sustain diversity and intensity, two mutations are proposed to improve the routes and reach different points in the solution space by changing the customers' assignments. Later, we modified this algorithm to solve the original problem having a two-echelon structure. After the arrangements made in the initialization and reproduction stages for locating CDCs and satellite allocations, the algorithm can now solve the original problem.

We solve the introduced Set 1 and Set 2 instances for both one-echelon and two-echelon versions separately. The results of Set 1 instances demonstrate that the instance properties are highly effective on the solution performance. For instance, if time windows are tight, then the percent deviation from the optimal (best-known) solution gets higher. As expected, customer spatial dispersion affects the performance since constructing routes for clustered sets considered easier. However, we can report fewer percent gaps for randomly distributed customers so the algorithm can capture points in the diverse space. The average deviation reported for Set 1 instances is 0.8% which can be considered a good indication for a metaheuristic performance.

The reason for sometimes not achieving optimal is that the instances have too many tight constraints and hard time windows. It is difficult for an exact method to handle such constraints and the metaheuristic method. The smallest feature or facility capacity structure or time windows change affects both the algorithm duration and solution quality. In this case, constructing a robust metaheuristic has its complexities.

Considering all these results, we can say that the proposed EA successfully finds the correct satellite configurations. Because different configurations are created due to randomness while opening satellites and the facility array crossover phase also contributes to this success. Even when the satellites are opened incorrectly, deviations are low since the algorithms to establish and improve routes are well-known successful heuristics. The power of allocation decisions is decisive, and the 'Best' parent selection and assignment processes are successful, but it is difficult to achieve success in every set. As far as we know, we are the second study addressing 2E-LRPTW in the literature, and we are the first study to propose a population-based metaheuristic in terms of the solution method. The proposed EA can handle time windows and capacity constraints in a promising way.

In order to advance our work, metaheuristic studies that experiment with new initialization or crossover techniques can be constructed. During our experiments, it is highly noticed that the allocation stage is the main factor of the deviations, and in literature, allocations are mostly done either random or favoring demand order to assign the closest facility. However, in these problems, there are no hard time windows, which significantly affects the assignments. Solution methods that focus the allocation stage for the problems having hard time windows and facility capacities can be considered. Metaheuristics embedded with optimization techniques can be a future research area to improve both solution quality and time.

Another future research direction can take factors of uncertainty into account. While constructing the mathematical model, we are assuming that the customer demands and travel times are constant. To improve the distribution network systems' capability to represent real-life constraints, customers' demand uncertainty can be considered. Another research can reflect the travel time uncertainty since not only distance but congestion and unexpected events affect the arrival time to customers.

REFERENCES

[1] M. Amiri, S. H. Amin, and R. Tavakkoli-Moghaddam. A Lagrangean decomposition approach for a novel two-echelon node-based location-routing problem in an offshore oil and gas supply chain. *Transportation Research Part E: Logistics and Transportation Review*, 128(March):96–114, 2019.

[2] D. Beasley, D. R. Bull, and R. R. Martin. An overview of genetic algorithms : Part 1, fundamentals. *University Computing*, 2(15):1–16, 1993.

[3] I. Ben Mohamed, W. Klibi, and F. Vanderbeck. Designing a two-echelon distribution network under demand uncertainty. *European Journal of Operational Research*, 280(1):102–123, 2020.

[4] J. Berger and M. Barkaoui. A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers and Operations Research*, 31(12):2037–2053, 2004.

[5] M. Boccia, T. G. Crainic, A. Sforza, and C. Sterle. A metaheuristic for a two echelon location-routing problem. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 6049 LNCS:288–301, 2010.

[6] U. Breunig, V. Schmid, R. F. Hartl, and T. Vidal. A large neighbourhood based heuristic for two-echelon routing problems. *Computers and Operations Research*, 76:208–225, 2016.

[7] C. Contardo, V. Hemmelmayr, and T. G. Crainic. Lower and upper bounds for the two-echelon capacitated location-routing problem. *Computers and Operations Research*, 39(12):3185–3199, 2012.

[8] J.-f. Cordeau, G. Laporte, and A. Mercier. 2001-A unified tabu search heuristic for vehicle routing problems with time windows.pdf. pages 928–936, 2001.

[9] T. Crainic, A. Sforza, and C. Sterle. Location-Routing Models for Two-Echelon Freight Distribution System Design Location-Routing Models for Two-Echelon Freight. *CIRRELT Working Paper*, page 28, 2011.

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

[11] G. A. Croes. A Method for Solving Traveling-Salesman Problems. *Operations Research*, 6(6):791–812, 1958.

[12] M. Darvish, C. Archetti, L. C. Coelho, and M. G. Speranza. Flexible two-echelon location routing problem. *European Journal of Operational Research*, 277(3):1124–1136, 2019.

[13] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization : NSGA-II.

[14] H. Derbel, B. Jarboui, S. Hanafi, and H. Chabchoub. Genetic algorithm with iterated local search for solving a location-routing problem. *Expert Systems with Applications*, 39(3):2865–2871, 2012.

[15] M. S. FARHAM. *Location-routing and synchronization problems in city logistics*. Phd thesis, Middle East Technical University, 2020.

[16] V. C. Hemmelmayr and T. Gabriel. Computers & Operations Research An adaptive large neighborhood search heuristic for Two-Echelon Vehicle Routing Problems arising in city logistics. 39:3215–3228, 2012.

[17] S. K. Jacobsen and O. B. Madsen. A comparative study of heuristics for a two-level routing-location problem. *European Journal of Operational Research*, 5(6):378–387, 1980.

[18] B. Jarboui, H. Derbel, S. Hanafi, and N. Mladenović. Variable neighborhood search for location routing. *Computers and Operations Research*, 40(1):47–57, 2013.

[19] S. Karakatič and V. Podgorelec. A survey of genetic algorithms for solving multi depot vehicle routing problem. *Applied Soft Computing Journal*, 27:519–532, 2015.

[20] G. Li. Research on open vehicle routing problem with time windows based on improved genetic algorithm. *Proceedings - 2009 International Conference on Computational Intelligence and Software Engineering, CiSE 2009*, 2009.

[21] R. B. Lopes, C. Ferreira, and B. S. Santos. A simple and effective evolutionary algorithm for the capacitated location-routing problem. *Computers and Operations Research*, 70:155–162, 2016.

[22] O. B. Madsen. Methods for solving combined two level location-routing problems of realistic dimensions. *European Journal of Operational Research*, 12(3):295–301, 1983.

[23] S. M. Mirhedayatian, T. G. Crainic, M. Guajardo, and S. W. Wallace. A two-echelon location-routing problem with synchronisation. *Journal of the Operational Research Society*, 0(0):1–16, 2019.

[24] V. P. Nguyen, C. Prins, and C. Prodhon. A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem. *Engineering Applications of Artificial Intelligence*, 25(1):56–71, 2012.

[25] V. P. Nguyen, C. Prins, and C. Prodhon. Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking. *European Journal of Operational Research*, 216(1):113–126, 2012.

[26] B. Ombuki, B. J. Ross, and F. Hanshar. Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, 24(1):17–30, 2006.

[27] B. Ombuki-Berman and F. T. Hanshar. Using genetic algorithms for multi-depot vehicle routing. *Studies in Computational Intelligence*, 161:77–99, 2009.

[28] K. Pichka, A. H. Bajgiran, M. E. Petering, J. Jang, and X. Yue. The two echelon open location routing problem: Mathematical model and hybrid heuristic. *Computers and Industrial Engineering*, 121(May):97–112, 2018.

[29] M. Polacek, R. F. Hartl, K. Doerner, and M. Reimann. A Variable Neighborhood Search for the Multi Depot Vehicle Routing Problem with Time Windows. *Journal of Heuristics*, 10(6):613–627, 2004.

[30] J. Y. Potvin and J. M. Rousseau. An exchange heuristic for routeing problems with time windows. *Journal of the Operational Research Society*, 46(12):1433–1446, 1995.

[31] C. Prins. A simple and effective evolutionary algorithm for the vehicle routing problem. *Computers and Operations Research*, 31(12):1985–2002, 2004.

[32] C. Prodhon and C. Prins. A survey of recent research on location-routing problems. *European Journal of Operational Research*, 238(1):1–17, 2014.

[33] M. Savelsbergh and T. Van Woensel. 50th Anniversary Invited Article—City Logistics: Challenges and Opportunities. *Transportation Science*, 50(2):579–590, 2016.

[34] M. Schneider and M. Drexl. A survey of the standard location-routing problem. *Annals of Operations Research*, 259(1-2):389–414, 2017.

[35] Schwengerer Martin, P. Sandro, and G. R. Raidl. A Variable Neighborhood Search Approach for the Two-Echelon Location-Routing Problem. In *Evolutionary Computation in Combinatorial Optimization*, pages 13–24. Springer Heidelberg Dordrecht London NewYork, 2012.

[36] M. M. Solomon. Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints Author (s): Marius M . Solomon Published by : INFORMS Stable URL : https://www.jstor.org/stable/170697 REFERENCES Linked references are available on JSTOR for this a. 35(2):254–265, 1987.

[37] K. Wang, S. Lan, and Y. Zhao. A genetic-algorithm-based approach to the two-echelon capacitated vehicle routing problem with stochastic demands in logistics service. *Journal of the Operational Research Society*, 68(11):1409–1421, 2017.

[38] Y. Wang, K. Assogba, Y. Liu, X. Ma, M. Xu, and Y. Wang. Two-echelon location-routing optimization with time windows based on customer clustering. *Expert Systems with Applications*, 104:244–260, 2018.

[39] M. Winkenbach, P. R. Kleindorfer, and S. Spinler. Enabling urban logistics services at la poste through multi-echelon location-routing. *Transportation Science*, 50(2):520–540, 2016.