

FINDING OPTIMAL SHARING POINT IN SHARED DELIVERY SERVICES



PEYMAN HASHEMI BARAGOORI

ÖZYEGİN UNIVERSITY

MAY, 2025

FINDING OPTIMAL SHARING POINT IN SHARED DELIVERY SERVICES

by
Peyman Hashemi Baragoori

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Civil Engineering

Advisor: Assoc. Prof. Bekir O. Bartin

Graduate School of Science and Engineering
Özyeğin University
İstanbul

May, 2025

FINDING OPTIMAL SHARING POINT IN SHARED DELIVERY SERVICES

Approved by:

Assoc. Prof. Bekir O. Bartin, Advisor
Department of Civil Engineering
Özyeğin University

Asst. Prof. Yiğit Can Altan
Department of Civil Engineering
Özyeğin University

Asst. Prof. Göker Aksoy
Transportation Technology Institute
Gebze Technical University

Approval Date: March 5, 2025

To my family for their endless support and love



DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution.

Peyman Hashemi Baragoori

ABSTRACT

The recent surge in on-demand and quick delivery services has prompted the exploration of strategies to enhance operational efficiency and customer satisfaction. This study introduces the concept of Optimal Sharing Point (OSP) for exchanging parcels between two drivers in a delivery problem. By incorporating sharing points, couriers can optimize parcel exchanges, leading to reduced delivery completion time (DCT) and improved flexibility, resulting in faster deliveries and enhanced customer satisfaction. To address this, a mixed-integer linear programming (MILP) formulation is designed and implemented to find OSP in a hypothetical scenario on a small scale. Additionally, Genetic Algorithm (GA) and Google OR-Tools routing solver were employed to solve the problem on a larger scale. Finally, a hypothetical case study from Istanbul, Turkey, was used to validate the effectiveness of the proposed methodology. The proposed approach effectively minimized the DCT by considering real-world scenarios, network topology, and sharing points. The model eliminated the need for physical transfer nodes, allowing efficient parcel sharing without temporary storage. The approach addressed key objectives such as timely services, simultaneous sharing, actual network topology, and traffic considerations. The model's flexibility accommodated various objectives, emphasizing total delivery time or minimizing DCT.

Keywords: Q-commerce, Pickup and Delivery Problem with Transfers, Vehicle Routing Problem, Optimal Sharing Point, Delivery Completion Time, Google OR-Tools

ÖZET

Talep üzerine ve hızlı teslimat hizmetlerinde son dönemde yaşanan artış, operasyonel verimliliği ve müşteri memnuniyetini artırmaya yönelik stratejilerin araştırılmasına yol açtı. Bu makale, bir teslimat probleminde iki sürücü arasında paket alışverişi için OSP kavramını tanıtmaktadır. Kuryeler, paylaşım noktalarını birleştirerek paket alışverişlerini optimize edebilir, bu da DCT miktarının azalmasına ve esnekliğin artmasına, dolayısıyla daha hızlı teslimatlara ve müşteri memnuniyetinin artmasına olanak sağlar. Bu sorunu çözmek için, küçük ölçekte varsayımsal bir senaryoda OSP'yi bulmak üzere bir MILP formülasyonu tasarlandı ve uygulandı. Ek olarak, sorunu daha büyük ölçekte çözmek için Google OR-Tools yönlendirme çözücüsü kullanıldı. Son olarak, önerilen metodolojinin etkinliğini doğrulamak için İstanbul, Türkiye'den varsayımsal bir vaka çalışması kullanıldı. Önerilen yaklaşım, gerçek dünya senaryolarını, ağ topolojisini ve paylaşım noktalarını dikkate alarak DCT'yi etkili bir şekilde en aza indirdi. Model, fiziksel transfer düğümlerine olan ihtiyacı ortadan kaldırarak, geçici depolamaya gerek kalmadan verimli parsel paylaşımına olanak sağladı. Yaklaşım, zamanında hizmetler, eşzamanlı paylaşım, gerçek ağ topolojisi ve trafik hususları gibi temel hedefleri ele aldı. Modelin esnekliği, toplam teslimat süresini vurgulayan veya DCT'yi en aza indiren çeşitli hedefleri barındırıyordu.

Anahtar Kelimeler: Q-ticaret, Transferli Alma ve Teslimat Problemi, Araç Rotalama Problemi, Optimal Paylaşım Noktası, Teslimat Tamamlama Süresi, Google OR-Tools

ACKNOWLEDGEMENTS

I am deeply indebted to my advisor, Assoc. Prof. Bekir O. Bartın, for his unwavering guidance and support throughout my master's study. His mentorship was not only a privilege but a pivotal factor in my academic and personal development.

I extend my heartfelt thanks to the esteemed members of my dissertation committee, Asst. Prof. Yiğit Can Altan and Asst. Prof. Göker Aksoy for their invaluable insights and unwavering commitment in reviewing my thesis.

I extend my appreciation to the vibrant community at Ozyegin University, especially my friends who shared in both the joys and challenges of this journey. Their camaraderie and support were a source of strength and motivation.

I would also like to acknowledge the Scientific and Technological Research Council of Turkey (TUBITAK) for their financial support and guidance within ARDEB 1001 program under Grant 123N053, which were crucial in the successful completion of this research.

To my beloved parents, Parviz and Shahnaz, words cannot fully express my gratitude. Their unwavering support and unconditional love have been the bedrock of my happiness and success. They are not only my parents but my greatest advocates and sources of inspiration.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY	iv
ABSTRACT	v
ÖZET	vi
ACKNOWLEDGEMENTS	vii
LIST OF TABLES	x
LIST OF FIGURES	xi
LIST OF ACRONYMS AND ABBREVIATIONS	xii
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Thesis Outlines	2
1.3 Organization	3
2 PROBLEM DEFINITION & FORMULATION	5
2.1 Problem Definition	5
3 LITERATURE REVIEW	16
3.1 Collaboration in Transportation	16
3.2 The Role of Transshipment	17
4 METHODOLOGY	21
4.1 Modelling OSP Problem Using GA	21
4.1.1 Chromosome Representation	22
4.1.2 Fitness Function	24
4.1.3 Initial Population	24
4.1.4 REPRODUCTION: PARENT SELECTION, CROSSOVER, MU- TATION	30
4.1.5 PRELIMINARY RESULTS ON BENCHMARK INSTANCES	34
4.2 Modelling OSP problem using Google Routing Solver	35

5	CASE STUDY AND RESULTS	39
5.1	Case Study.....	39
5.1.1	Analysis Results	41
5.1.2	GA results.....	42
5.1.3	Google Solver Results.....	43
5.1.4	Google Solver vs. GA Results	46
6	CONCLUSIONS	48
	REFERENCES	51



LIST OF TABLES

Table 2.1: Notations	8
Table 4.1: ERX Cross Over Operator	31
Table 4.2: Swap Move Mutation Sample	33
Table 4.3: Remove-Insert Move Mutation Sample	34
Table 5.1: Costs of Sharing Plan Found by GA (All Units are in Seconds)	42
Table 5.2: Details of Costs for Each Considered Plan (All Units Are in Seconds) ...	45

LIST OF FIGURES

Figure 2.1: A hypothetical shared delivery example: (a) without considering sharing points. (b) considering transshipment at sharing points.	6
Figure 2.2: Stages of a hypothetical shared delivery process.	7
Figure 2.3: Scenario 1: optimal OSP and routes for each driver ($DCT = 121$ time units).....	11
Figure 2.4: Scenario 2: (a) OSP and routes for each driver ($DCT=177$ time units). (b) Second best OSP and routes for each vehicle ($DCT=178$ time units).	12
Figure 2.5: Scenario 3: (a) Optimal OSP and routes for each driver ($DCT=158$ time units). (b) Second best OSP and routes for each vehicle ($DCT=159$ time units).	14
Figure 4.1: Chromosome representation of sample solution	23
Figure 4.2: Initialization of Feasible Solution of GA, phase 1	27
Figure 4.3: Initialization Of Feasible Solution of GA, phase 2	28
Figure 4.4: Initialization of feasible solution for GA	29
Figure 4.5: Preliminary analysis results of GA solution.....	35
Figure 4.6: Modelling OSP for a single sharing point	37
Figure 5.1: Map of the case study area	40
Figure 5.2: Two-minute isochrone maps	41
Figure 5.3: Average GA result	43
Figure 5.4: Optimal sharing plan, Senario I.....	45
Figure 5.5: Optimal sharing plan, Senario 2	46

LIST OF ACRONYMS AND ABBREVIATIONS

2PX	2-point crossover
CP	Constraint Programming
DCT	delivery completion time
DT	Driving Time
ERX	edge recombination crossover
GA	Genetic Algorithm
GPDP	General Pickup and Delivery Problem
LOX	linear ordered crossover
MIP	Mixed-Integer Programming
MILP	mixed-integer linear programming
NN	Nearest Neighbor
OSP	Optimal Sharing Point
OX	Ordered crossover
PDPT	Pick-up and Delivery Problems with Transshipment
PDP	Pickup and Delivery Problem
PMX	Partially matched crossover
RBX	route-based crossover
REX	route exchange crossover
TSP	Travelling Salesman Problem
VRP	Vehicle Routing Problem

WT

Wait Time



1. INTRODUCTION

1.1 Motivation

The rise of q-commerce, also known as quick commerce or on-demand delivery, has revolutionized the online shopping landscape, emphasizing rapid deliveries within a short period. Initially focused on food delivery, q-commerce has rapidly expanded to encompass various categories such as grocery and medicine. The COVID-19 pandemic and the subsequent restrictions played a significant role in boosting q-commerce as it allowed retailers to continue operating through quick home deliveries [1]. This sustained period of limitations resulted in a long-term shift in consumer behavior towards faster deliveries. The growth of companies like Uber Eats, DoorDash, Gorillas, Dija, and Getir worldwide is a testament to the increasing popularity of q-commerce [1]. Due to these changes in consumer purchasing patterns, it is necessary for the transportation and logistics sector to adapt itself to these changes and address the associated efficiency issues accordingly.

While smaller companies may rely on their own fleet to manage deliveries, larger companies often outsource these logistics to courier services [2]. In both cases, the delivery cost (i.e., time and distance) is a critical factor that significantly impacts customer satisfaction and the reorder numbers. In many cases, customers are willing to pay a higher price for faster delivery, making this factor even more crucial for companies that specialize in quick delivery services. Due to financial constraints, small companies typically have a small group of drivers or couriers. External courier companies may be needed during peak hours or when orders exceed capacity.

Evidently, the need to optimize delivery routes is not limited to small companies but also applies to larger companies in terms of quick delivery services. These companies prioritize time as the most valuable resource; for example, apps such as Gorillas in

Germany, Dija in Britain, and Getir in Turkey are among the many platforms that assure users swift grocery deliveries within ten minutes. As more parcels are assigned to a driver, meeting a ten-minute delivery time becomes increasingly challenging. The distance between the location of customers and the wait time at each destination would result in delayed delivery, making it an unfeasible task.

In certain cases, there are opportunities to seek assistance from other drivers to ensure timely deliveries. Drivers from the same company who have completed deliveries and are returning to the depot can provide support. Additionally, collaborative transportation initiatives that foster cooperation among drivers from different companies offer another avenue for help. Moreover, in “crowdshipping” scenarios, independent drivers or individuals in close proximity can also step in to offer their assistance, contributing to the overall efficiency of the delivery process. The challenge then becomes identifying where these drivers should meet to exchange packages to optimize delivery routes.

1.2 Thesis Outlines

In this study, we specifically focused on a scenario involving two homogeneous drivers, with only one of them carrying parcels at the onset. As a result, the other driver is not carrying any parcel and has full capacity available at the start. The primary objective was for these drivers to meet at designated transfer or customer nodes, where they would wait and share the parcels. In our problem, the transfer points were solely intended for the exchange between drivers, without any provision for parcel storage, such as lockers. To underscore the notion of sharing and distinguish it from the broader concept of transfer nodes, we renamed the transfer nodes in our model as “sharing points.” It is crucial that sharing takes place simultaneously, without any parcels being left for later pick-up. Moreover, there were no additional pickup tasks for the drivers; their sole responsibility was the delivery of parcels. Hence, it can be more accurately described as a Vehicle

Routing Problem (VRP) with transfer nodes than a Pick-up and Delivery Problems with Transshipment (PDPT). The main contributions of this study are as follows:

- Developing an exact model and a GA as a benchmark to determine the OSP within a VRP scenario by incorporating the concept of transfer nodes in the PDPT.
- Employing Google OR-Tools VRP solver with a Constraint Programming (CP) approach to find the OSP.
- Considering customer nodes as additional sharing points to enhance the efficiency of the delivery system.
- Incorporating real-world scenarios by utilizing actual network topology and traffic information.

1.3 Organization

The organization of the rest of the thesis is outlined as follows. Chapter 2 addressed the problem definition and formulation. It outlined the specific objectives and constraints of the study and presented the developed exact model applied to hypothetical delivery scenarios for illustrative purposes. Chapter 3 then shifted focus to the existing body of knowledge. A literature review was conducted on collaboration in transportation and joint route planning, with a particular emphasis on PDPT. Recognizing the limitations of exact models for large-scale problems, Chapter 4 proposed alternative approaches for solving the OSP problem. This chapter introduced two distinct methods: the GA as a metaheuristic method and Google OR-Tools as a CP solver. Chapter 5 applied these approaches to a case study. It presented a case study that utilized both the GA and Google solver on the previously defined hypothetical delivery scenarios. This chapter compared and analyzed the results obtained from each approach, allowing for an assessment of their effectiveness in the context of the OSP problem. Finally, Chapter 6 concludes the thesis

by summarizing the key findings of the research. It discussed any limitations encountered during the study and proposed potential avenues for future work that could build upon this research.



2. PROBLEM DEFINITION & FORMULATION

2.1 Problem Definition

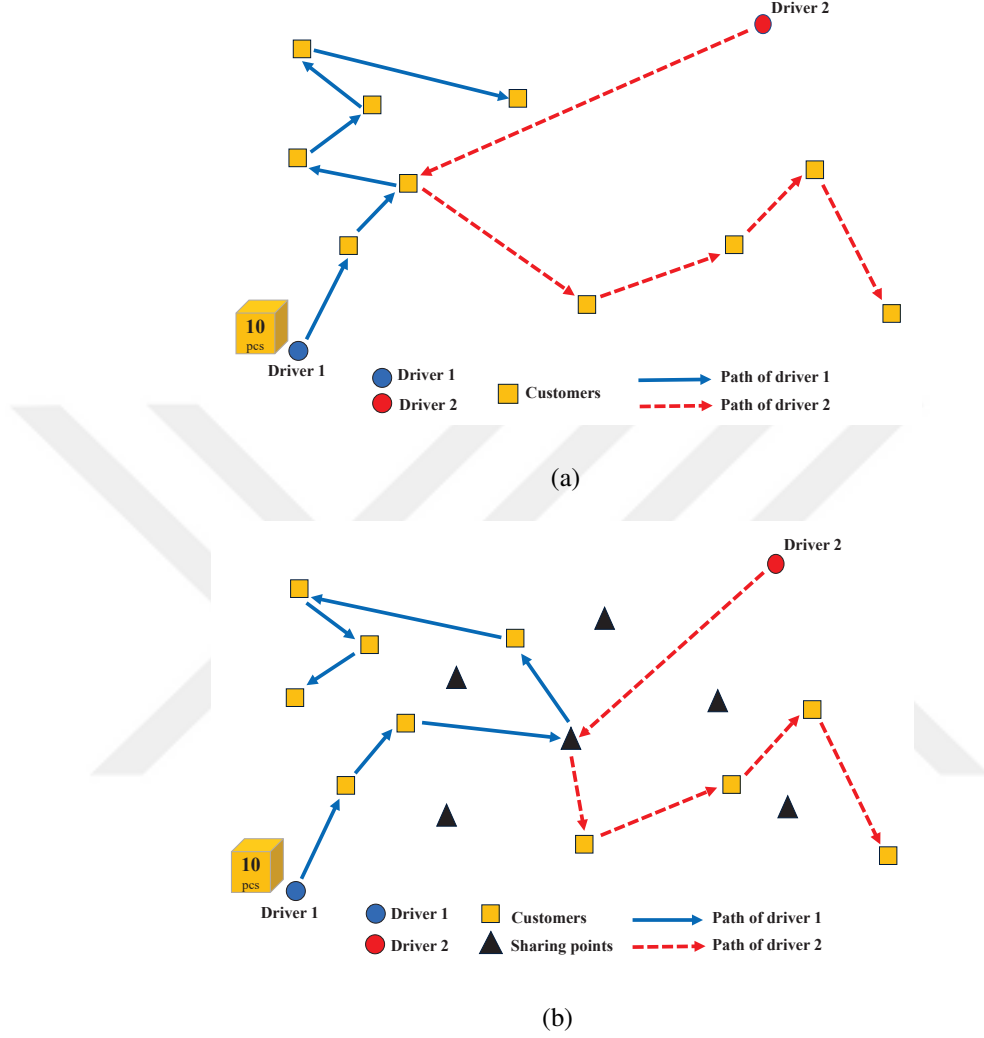
When identifying the OSP for two drivers to divide their parcels, the delivery cost can be measured in various ways, such as distance, time, or other relevant parameters, depending on the intended application [3]. In practice, minimizing the time of delivery is often the most critical cost factor in the case of quick delivery businesses. Thus, minimizing the delivery time for the last parcel, i.e., DCT of the last customer, is selected here as the main goal when identifying the OSP.

Fig. 2.1 depicts an example of the problem at hand. Driver 1 is responsible for delivering 10 parcels to customers as quickly as possible, while driver 2 is available to assist. Consider that driver 1 has already left the depot. Without sharing points, drivers would likely have to meet at customers' locations (Fig. 2.1 (a)), which might not result in the most efficient routing due to the limited number of these locations. To enhance routing flexibility and achieve better coverage and accessibility in the service area, we use not only customer locations but also other sharing points (Fig. 2.1 (b)). The objective is then to minimize the duration of the longest single route between two drivers, as the selected goal is to complete all deliveries as soon as possible. As demonstrated in the literature review, the incorporation of sharing points can lead to more efficient routes and improved overall performance.

In this scenario (Fig. 2.1 (b)), driver 2 is expected to move directly to the selected sharing point. On the other hand, driver 1 either goes to the sharing point directly or after delivering some of the assigned parcels.

As shown in Fig. 2.2, the entire process is composed of three stages. Part 1: Driver 1 traveling from its initial location (O_1) to the sharing point. Part 2: Driver 2 traveling from its initial location (O_2) to the sharing point. Part 3: Drivers deliver parcels after

Figure 2.1: A hypothetical shared delivery example: (a) without considering sharing points. (b) considering transshipment at sharing points.



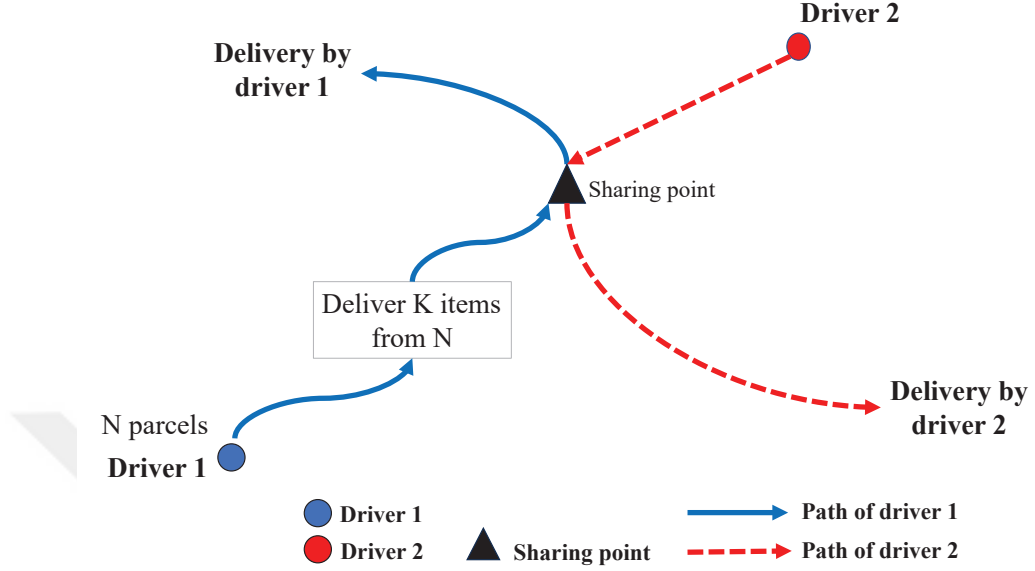
sharing.

In Part 1, driver 1 has the option to deliver some of the parcels before meeting with driver 2, depending on the selected sharing point and the locations of the customers. The path of driver 1 can be expressed as:

$$\text{Driver}_1 (O_1 \rightarrow \text{item}_1 \rightarrow \dots \rightarrow \text{item}_K \rightarrow \text{sharing point})$$

In Part 2, driver 2 should directly move to the sharing point without any delivery.

Figure 2.2: *Stages of a hypothetical shared delivery process.*



Therefore, the path of this part is solely dependent on the location of the sharing point and can be expressed as:

$$\text{Driver}_2 (O_2 \rightarrow \text{sharing point})$$

In Part 3, the parcels will be optimally divided between the two drivers. It is assumed that driver 1 is carrying N parcels, and K represents the number of parcels to be delivered before reaching the sharing point. In Part 3, there will be a total of $N - K$ items that need to be delivered. Then the remaining paths of drivers would be:

$$\text{Driver}_1 \& 2 (\text{sharing point} \rightarrow [\text{item}_1 \rightarrow \dots \rightarrow \text{item}_{N-K}])$$

The location of the OSP is directly impacted by the distribution of customer locations and the relative positions of driver 1 and 2.

Table 2.1 lists the notations and their descriptions used in the formulation. An

undirected graph $G(V, E)$ is used to model a road network, where V represents the set of all nodes. The V set includes nodes for customers (CU), sharing points (SH), the origin nodes of two drivers (O), and a virtual depot location as the returning points of drivers (the d point).

Table 2.1: *Notations*

Notation	Description
V	The set of all nodes
CU	The set of customers' nodes (i.e., delivery points)
SH	The set of sharing nodes
O_k	The set of origin of two drivers
i, j	The indices of all nodes
k	The indices of drivers
d	The index of the virtual depot
y_{ijk}	Binary variable if driver k goes from point i to j
u_j	Number of nodes visited at node j
z_j	Binary variable if node j selected as a sharing point
c_{ij}	Considered traveling cost from node i to j
C_j	Considered cumulative cost for node j
M	A big integer number
DT_k	Driving Time of driver k
WT_k	Wait Time of driver k at OSP
DCT	Delivery Completion Time

A min-max model has been used to minimize the delivery completion time. In this model, the objective function is defined as the minimization of the delivery time of the last parcel to the last customer. The DCT is calculated as the maximum of delivery times of driver 1 and 2. In other words, if driver 1 finishes delivering parcels at t_1 , and driver 2 finishes at t_2 , DCT is determined as the $\max(t_1, t_2)$. It should be noted that the speed of the vehicles is considered constant and equal, and accordingly, the Euclidean distance is used as the basis for travel cost measurement. Now the model is:

$$\text{minimize } DCT$$

$$s.t.$$

$$\sum_j z_j = 1 \quad \forall j \in (CU \cup SH) \quad (2.1)$$

$$\sum_i \sum_k y_{ijk} = 1 + z_j \quad \forall j \in (CU) \quad (2.2)$$

$$\sum_j \sum_k y_{ijk} = 1 + z_i \quad \forall i \in (CU) \quad (2.3)$$

$$\sum_i \sum_k y_{ijk} = 2z_j \quad \forall j \in (SH) \quad (2.4)$$

$$\sum_j \sum_k y_{ijk} = 2z_i \quad \forall i \in (SH) \quad (2.5)$$

$$y_{O_2j2} = z_j \quad \forall j \in (CU \cup SH) \quad (2.6)$$

$$\sum_i y_{iO_kk} = 0 \quad \forall i \in (CU \cup SH) \quad (2.7)$$

$$\sum_i y_{idk} = 1 \quad \forall i \in V \quad (2.8)$$

$$\sum_j y_{dj k} = 0 \quad \forall j \in V \quad (2.9)$$

$$u_i - u_j + 1 \leq (|V| - 1)(1 - y_{ijk}) \quad \forall i, j, k \quad (2.10)$$

$$C_j \geq C_i + c_{ij} - M(1 - y_{ijk}) \quad \forall i, j, k \quad (2.11)$$

$$DCT \geq C_j \quad \forall j \in V \quad (2.12)$$

The constraint set (2.1) states that only one sharing point should be selected. The constraint set (2.2) states that each delivery node can have one or two entry arcs. If a delivery node is not selected as a sharing point, it will have one entry arc and two entry arcs otherwise. The constraint set (2.3) states that each delivery node can have one or two departure arcs: two departure arcs if selected as a sharing point and one otherwise. The constraint set (2.4) states that each sharing node can have zero or two entry arcs: two entry arcs if selected as a sharing point and zero otherwise. The constraint set (2.5) states that each sharing node can have zero or two departure arcs, and if not selected as a sharing point, it will have zero departure arcs and two departure arcs otherwise. The constraint set (2.6) states that driver 2 must go to the selected sharing point as its first destination. The constraint set (2.7) states that there should be no entering arcs to the drivers' origin nodes. The constraint set (2.8) states that both drivers should go to the virtual depot point. The constraint set (2.9) states that no driver should leave the virtual depot point. The constraint set (2.10) guarantees that there will be no subtours in the final solution. The constraint set (2.11) states that the cumulative delivery time at point j is greater than or equal to the cumulative delivery time of the previous node i plus the cost of travel from i to j . The constraint set (2.12) ensures that the model considers the maximum delivery time to node j as the final cost. In other words, the final cost must be equal to the maximum of the arrival time of two drivers to their final delivery point.

The presented optimization problem is solved using the Pyomo package and CPLEX solver in Python [4]. The results are demonstrated by considering three hypothetical delivery scenarios with different locations for drivers and delivery points. Each scenario involves two drivers (indexed as 1 and 2), eight delivery points (indexed from 3 to 10), and a total of 16 potential sharing points (indexed from 11 to 26). In this section, we introduce Driving Time (DT) and Wait Time (WT) to be used in the analyses of the results of each scenario. DT_k represents the cumulative driving time of each driver, and WT_k

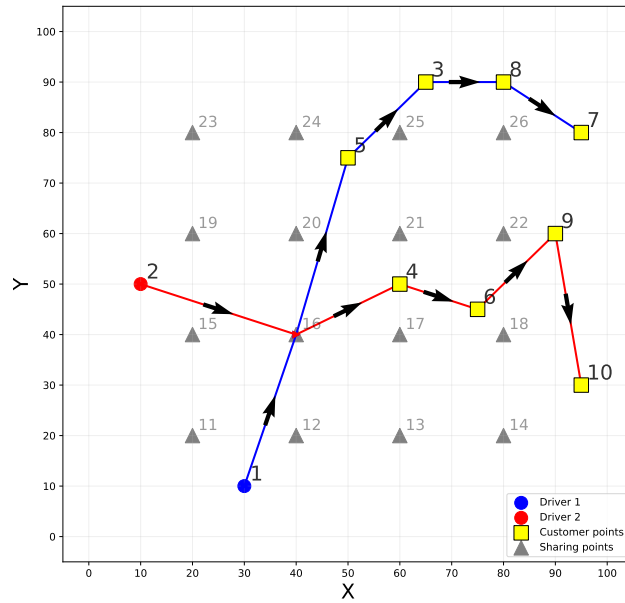
represents the wait time of each driver at the OSP.

$$DCT = \max(DT_1 + WT_1, DT_2 + WT_2) \quad (2.13)$$

DCT equals the greater delivery makespan of both drivers. In (2.13), we partition service time into driving time and wait time for each driver. When two drivers arrive at the OSP simultaneously, their wait times are both set to zero; otherwise, only one driver's wait time is equal to zero. In Scenario 1, depicted in Fig. 2.3, drivers are positioned in the lower-left region in close proximity, while most delivery points are located in the upper-right region, far from the drivers. The optimization result shows that the OSP for this scenario is point number 16, as shown in Fig. 2.3.

The optimal routes for drivers 1 and 2, and the respective costs are as follows.

Figure 2.3: Scenario 1: optimal OSP and routes for each driver ($DCT = 121$ time units)



$$\text{Driver 1's route: } 1 \xrightarrow{31} 16_{OSP} \xrightarrow{36} 5 \xrightarrow{21} 3 \xrightarrow{15} 8 \xrightarrow{18} 7 \quad (DT_1 = 121) \quad (2.14)$$

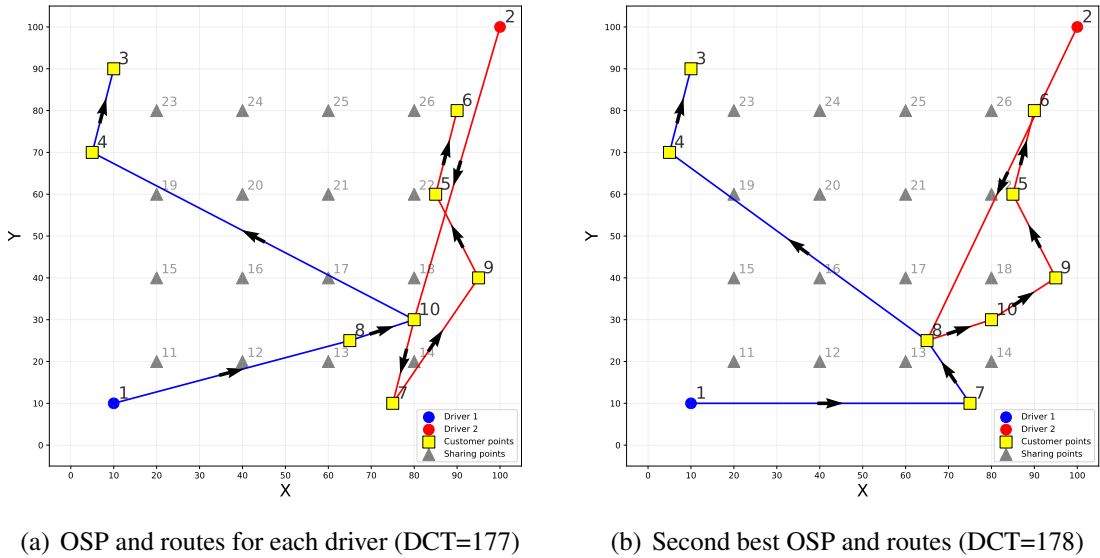
$$\text{Driver 2's route: } 2 \xrightarrow{31} 16_{OSP} \xrightarrow{22} 4 \xrightarrow{15} 6 \xrightarrow{21} 9 \xrightarrow{30} 10 \quad (DT_2 = 119) \quad (2.15)$$

$$WT_1 = 0, WT_2 = 0 \rightarrow DCT = \max(121, 119) = 121 \quad (2.16)$$

This choice of OSP is not surprising, as the drivers' proximity to one another suggests that the optimal strategy should be promptly sharing parcels, taking into account the customers' locations. In this scenario, the WT for both drivers are equal to zero because they are at an equal distance from the OSP and reach there simultaneously. Therefore, DCT is equal to 121, the maximum of vehicles' driving times. The second optimal option for OSP would be the delivery point number 4 with $DCT = 130$.

In Scenario 2, depicted in Fig. 2.4, the drivers are positioned farther away from each other. Driver 1 is situated in the lower-left area, while driver 2 is in the upper-right region. The delivery points, on the other hand, are mostly concentrated in the lower-right region. The OSP for parcel distribution, as revealed by the optimization solution, is point number 10, which also happens to be a delivery point (Fig. 2.4(a)).

Figure 2.4: Scenario 2: (a) OSP and routes for each driver ($DCT=177$ time units). (b) Second best OSP and routes for each vehicle ($DCT=178$ time units).



$$\text{Driver 1's route: } 1 \xrightarrow{57} 8 \xrightarrow{15} 10_{OSP} \xrightarrow{85} 4 \xrightarrow{20} 3 \quad (DT_1 = 177) \quad (2.17)$$

$$\text{Driver 2's route: } 2 \xrightarrow{72} 10_{OSP} \xrightarrow{20} 7 \xrightarrow{36} 9 \xrightarrow{22} 5 \xrightarrow{20} 6 \quad (DT_2 = 170) \quad (2.18)$$

$$WT_1 = 0, WT_2 = 0 \rightarrow DCT = \max(177, 170) = 177 \quad (2.19)$$

In contrast to the results of Scenario 1, here, driver 1 is tasked with delivering a parcel at delivery point number 8 before proceeding to the OSP. This result is anticipated due to the higher density of delivery points in the lower-right area, thereby substantiating the rationale for the OSP to be situated near that region. This allows the drivers to allocate more time in the densely populated area to efficiently deliver the parcels. On the other hand, similar to Scenario 1, the WT for both drivers are zero because they both reach the OSP at the same time. DCT, in the optimal solution, is equal to 177 time units.

Additionally, the optimization results indicate that the second best option for OSP is point number 8, as shown in Fig. 2.4(b). The routes for two drivers and their respective costs at the second optimal solution are:

$$\text{Driver 1's route: } 1 \xrightarrow{65} 7 \xrightarrow{18} 8_{OSP} \xrightarrow{75} 4 \xrightarrow{20} 3 \quad (DT_1 = 178) \quad (2.20)$$

$$\text{Driver 2's route: } 2 \xrightarrow{82} 8_{OSP} \xrightarrow{15} 10 \xrightarrow{18} 9 \xrightarrow{22} 5 \xrightarrow{20} 6 \quad (DT_2 = 157) \quad (2.21)$$

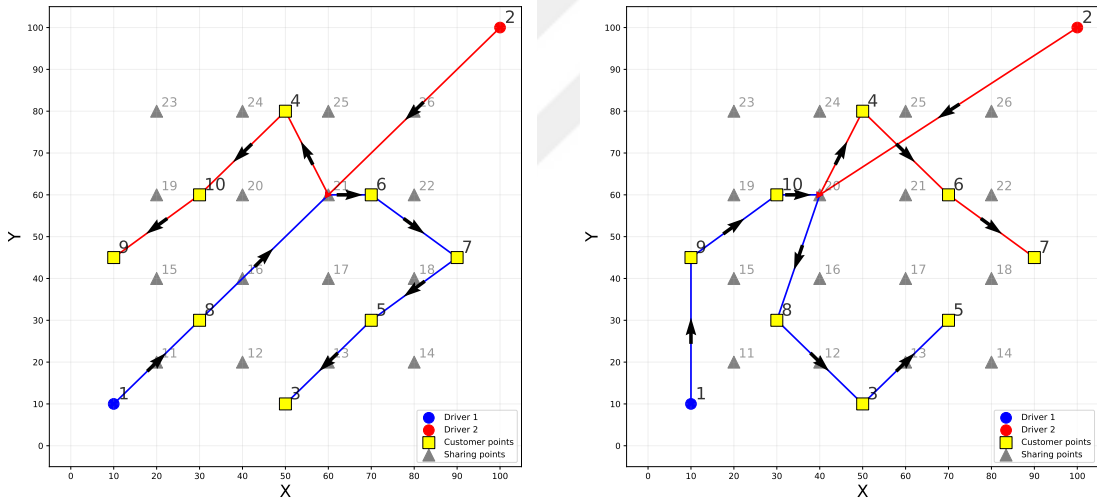
$$WT_1 = 0, WT_2 = 1 \rightarrow DCT = \max(178 + 0, 157 + 1) = 177 \quad (2.22)$$

There is a slight difference of only one time unit in DCT, compared to that of the optimal solution (177 to 178 time units, respectively). Another important aspect to consider is that if delivery point number 8 was chosen as the OSP, the driving cost for the second vehicle would be 157 time units ($DT_2 = 157$), 13 time units less compared to that of selecting point number 10 as the OSP ($DT_2 = 170$). While opting for faster parcel delivery by drivers, it is important to consider that it will only reduce delivery time by 1 time unit, while requiring an additional 12 time units of driving time. This case

illustrates the significance of choosing an appropriate criterion in real-life scenarios and emphasizes the need for careful consideration between the most precise solution and the most practical solution.

In Scenario 3, as depicted in Fig. 2.5, the distribution of delivery points exhibits a nearly symmetrical pattern in relation to the drivers' locations. In contrast to the previous two scenarios, where the OSP could be estimated by visual inspection, here determining the OSP becomes slightly more challenging. The optimization solution shows that point number 21 (Fig. 2.5(a)) is identified as the OSP. The optimal routes for two drivers and their respective costs are:

Figure 2.5: Scenario 3: (a) Optimal OSP and routes for each driver ($DCT=158$ time units). (b) Second best OSP and routes for each vehicle ($DCT=159$ time units).



(a) Optimal OSP and routes ($DCT=158$)

(b) Second best OSP and routes ($DCT=159$)

$$\text{Driver 1's route: } 1 \xrightarrow{28} 8 \xrightarrow{42} 21_{OSP} \xrightarrow{10} 6 \xrightarrow{25} 7 \xrightarrow{25} 5 \xrightarrow{28} 3 \quad (DT_1 = 158) \quad (2.23)$$

$$\text{Driver 2's route: } 2 \xrightarrow{56} 21_{OSP} \xrightarrow{22} 4 \xrightarrow{28} 10 \xrightarrow{25} 9 \quad (DT_2 = 157) \quad (2.24)$$

$$WT_1 = 0, WT_2 = 14 \rightarrow DCT = \max(158, 131 + 14) = 158 \quad (2.25)$$

The DCT for the optimal OSP is only one time unit lower than that for selecting point 20 as the OSP, the second optimal solution (158 to 159 time units, respectively), as shown in Fig. 2.5(b). The second optimal routes for the two drivers and their respective costs are as follows:

$$\text{Driver 1's route: } 1 \xrightarrow{35} 9 \xrightarrow{25} 10 \xrightarrow{10} 20_{OSP} \xrightarrow{31} 8 \xrightarrow{28} 3 \xrightarrow{28} 5 \quad (DT_1 = 158) \quad (2.26)$$

$$\text{Driver 2's route: } 2 \xrightarrow{72} 20_{OSP} \xrightarrow{22} 4 \xrightarrow{28} 6 \xrightarrow{25} 7 \quad (DT_2 = 157) \quad (2.27)$$

$$WT_1 = 2, WT_2 = 0 \rightarrow \text{DCT} = \max(157 + 2, 147) = 159 \quad (2.28)$$

When point number 20 is selected as the OSP, the driver number 1 reaches the OSP 2 time units earlier, resulting in an increase in the DCT.

It should be noted that the formulated OSP problem is solved using linear models and exact methods for the hypothetical delivery scenarios. However, in problems such as VRP, exact methods are not computationally feasible when applied to solving large-scale problems. For example, in the hypothetical problem presented above with eight delivery points, the runtime was approximately four minutes. If one more delivery point was added, the total runtime increased to 19 minutes. The following section presents the use of a metaheuristic method for solving this problem for larger-scale operations and real-world scenarios.

3. LITERATURE REVIEW

This chapter delves into the existing body of knowledge by reviewing research on collaboration in transportation and joint route planning. The focus then shifts to PDPT, exploring how this concept can be utilized for optimizing delivery operations.

3.1 Collaboration in Transportation

Collaboration in transportation and joint route planning between companies has gained significant attention in recent years as an effective approach to finding efficient solutions and improving operational performance [5, 6]. Efficiency is a critical factor in collaborative route planning, with lower costs and increased flexibility being primary objectives. Research has shown that carriers can achieve significant cost reductions and environmental benefits by collaborating through platforms [7]. By pooling resources and sharing routes, logistics providers can enhance operational efficiency and reduce transportation costs, particularly in last-mile delivery operations [8]. Two main types of collaboration are identified in the literature: horizontal and vertical collaboration [9]. Horizontal collaboration refers to resource sharing among companies operating at the same level of the supply chain, whereas vertical collaboration involves cooperation between companies at different levels of the chain, such as manufacturers and carriers [9].

While collaboration offers opportunities to improve customer service levels, sharing customer data and delivery orders with competitors remains a concern for logistics service providers [10]. This hesitancy often limits the practical implementation of collaboration strategies proposed in the literature [10]. Addressing the issue of data sharing and establishing trust among stakeholders is crucial for successful collaboration. Another important aspect of collaboration is determining how to share the benefits, whether in terms of cost savings, profit distribution, or resource allocation [5]. Finding equitable and mutually beneficial models for sharing benefits is crucial for fostering successful collaborative

partnerships.

Although collaboration possesses benefits such as cost reductions, heightened flexibility, and enhanced operational efficiency, it is necessary to address significant obstacles in the realm of data sharing and benefit distribution. Instead of utilizing collaborative route planning due to its associated challenges, we have opted to focus on drivers within the same company. However, inspired by the concept of transshipment and transfer nodes from pick-up and delivery problems, as described in [8], we have incorporated these elements into our approach.

In the face of rising delivery cost pressures driven by the need for speed, some companies are exploring alternative models like crowdshipping, a subset of crowd logistics. Crowdshipping, as described in [11], leverages a network of individuals to deliver packages. These individuals utilize their own vehicles and typically have some spare capacity, such as empty space in their car trunk or luggage, aligning with the concept of "underutilized assets" [11]. This approach parallels the "occasional drivers" concept explored by [12], where companies supplement their own delivery fleet with individuals willing to make a single delivery for a fee. By integrating with crowdshipping platforms and mobile apps, companies can connect with this network of individuals and potentially fulfill deliveries more efficiently, especially during peak times or for smaller deliveries, while potentially reducing overall delivery costs. This approach aligns with the broader definition of crowd logistics provided by [13], which emphasizes the use of collaborative platforms to utilize distributed and idle resources for logistics needs.

3.2 The Role of Transshipment

Transshipment plays a pivotal role in facilitating the seamless transfer of goods between different stakeholders within collaborative and crowdsourced delivery services. In these models, transshipment occurs either at depots or through specialized transfer

points strategically positioned along delivery routes. The concept of transshipment is integral to the formulation of the PDPT, where requests or parcels have the flexibility to be transferred from one vehicle to another at designated transfer stations. This practice enables the efficient drop-off and temporary storage of parcels at transfer points, allowing subsequent vehicles to pick them up and continue the delivery process. By leveraging transshipment nodes, vehicles can adjust their loads to optimize route efficiency and minimize overall transportation costs, enhancing the effectiveness of collaborative delivery operations.

In pickup and delivery scenarios, vehicles are responsible for transporting loads directly from their origins to destinations without transshipment at intermediate points. The General Pickup and Delivery Problem (GPDP) aims to create optimized routes using available vehicles. GPDP encompasses three well-known routing problems: The Pickup and Delivery Problem (PDP) handles single-origin and single-destination requests. The Dial-a-Ride Problem involves a PDP where the items to be transported are individuals or people, and the VRP considers all origins or destinations at the depot [3].

With the advancement of e-commerce and information technology, there is a growing interest in exploring innovative approaches to enhance transportation systems [14]. Urban areas experiencing rapid growth, coupled with the rising significance of e-commerce and heightened consumer service expectations, have led to the emergence of novel models for urban freight delivery [15]. Enabling transshipment is regarded as a highly promising endeavor, leading to the formulation of the PDPT. In the context of PDPT, requests have the flexibility to be transferred from one vehicle to another. This transshipment allows for the drop-off of requests at designated transfer stations, where they can be temporarily stored. Subsequently, other vehicles can visit these transfer stations to pick up the requests and finalize the delivery process [14]. To minimize the overall cost of the vehicle routes, the two vehicles responsible for carrying the requests or transport loads have the

flexibility to transfer, adjust, or swap their loads at designated transshipment nodes. This flexibility allows them to accommodate different load-carrying capacities and consider varying transportation costs along the routes [16].

The concept of transshipment is well-established, and its optimization impact on transportation networks is evident [16, 17]. In the context of the PDP, its utility has been confirmed through empirical analysis on data from a San Francisco-based courier service company, as studied by Mitrovic ´-Minic ´ and Laporte in 2006 [18]. The company effectively utilized transshipment to serve multiple cities using homogeneous, uncapacitated vehicles [16]. In practical terms, transshipment involves one vehicle collecting the load from the pickup location and dropping it at a designated transshipment point. Subsequently, another vehicle carries the load to its intended delivery location [18].

Transfer nodes have been proven to be highly beneficial in various transportation scenarios, particularly in real-life situations. These scenarios include demand-responsive transport for people with mental disabilities [19], microtransit with synchronized en-route transfers [20], system-wide travel distance reduction [15], improved taxi service during rush hours [21], optimized ridesharing systems [22], and enhanced passenger transfer over traditional ridesharing methods [23]. Overall, these examples highlight the significant advantages of introducing transfer nodes, showcasing their potential to optimize transportation systems.

Cortés et al. [9] presented a comprehensive Mixed-Integer Programming (MIP) model for the PDPT. Their model includes provisions for passengers to transfer from one vehicle to another at designated transfer nodes. In this context, vehicles have the option to either load or unload passengers at these transfer nodes [9]. In another study, Rais et al. [16] focused on addressing the PDPT with and without time windows for services. They proposed a novel MILP model to solve the problem, employing a directed graph approach. Their MILP model for the PDPT was highly versatile, accommodating multiple features,

such as multiple depots, different vehicle types, unlimited transshipment at nodes, and flexible vehicle fleet sizing to fulfill all customer requests [16].

Feasible solutions in PDPT problems require satisfying various typical constraints, such as precedence constraints, time window constraints, and capacity constraints [9, 24]. These constraints are crucial in formulating effective solutions for PDPT scenarios to optimize various aspects of the problem. Each of these constraints is defined based on specific problem conditions, available resources, and the objectives set for the problem. For instance, some problems may not consider time windows, while others may not require capacity considerations due to parcel size. However, since our problem can be considered a special case within the broader concept of PDPT, with some distinctive constraints, we opted to formulate an exact model tailored to our specific assumptions and objectives. Consequently, we chose to refrain from introducing a new problem formulation for PDPT, as more general formulations already exist in the literature [15, 16, 9]. By presenting our problem with well-defined assumptions and objectives, we aim to clearly and concisely depict its unique characteristics.

In most PDPT problems, transfer nodes need to meet specific criteria, limiting the number of available transfer points. Typically, parcels are placed by the first driver at the transfer points for subsequent drivers to pick up later. Examples of suitable transfer point locations include shops, gas stations with a backroom, automated locker stations, or intermediate depots [14, 15, 16, 10, 25]. It is assumed that these facilities have the capacity to handle the volume of parcels and provide storage for a limited time, eliminating the need for drivers to wait when handing over parcels to the next driver [15, 25]. The specific conditions for transfer points can vary depending on the application, with considerations for safety, convenience, and operational requirements [21].

4. METHODOLOGY

In this chapter, we address the problem of finding OSP by leveraging both a GA and the Google OR-Tools VRP solver model. The following sections will detail the modeling approach for the problem and present the results for hypothetical delivery scenarios that were previously solved using exact methods in earlier chapters.

4.1 Modelling OSP Problem Using GA

GA is a type of metaheuristic optimization technique inspired by the principles of natural selection and biological evolution [26]. It operates on a population of individuals, each representing a potential solution to the problem. Through processes like selection, crossover, and mutation, the GA iteratively evolves the population towards better solutions. GA is particularly well-suited for problems where finding an optimal solution is difficult due to a complex search space or the presence of multiple local optima.

Designing a GA involves several general steps that we will describe and tailor to solve our specific problem. These steps include:

1. Solution (chromosome) representation
2. Fitness function
3. Initial population: Size and generation method
4. Reproduction: Parent selection, crossover, mutation
5. Stopping (termination) condition

In the context of our vehicle routing problem, consider a scenario with two drivers indexed as 1 and 2. The delivery points consist of eight locations, indexed from 3 to 10. Additionally, there are 16 potential sharing points, indexed from 11 to 26. Customers are

denoted as black nodes, and for representation purposes, these customers are duplicated as sharing points with indexes 27 to 34.

Our primary objective is to determine the OSP, which serves as a transfer node for the parcels. The OSP must be strategically chosen to minimize the overall delivery completion time for all parcels. Notably, the cost function focuses on minimizing service time, resembling an open VRP. Now, let's delve into the design of our solution representation using a genetic algorithm.

4.1.1 Chromosome Representation

To mimic the solution representation in a GA, we use a 1D array that signifies the solution with specific specifications for its structure.

- **Array length:** The chromosome is essentially a linear array containing 12 elements. Each element is a placeholder for a specific action or location in the delivery process.
- **Driver identification:** The first element of the array is designated to identify the driver. It is consistently set to 1, indicating that the subsequent actions pertain to the path of driver 1.
- **Driver 1's path:** The subsequent elements in the array represent the path of driver 1. This path includes the delivery of parcels to customers. Notably, it must incorporate at least the traversal to the sharing point and subsequent deliveries to customers.
- **Driver 2's position:** The second element of the array denotes the presence of driver 2. Its specific position within the array is variable, allowing for flexibility in the exploration of different solutions. However, it is crucial to note that the next cell after the identification of driver 2 is reserved for the sharing point ID.

- **Sharing points:** Following the identification of driver 2, the next element in the array is reserved for the sharing point ID. This strategic placement ensures that the sharing point is the first location visited by the second driver. Sharing points play a pivotal role as transfer nodes for parcel distribution.
- **Driver 2's path:** The subsequent elements in the array represent the path of driver 2. This path includes the delivery of parcels to customers after leaving the sharing point.

By structuring the chromosome in this manner, we address the intricacies of the vehicle routing problem and the specific constraints outlined for this instance. The design ensures that each customer is visited only once by a single driver, the sharing point is visited only once by both drivers, and the order of visits adheres to the prescribed sequence.

This representation provides a clear and organized way to encode a solution, making it amenable to the application of genetic operators such as crossover and mutation. It serves as the genetic algorithm's foundation, allowing for the systematic exploration of the solution space while adhering to the constraints inherent in the vehicle routing problem (see Fig. 4.1).

Figure 4.1: *Chromosome representation of sample solution*

D	CU	CU	SH	CU	CU	D	SH	CU	CU	CU	CU
1	7	8	17	4	3	2	17	10	9	5	6
D	Driver 1		D	Driver 2		SH	Sharing Point		CU	Customer	

In the given chromosome [1, 7, 8, 17, 4, 3, 2, 17, 10, 9, 5, 6], it represents the following scenario:

Driver 1's path = [1, 7, 8, 17, 4, 3], Driver 2's path = [2, 17, 10, 9, 5, 6]

Driver 1 starts from its initial location and travels to customer 7, then to customer 8. Afterward, it heads to sharing point 17, distributes parcels, and moves on to customer

4, concluding with a delivery to customer 3. Simultaneously, Driver 2 begins at its starting point, heads directly to sharing point 17, distributes parcels, and proceeds to customer 10. It then delivers to customer 9, followed by customers 5 and 6, concluding its route. This interpretation captures the sequence of actions for both drivers, highlighting their distinct paths and adherence to the specified order of visits and sharing point utilization.

In this particular example, both drivers visit a total of four customers, suggesting a symmetric distribution of parcels between them in terms of the number of customers served. It is important to note that in different scenarios, taking into account factors like distance and customer locations, there is a high likelihood that drivers may be assigned varying numbers of customers.

4.1.2 *Fitness Function*

The fitness function remains consistent between the exact method and genetic heuristics. The objective function is determined by the maximum of the total driving time and waiting time for the two drivers. The calculation of driving and waiting costs relies on the assigned paths of both drivers. As clarified in the preceding section, the DCT is equated to the maximum value between driving and waiting costs.

4.1.3 *Initial Population*

The size of the population in a GA is a crucial parameter that involves a careful balance between computational time and the likelihood of discovering superior solutions across a broader solution space. Augmenting the population size typically enhances solution quality by enabling a more comprehensive exploration of potential solutions. However, this enhancement comes at the cost of increased computational effort. It's important to note that the population size is only a fraction of the overall search space. As a general guideline, larger instances of a problem benefit from a larger population, as it provides a

more robust foundation for thorough exploration and better chances of uncovering optimal solutions within the expansive solution space.

In the initial phase of our project, we have opted for an initial population size of 100, a decision influenced by the size and complexity of the problem at hand. This initial population is created by generating 100 arrays of solutions, each representing a potential route assignment for drivers tasked with delivering parcels. The generation process employs a random but feasible chromosome generation approach. Specifically, for the sample instance featuring 8 customers and 16 sharing points, these 100 arrays serve as the starting point for our GA.

In addition, we have introduced a heuristic strategy to establish initial feasible solutions for each potential sharing point. Considering that the sharing process can occur at customer points as well, we have devised a greedy heuristic to systematically derive an initial feasible solution for each potential sharing point. Essentially, following this procedure, we will have a total of 34 initial feasible solutions - 8 corresponding to each of the 8 customers and 16 corresponding to the sharing points. These solutions will subsequently be incorporated into the initial population, contributing to the diverse set of starting points for our algorithmic exploration.

4.1.3.1 Initialization of Feasible Solution

In our proposed approach, the primary objective is to achieve equitable parcel distribution between drivers, ensuring fairness in terms of both driving time and wait time. To realize this goal, we employ a strategy that systematically divides parcels between the two drivers. In the initial step, we construct a Travelling Salesman Problem (TSP) tour encompassing all customers, disregarding sharing points and their locations. With the inclusion of sharing points, our aim is to facilitate a swift sharing process. However, the most expedient meeting point may not always be the optimal strategy. To address this, we

assess the distance of driver 2 to the specified sharing point. Depending on this distance, driver 1 may deliver parcels en route to the sharing point.

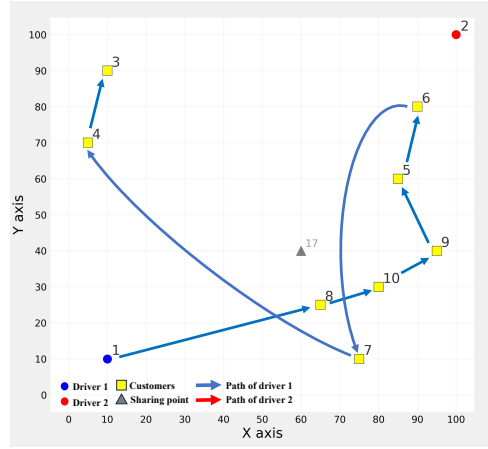
The assumption is that driver 1 can deliver parcels based on the calculated TSP tour. At each step of this tour, driver 1 has the opportunity to divert to the sharing point, where driver 2 awaits. Driver 1 initiates parcel deliveries, and at each juncture, we identify the optimal detour location considering the minimized difference between the path cost of driver 1 to the sharing point and that of driver 2 to the sharing point. This strategic decision minimizes the waiting time for both drivers at the selected sharing point. Additionally, it's essential to note that driver 1 also has the option to directly proceed to the sharing point to meet driver 2. This comprehensive analysis shapes the initial phase of our feasible solution.

Let's illustrate this by sharing point 17. To establish an initial feasible solution, we compute an initial TSP tour for driver 1 utilizing the nearest neighbor method (see Fig. 4.2(a)). The distance from driver 2 to point 17 is determined to be 72. Driver 1 can directly go to point 17, incurring a cost of 58. Subsequently, the driver would wait for 14 time units without delivering any parcels. Alternatively, following the TSP, driver 1 can first go to the nearest neighbor, customer number 8, and then proceed to point 17. The cost of this route is calculated as $57 + 15 = 72$. In this scenario, both drivers reach point 17 theoretically simultaneously, resulting in zero waiting times, as shown in Fig. 4.2(b).

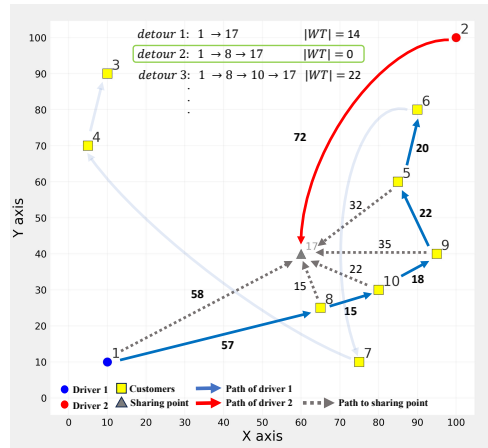
Exploring further, if driver 1 extends the route to another customer, for instance, customer number 7, before heading to point 17, the total cost becomes $57 + 15 + 22 = 94$. Consequently, driver 2 would need to wait for 22 time units. Given this outcome, we opt to truncate the path of driver 1 to point 8 since it minimizes the wait time. This analysis, as illustrated by Fig. 4.2, informs the determination of the optimal route for driver 1 to reach sharing point 17 in our initial feasible solution.

Moving on to the second phase, as demonstrated in Fig. 4.3, our focus shifts to

Figure 4.2: *Initialization of Feasible Solution of GA, phase 1*



(a) TSP tour for driver 1 using NN

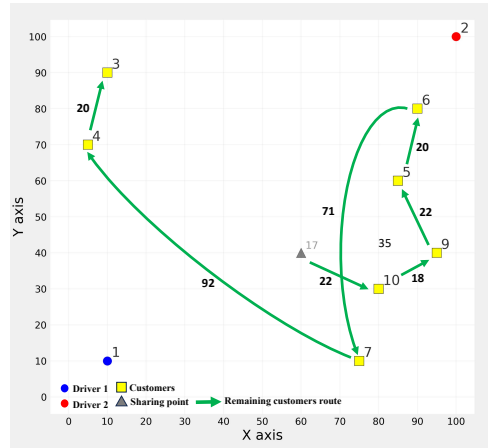


(b) Detours to sharing point

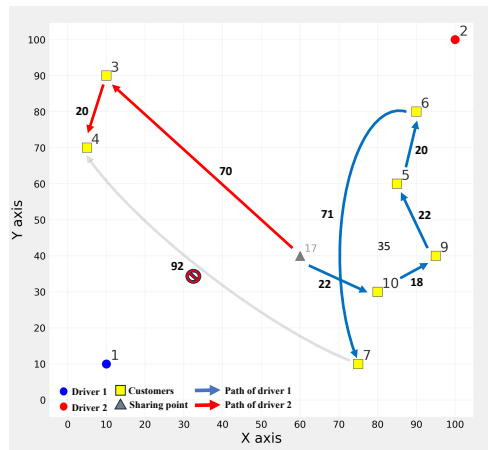
the scenario where both drivers converge at point 17 to deliver the remaining parcels, emphasizing fairness in the distribution. In this iteration, we calculate a second TSP tour for the drivers, akin to a "route first, cluster second" approach commonly employed in VRP. The resulting TSP tour for this phase is depicted in Fig. 4.3(a). We then partition this TSP tour into paths, aiming to minimize the difference between the two paths, as presented in Fig. 4.3(b). These resulting paths are assigned to the two drivers. We adopt

a strategy where the longest route is assigned to the driver who traveled less in the initial step, and conversely, the shortest path is assigned to the driver who had the longest route initially.

Figure 4.3: *Initialization Of Feasible Solution of GA, phase 2, a) TSP tour for remaining points using Nearest Neighbor (NN), b) routes clustering*



(a) TSP tour for remaining points using NN



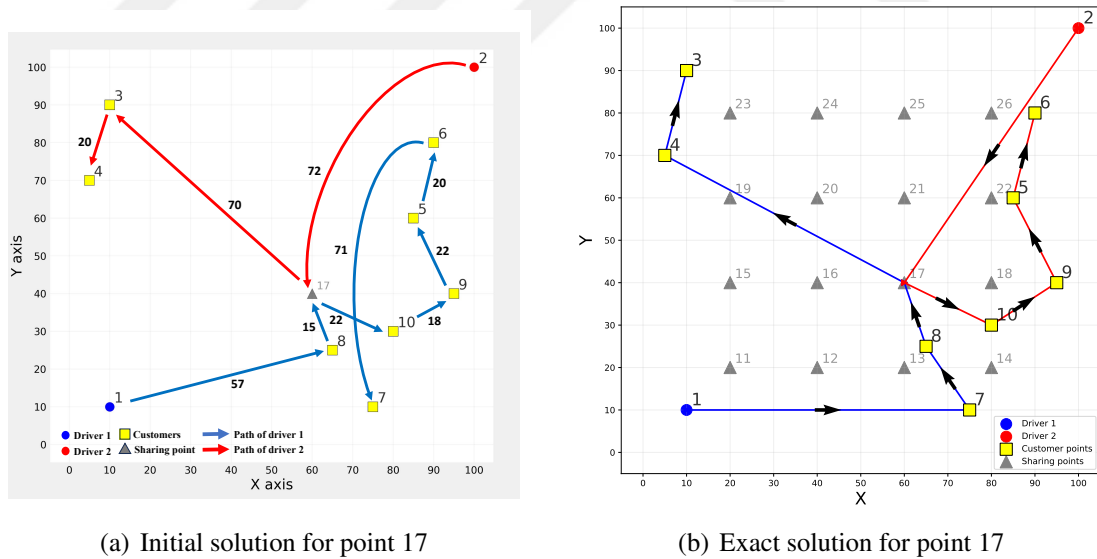
(b) Detours to sharing point

In our specific instance, the lengths of the two paths in the first step were equal to 72. Consequently, the division of the second TSP tour between the two drivers yields no significant difference. The segmented tour between the two drivers is visually presented

in the Fig. 4.3(b). Notably, during this division, the route from point 7 to 4 is deemed unnecessary. Consequently, we allocate customer 3 and 4 to driver 2, and customers 10, 9, 5, 6, and 7 to driver 1. This delineates the optimized division of responsibilities for the remaining parcel deliveries at sharing point 17.

Fig. 4.4 depicts the calculated initial solution and the exact solution for point 17. Fig. 4.4(a) and Fig. 4.4(b) show these results, respectively. As the calculated DCT shows, there is a 20 percent optimality gap (between $DCT = 225$ and $DCT = 180$) considering the initial result. However, as can be seen in Fig. 4.4(a), the main part of this gap is due to the fact that driver 1, based on the NN method, goes to point 8 instead of point 7 first. This is the main reason for the gap, which should be compensated for in further processing using GA by the mutation step, for example.

Figure 4.4: Initialization of feasible solution for GA, a) initial solution for point 17, b) exact solution for point 17



4.1.4 REPRODUCTION: PARENT SELECTION, CROSSOVER, MUTATION

The reproduction process in Genetic Algorithms consists of two key steps: parent selection and GA operators for recombination.

4.1.4.1 Parent Selection

In the parent selection phase, specific individuals are selected from the population to form a mating pool. The selection process may employ various strategies, including fitness proportionate selection, tournament selection, or other heuristic methods. The objective is to prioritize the selection of individuals with higher fitness values, as they are more likely to contribute desirable traits to the next generation. In our proposed strategy, parents are chosen randomly to undergo GA operators. This decision is motivated by our prior truncation of the population, excluding parents with very low fitness values. The purpose is to provide an equal opportunity for other parents to participate in crossover and mutation processes. Once the mating pool is established, genetic operators are applied to facilitate the creation of new individuals for the next generation.

4.1.4.2 Cross Over

Crossover involves the swapping of locations of substrings within binary strings. This mimics the genetic recombination that occurs in nature. By exchanging genetic material between two parents, new individuals or offspring are generated. The specific mechanism of crossover can vary, such as single-point crossover or multi-point crossover, influencing the diversity and characteristics of the resulting offspring.

Several crossover methods for combining two arrays in GAs have been explored [27, 28, 29]. The 1-point crossover involves choosing a random point in the arrays and taking the first part from the first parent and the second part from the second parent. To address feasibility issues, tie-breaker crossover or 2-point crossover (2PX) exchanges the

middle part, with researchers introducing random insertions to improve feasibility [27]. Partially matched crossover Partially matched crossover (PMX) exchanges indexes in the middle part and incorporates missing stops from the matching index in the second parent. Ordered crossover Ordered crossover (OX) and linear ordered crossover linear ordered crossover (LOX) refine 2PX, but they still neglect route considerations. This limitation is addressed by route exchange crossover route exchange crossover (REX) and route-based crossover route-based crossover (RBX), where routes are explicitly considered in the crossover process [27].

However, these methods can be computationally demanding. To strike a balance between computational efficiency and route consideration, researchers turn to the edge recombination crossover (ERX) method. Given the importance of considering neighboring edges, the ERX method is chosen for the multi-depot vehicle routing problem. ERX preserves the logical sequencing of genes from parent chromosomes, ensuring that neighboring relationships are maintained [27]. This approach enhances the quality of solutions and overcomes the limitations of route-based crossovers. Another variation, pheromone-based crossover, extends the consideration beyond neighbors to include lengths and adjacency relations on a global scale when combining genes [27].

Consider these sample parents in Table 4.1:

Table 4.1: *ERX Cross Over Operator*

Parents	
Parent 1	[1, 12, 7, 5, 3, 4, 2, 12, 10, 8, 6, 9]
Parent 2	[1, 9, 11, 6, 3, 10, 5, 2, 11, 7, 4, 8]
Offsprings	
Offspring 1	[1, 9, 12, 6, 2, 12, 8, 4, 3, 10, 5, 7]
Offspring 2	[1, 9, 6, 11, 8, 2, 11, 10, 3, 4, 5, 7]

In the given example, the ERX method is applied to two parents, namely parent 1 and parent 2, each represented as arrays of genes. The resulting offspring arrays, offspring 1 and offspring 2 are generated through the ERX method. For offspring 1, the

ERX method combines genetic material from both parents, considering neighboring relationships and maintaining logical sequencing. Notably, in the modified offspring 1, the sharing point 12 from parent 1 is preserved. Similarly, for offspring 2, the ERX method is applied again, this time preserving sharing point 11 from parent 2 in the modified offspring 2.

It's important to highlight that the ERX method, in this instance, produces only one offspring for each pair of parents. However, since there may be different sharing points for each parent, the process is repeated for each parent separately, taking into account the specific sharing point from each chromosome. In this way, the resulting offspring arrays are tailored to consider the unique characteristics of each parent, ensuring a more customized and context-aware recombination process.

In the first offspring, the drivers and sharing points were inserted exactly as they existed in parent 1. Conversely, the second offspring had its drivers and sharing points inserted based on their indexes as they appeared in parent 2. This approach aimed to retain the favorable traits present in both parents, allowing for a more nuanced recombination process.

During the initial stages, we experimented with placing sharing points randomly within the offspring. However, this introduced a level of randomness that affected the objective function value significantly, resembling a random move. Recognizing the crucial impact of the position of drivers and sharing points on the solution, we opted to preserve their previous locations. To introduce variability, the mutation step was utilized, injecting randomness into the solution and maintaining a balance between preserving parent characteristics and exploring new possibilities.

4.1.4.3 Mutation

Mutation introduces random changes to an individual's genetic material. In the context of binary strings, this typically involves randomly altering the value of one gene (or bit). Mutation serves as a mechanism for introducing novel genetic material into the population and preventing premature convergence to a suboptimal solution. The mutation rate determines the likelihood of a gene being altered. In the mutation process, we have introduced two distinct moves that augment the exploration capabilities of our genetic algorithm.

- **Swap Move Mutation:** This mutation involves the exchange of nodes with their neighboring nodes within the route. Specifically, a node can be swapped with the node before or after its current location. Considered a minor mutation, the swap move contributes to the intensification of solutions by allowing nodes to subtly adjust their positions within the route. For example, consider this sample move shown in Table 4.2: node 7 is swapped with its neighboring node, 5, refining the existing route configuration.

Table 4.2: *Swap Move Mutation Sample*

Input chromosome	[1, 3, 11, 4, 7, 5, 2, 11, 9, 8, 10, 6]
Randomly selected node	7
After swap move mutation	[1, 3, 11, 4, 5, 7, 2, 11, 9, 8, 10, 6]

- **Remove and Insert Move Mutation:** In contrast to the swap move, this mutation adopts a more flexible approach. A random node is selected, removed from its current location, and then inserted into a random index within the same route. Unlike the swap move, this mutation is not restricted to neighboring nodes, providing greater freedom in reshuffling the configuration of nodes within a route. The remove and insert move aims to introduce diversity into the solution space, exploring

different node configurations to potentially discover more optimal solutions. For example, consider this sample move shown in Table 4.3: node 7 (index 4) is selected, removed, and inserted into a new random index (index 2), introducing flexibility and diversification into the solution space.

Table 4.3: *Remove-Insert Move Mutation Sample*

Input chromosome	[1, 3, 11, 4, 7, 5, 2, 11, 9, 8, 10, 6]
Randomly selected node	7
Random insertion index	3
After removing and inserting move	[1, 3, 7, 11, 4, 5, 2, 11, 9, 8, 10, 6]

By incorporating these two mutation moves, our strategy seeks to strike a balance between intensification and diversification. This enhancement allows the genetic algorithm to navigate the solution space more effectively, uncovering a wider range of potential solutions and ultimately improving the overall optimization process.

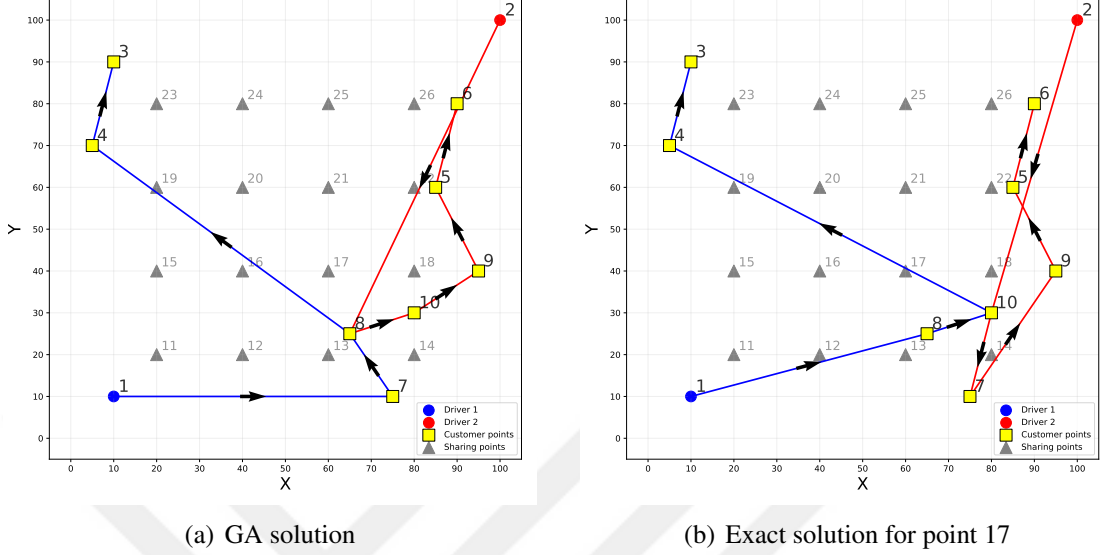
4.1.5 PRELIMINARY RESULTS ON BENCHMARK INSTANCES

In the initial phase of our proposed algorithm application on sample instances, a routing plan with promising results was obtained within a runtime of 43.24 seconds. The best feasible individual identified during this process is represented by the array:

Best feasible solution = [1, 7, 32, 8, 4, 3, 2, 32, 10, 9, 5, 6]

Fig. 4.5 compares the GA' result and the exact solution for the hypothetical delivery. The selected sharing point ID is 32, denoting customer 8 as the sharing point, as shown in Fig. 4.5(a). Interestingly, it is observed that customer 8 is visited immediately after sharing point 33, constituting the same location with zero distance. The visual representation of this routing plan can be seen in Fig. 4.5(b). The objective function, measured as the DCT, is determined to be 178 ($DCT = 178$). Notably, this DCT is only one unit higher than the exact solution, which was 177, achieved when sharing occurred at customer point 10.

Figure 4.5: Preliminary analysis results, a) GA solution, b) exact solution



Upon initial analysis, these results indicate a theoretical optimality gap of 0.6% for this specific sample instance. While this suggests a degree of optimism in achieving similar gaps for larger instances, it is crucial to recognize that further analysis and refinement of the algorithm are required. It is worth noting that the problem was addressed holistically, considering all points (both customers and sharing points) in a single run, without running the algorithm independently for each sharing point. This highlights the correctness of the algorithm's implementation, paving the way for future modifications and in-depth analysis to enhance its performance in more complex scenarios.

4.2 Modelling OSP problem using Google Routing Solver

The Google OR-Tools package was proposed to solve the OSP problem for a large-scale network. Google OR-Tools is a versatile open-source software suite designed to solve combinatorial optimization problems efficiently [30]. In Google routing solver, local search methods such as simulated annealing, tabu search, and guided local search are employed to enhance the initial solutions generated for routing problems during the

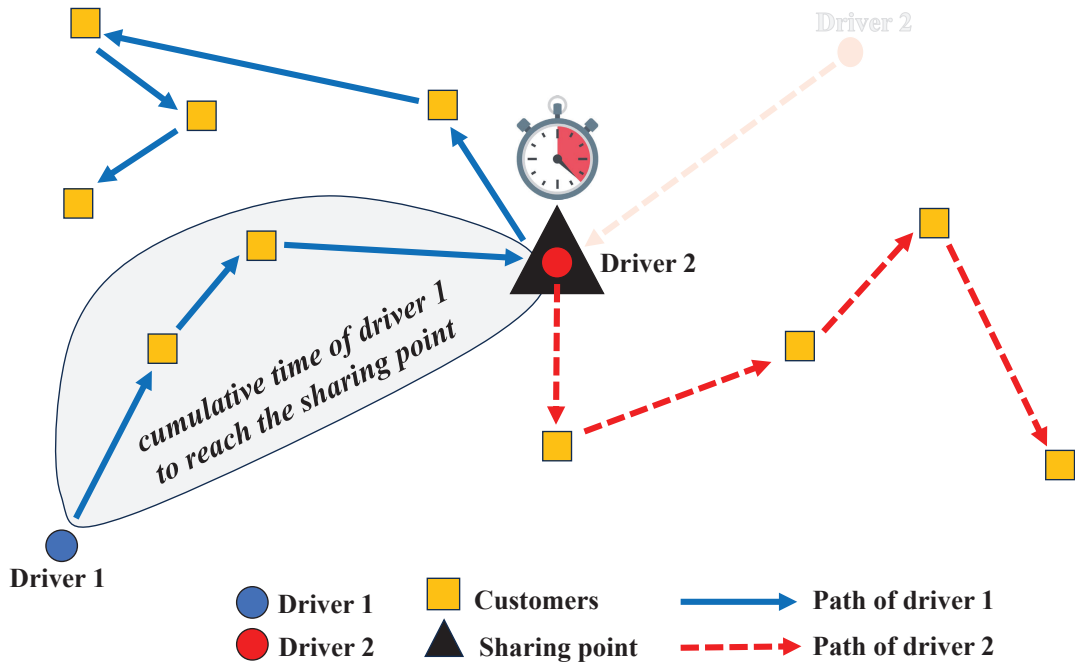
initial step [30, 31].

However, solving the OSP problem formulated earlier is not straightforward using Google OR-Tools VRP solver because while the OSP problem falls within the broader category of VRP, it possesses unique characteristics that distinguish it from the common VRP formulation. In contrast to the conventional VRP, where all nodes must be visited, our approach considers sharing points from which only one must be visited. Nevertheless, when only one sharing point is considered, the problem bears a resemblance to an open-VRP scenario: two drivers that depart from their origins, similar to a multi-depot VRP case, and go to the sharing point and then deliver parcels. This open-VRP scenario can be modeled and solved using the Google solver by incorporating additional constraints to the generic VRP objective function.

As shown in Fig. 4.6, to ensure that the sharing point is the first location visited by driver 2, we modified the path by letting driver 2 start from the sharing point with a fixed initial cost equal to the duration from its origin to the selected sharing point. Driver 1 would start from its origin and must visit the sharing point as a mandatory destination along its path. Note that the redefinition of the problem in this manner does not alter the original problem setup. After redefinition of the problem, considering one sharing point at a time, our approach aligns with the multi-depot open-VRP framework.

In the new VRP problem configuration with two drivers, it is essential to model the WT of the drivers. Driver 1, while traveling to the sharing point, has the flexibility to choose the path, yet can arrive too early or too late. To address this issue, soft lower and upper bounds are employed to account for the WT by utilizing the methods *SetCumulVarSoftLowerBound* and *SetCumulVarSoftUpperBound* available in the Google OR-Tools routing module. These functions establish flexible bounds for the cumulative variable associated with a specific index. Whenever the value of the cumulative variable exceeds these bounds, a cost proportional to the difference between the value and the bound is

Figure 4.6: *Modelling OSP for a single sharing point*



incorporated into the overall cost function. In our case, the soft lower and upper bounds are set equal to the travel time from driver 2's origin to the sharing point.

To reduce the maximum delivery time for drivers, the *SetGlobalSpanCostCoefficient* method is employed, which allows for setting a coefficient that influences the global span of the routes, where the global span represents the maximum delivery time among the routes. By assigning greater importance to the global span in the objective function, the solver focuses on minimizing the length of the longest route.

By considering both WT and the length of the longest route, the solver tries to minimize the cumulative time, calculated by the following objective function:

$$\begin{aligned}
\text{Objective function} = & \text{summation of all delivery times} + \\
& \text{coefficient}_1 (|\text{DT of driver 1 to OSP} - \text{DT of driver 2 to OSP}|) + \\
& \text{coefficient}_2 (\text{maximum delivery times among two drivers})
\end{aligned} \tag{4.1}$$

In our model, unity is assigned to the coefficients, i.e., $\text{coefficient}_1 = \text{coefficient}_2 = 1$, ensuring that each coefficient exerts an equal influence on the objective value. Once the model is configured, the VRP problem is solved for each customer point and potential sharing point. In the next section, we delve into a real-world application of parcel delivery to illustrate the practical implications of the proposed method.

5. CASE STUDY AND RESULTS

This chapter presented a case study applying both the GA and Google VRP solver to the previously defined hypothetical delivery scenario. By comparing and analyzing the results from each approach, the chapter assessed their effectiveness in solving the OSP problem.

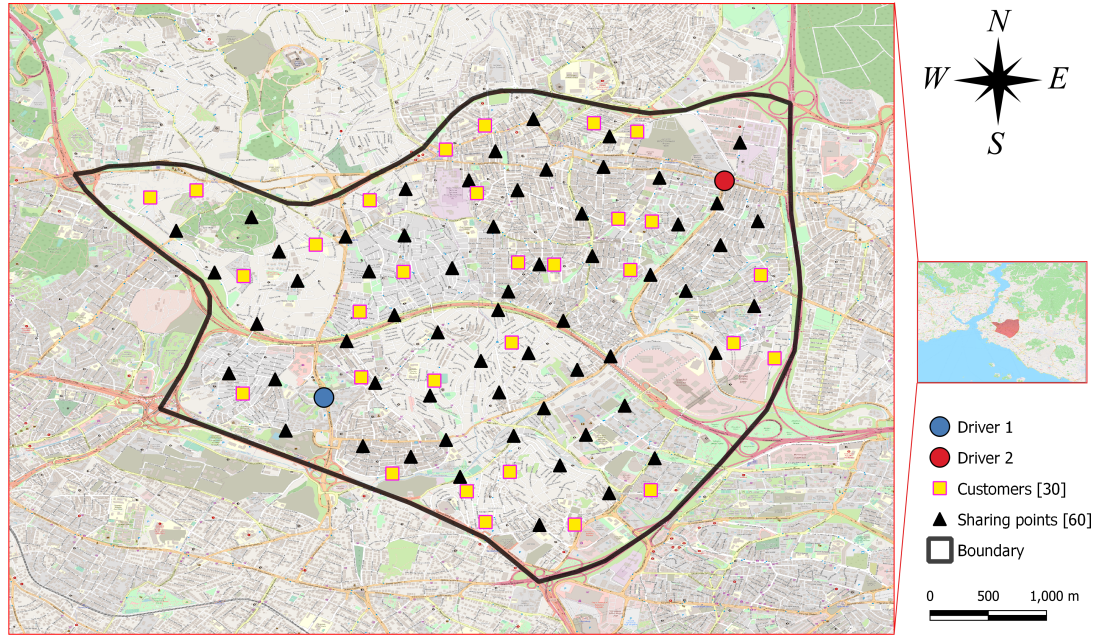
5.1 Case Study

The case study area, shown in Fig. 5.1, covers two neighborhoods, namely Umraniye and Atasehir, located on the Asian side of Istanbul, Turkey, spanning approximately 21.8 square kilometers. These neighborhoods were chosen due to their high population density, encompassing a mix of residential and commercial areas.

The candidate sharing points were identified as convenient and safe locations where drivers could stop and divide parcels without requiring any special facilities. Moreover, the density and dispersion of the selected sharing points are crucial factors. It is essential to strike a balance to avoid problems associated with points being either too close or too far from each other.

To determine the appropriate number of sharing points, we employed isochrone maps, which visually depict travel time as a key factor in the selection process. Isochrone maps are graphical representations that display areas of equal travel time or travel distance from a specific location. These maps provide valuable insights into accessibility and transportation patterns by illustrating how far one can travel within a given time frame or distance from a central point. After conducting a visual inspection, it was determined that approximately 60 points, with a travel time of two minutes, would effectively cover the selected area. This selection implies that drivers will have access to a sharing point within the study area that can be reached in less than two minutes. This selection of a two-minute threshold strikes a reasonable balance and guarantees that drivers will

Figure 5.1: *Map of the case study area*

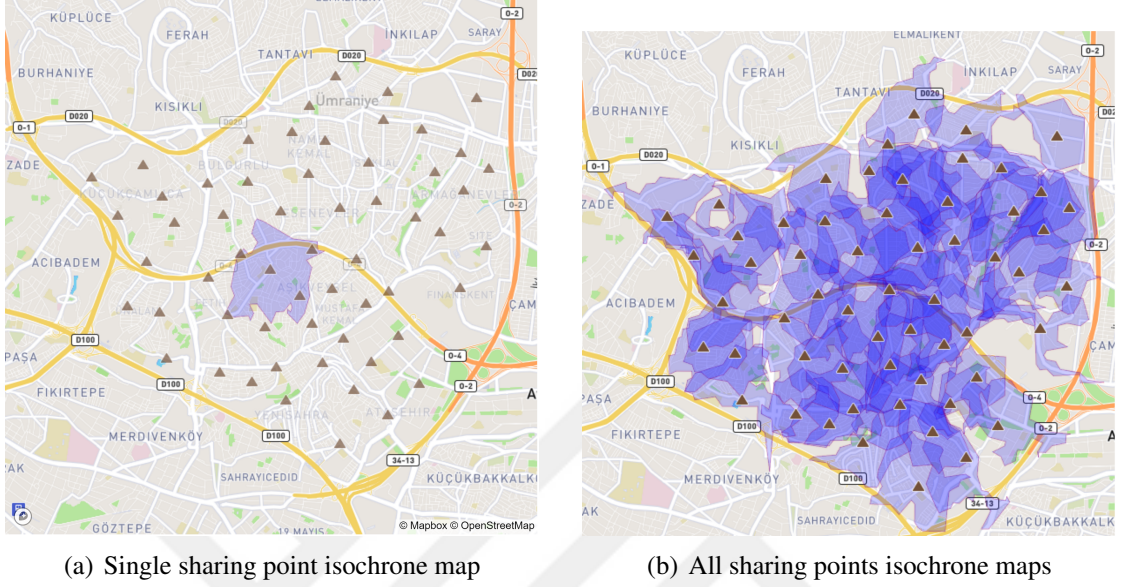


have sufficient nearby points to reach within a practical timeframe. Fig. 5.2 depicts the isochrone map, considering a two-minute travel time threshold, for a sample sharing point (see Fig. 5.2(a)) and all overlapping sharing points across the entire neighborhood (see Fig. 5.2(b)).

A total of 30 customers were selected for our analysis. The customer locations can be distributed anywhere within the designated area. In considering the initial locations of the two drivers, we intentionally selected positions that were relatively distant from each other. This decision was made to explore scenarios where finding the OSP would not be straightforward. If the drivers were in close proximity, the solver would select the closest sharing point to share the parcels as soon as possible.

To utilize GA or Google VRP solver for our problem, a travel time matrix was required. Possible options for obtaining this matrix included the Google Distance Matrix API [32], Mapbox Matrix API [33], and others. The Mapbox API was chosen for this

Figure 5.2: Two-minute isochrone maps, a) for a single sharing point, b) the entire area covered by all sharing points



study as it incorporates live and typical traffic data. Live traffic data are updated every five minutes, and typical traffic data provide estimated speeds throughout the week derived from historical observations [33]. However, since live traffic data are yet to be available for Istanbul, we chose the historical traffic data option.

5.1.1 Analysis Results

GA and Google VRP solver were used for the OSP problem for 30 customers and 60 sharing points. To use Google VRP solver, after decomposing the main problem into subproblems of open-VRP for each potential sharing point, 90 instances of the VRP problem needed to be solved. The cost matrix was calculated using Mapbox API; based on that, GA solved the problem considering all points in a single run. However, the Google VRP solver identified the optimal path for each input point. The code's runtime depends on the computation of the cost matrix and solving the VRP instances. The proposed method was developed in Python and ran on an Intel(R) Core (TM) i7-1065G7 CPU at

1.3 GHz with 16 GB of RAM on Windows 10. In our specific setup, calculating the cost matrix using the Mapbox API took approximately 10 seconds. For the GA, we considered a one-minute limit to run. Subsequently, Google VRP solver utilized this calculated cost matrix to solve the problem for all 90 points in around 30 seconds. Consequently, the OSP can be determined in around one minute.

5.1.2 GA results

To account for the stochastic nature of the genetic algorithm, we executed the GA five times, aiming to provide a comprehensive overview of the algorithm's mean accuracy in different repetitions. Table 5.1 contains the outcomes from the five independent runs of the GA.

Table 5.1: *Costs of Sharing Plan Found by GA (All Units are in Seconds)*

Iteration	DCT	Paths
1	3976	[1, 32, 22, 78, 4, 3, 8, 29, 19, 16, 15, 28, 9, 27, 30, 5, 17, 10, 2, 78, 12, 11, 20, 25, 24, 21, 23, 26, 7, 14, 13, 31, 6, 18]
2	3976	[1, 32, 22, 78, 4, 3, 8, 29, 19, 16, 15, 28, 9, 27, 30, 5, 17, 10, 2, 78, 12, 11, 20, 25, 24, 21, 23, 26, 7, 13, 14, 6, 18, 31]
3	3915	[1, 22, 32, 44, 11, 12, 20, 25, 24, 21, 23, 26, 7, 13, 6, 14, 31, 18, 2, 44, 4, 3, 8, 29, 19, 16, 15, 28, 9, 27, 30, 5, 17, 10]
4	3976	[1, 32, 22, 78, 11, 12, 20, 25, 24, 23, 21, 26, 7, 13, 14, 6, 31, 18, 2, 78, 4, 3, 8, 29, 19, 16, 15, 28, 9, 27, 30, 5, 17, 10]
5	3956	[1, 22, 78, 4, 3, 8, 29, 19, 16, 15, 28, 9, 27, 30, 5, 17, 10, 2, 78, 12, 11, 20, 25, 24, 21, 23, 26, 32, 7, 14, 13, 6, 31, 18]
Average	3960	

After multiple iterations, our GA method consistently yields an average DCT of 3,960. Fig. 5.3 shows the result of the GA, where point 78 was selected as the OSP with a corresponding $DCT = 3,976$.

Figure 5.3: Average GA result



5.1.3 Google Solver Results

To examine the results and efficiency of the model under different scenarios, the following two scenarios were considered, and the model's outcomes were analyzed. These scenarios were based on the emphasis given to the model's priority in optimizing the minimizing of the duration of the longest route or the total time of two drivers' paths

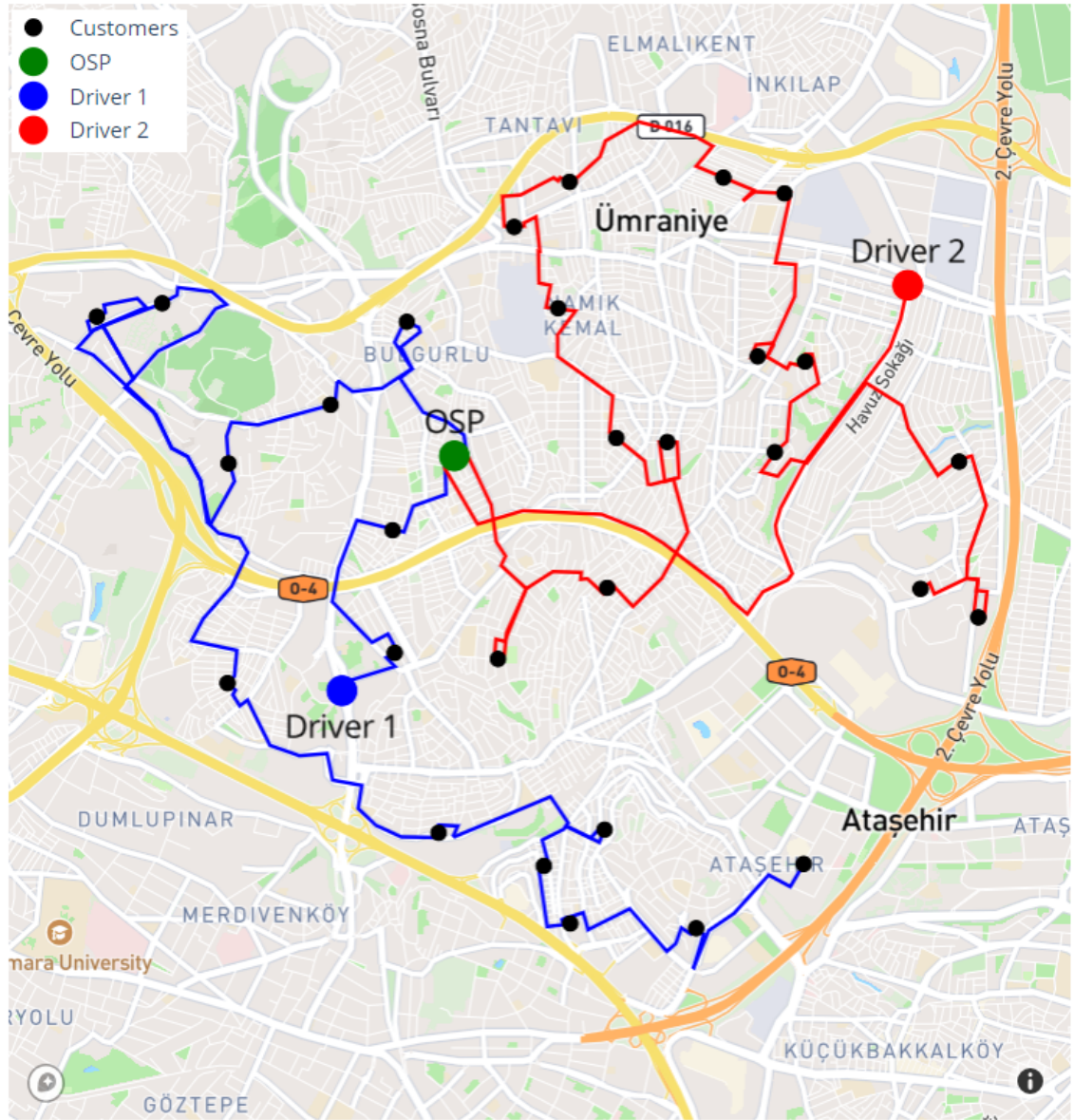
or by adjusting the value of the parameter *GlobalSpanCostCoefficient*. In Scenario I, the main proposed model, shown in Fig. 5.4, the value of *GlobalSpanCostCoefficient* was set to unity in the model settings. This setting allowed the model to consider the sum of both delivery times and the maximum delivery time of the two drivers, ensuring fairness between the two drivers in delivering parcels.

In this case, the total delivery time for all parcels, considering both DT and WT, amounted to 6,704 seconds, with 3,368 seconds for driver 1 and 3,336 seconds for driver 2, respectively. The delivery time difference of 32 seconds between the two drivers is negligible in practice, and the delivery task was completed at $DCT = 3,368$.

In Scenario II, shown in Fig. 5.5, the fairness constraint was removed by setting the value of *GlobalSpanCostCoefficient*, i.e., *coefficient2* in Equation 4.1, to zero. Consequently, the model did not consider the maximum delivery time when calculating the objective value. In this case, the total delivery time for all parcels amounted to 6,694 seconds, with 2,167 seconds for driver 1 and 4,527 seconds for driver 2. Comparatively, the summation of DT and WT decreased by 10 seconds compared to Scenario I. However, the delivery time difference between the two drivers is 2,360 seconds, and clearly, a balance in driving time was not achieved, as was the case in Scenario I. The results of Scenario I and Scenario II are summarized in Table 5.2.

The total delivery time for the two scenarios examined above, they are 6,704 and 6,694, respectively. These values are not significantly different from each other, and at first glance, it might seem that there is not much difference between the two optimization scenarios. In fact, one might favor Scenario II as it produces lower total cost, a common objective in the relevant literature [15, 19]. However, even though the overall costs may not show a significant difference in total cost, the proposed routes, generated under different priorities, have noticeable distinctions. As a result, in Scenario I, not only were the parcels delivered 1,210 seconds faster compared to Scenario II, but it also resulted in a

Figure 5.4: Optimal sharing plan, Senario I, considering fairness between the two drivers

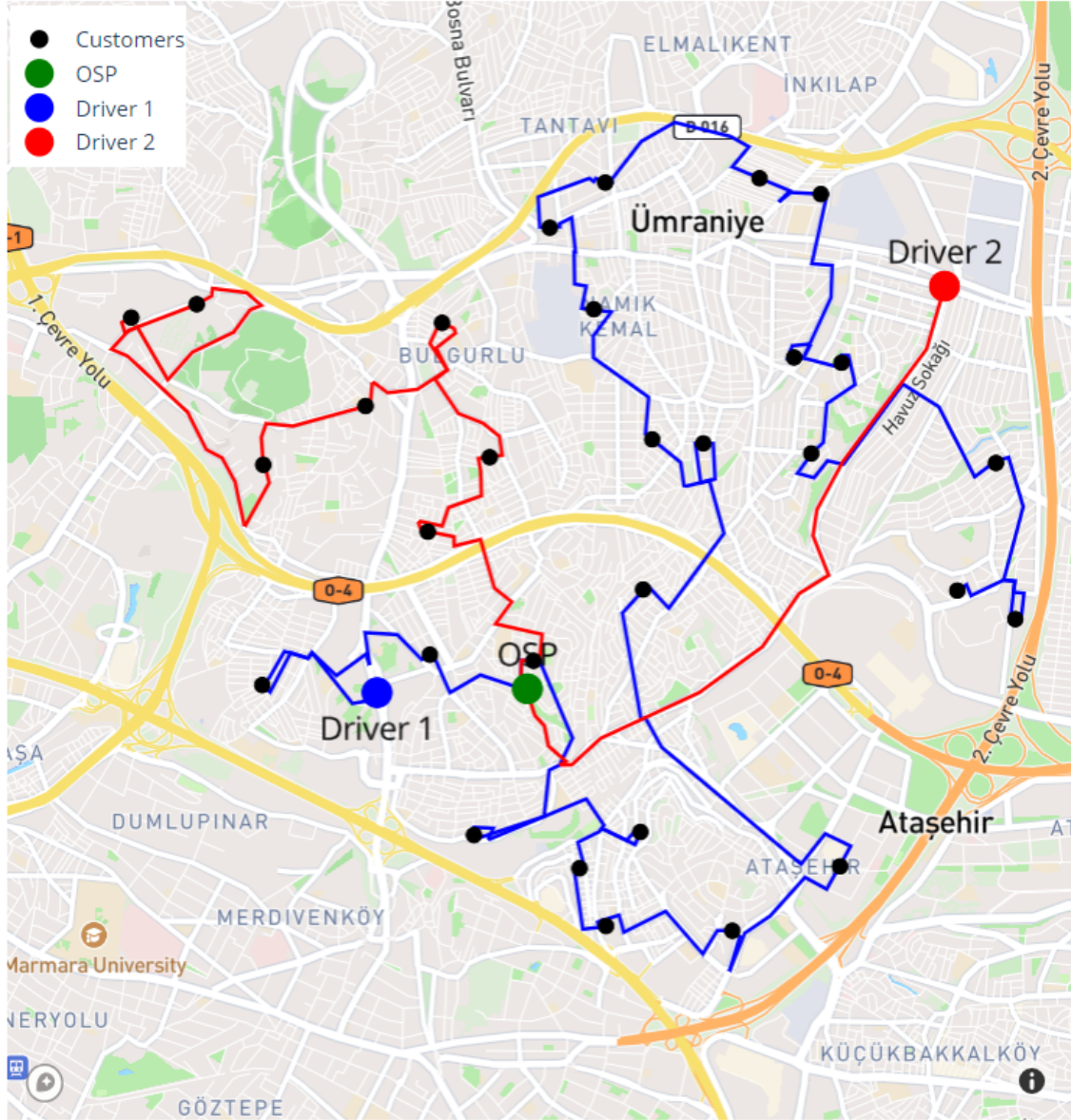


more balanced DT among drivers. A breakdown of costs for each scenario is provided in Table 5.2.

Table 5.2: Details of Costs for Each Considered Plan (All Units Are in Seconds)

Optimization Scenario	Driver 1			Driver 2			Total			DCT
	DT ₁	WT ₁	Total	DT ₂	WT ₂	Total	DT	WT	DT+WT	
Scenario I	3,270	98	3,368	3,336	0	3,336	6,606	98	6,704	3,368
Scenario II	2,051	116	2,167	4,527	0	4,527	6,578	116	6,694	4,527

Figure 5.5: Sharing plan, Senario II, focusing on minimizing the summation of delivery times



5.1.4 Google Solver vs. GA Results

When comparing this result with the output from the Google VRP solver, we observe a minor optimality gap of approximately 0.015. Notably, Google identifies point 58 as the sharing point, while our GA solver designates point 78 as the OSP. Figure 8

shows the Google and GA plan for delivering the parcels.

In some runs, our GA method identified point 58 with a DCT of 3,900 as the OSP, which we initially considered as a "chance" discovery. Through meticulous parameter tuning and the implementation of refined exploration and exploitation strategies, we aimed to align our GA's objective function results more closely with those of Google's solver. It's crucial to note that Google's solver, while offering speed advantages, doesn't guarantee optimality, leaving room for potentially better solutions.

One notable positive aspect of Google's solver lies in its speed, making it advantageous for quickly solving VRP-like problems with satisfactory results. However, its drawback is the challenge of incorporating user-defined and problem-specific constraints, posing limitations on the customization of constraints within the solver.

6. CONCLUSIONS

To sum up, this study addressed the problem of determining the OSP for two delivery drivers. The problem was formulated as a MILP model and solved using the Pyomo package [4] and CPLEX solver in Python. The results were evaluated through a hypothetical delivery scenario, and the exact method was applied to a small-scale dataset. Additionally, the problem was reformulated as a multi-depot open-VRP to address real-world scenarios with larger scales and solved using the Google OR-Tools routing solver. The case study in Istanbul, Turkey, accounted for network travel time and investigated various scenarios to identify the optimal OSP and devise effective plans to minimize DCT. Different priorities were considered in each scenario, leading to the selection of distinct optimal sharing points and corresponding optimized paths.

In addressing the OSP problem, we leveraged the speed advantage of Google's solver and devised a penalty mechanism within our model. The subsequent comparison with our GA results demonstrated that Google achieved commendable outcomes. Importantly, our developed GA exhibits the flexibility to easily integrate user-defined constraints and auxiliary objective functions. Despite a marginal optimality gap of only 1.5% with Google's solver on average, this flexibility showcases the potential to extend the GA's applicability to more complex problems involving additional drivers and constraints. This underlines the adaptability and versatility of our GA method in addressing diverse problem scenarios. While the primary focus was on practicality and efficiency rather than absolute optimality, the model provided a reasonable solution within a manageable time-frame. The problem's uniqueness lies in the context of q-commerce and fast delivery applications, where timely delivery is critical to customer satisfaction. In contrast to generic models that consider transfer nodes, our approach accounted for specific constraints and objectives. Fast delivery services often deal with time-sensitive items. Consequently, fast delivery is essential to meet customer expectations. In many cases, customers are willing

to pay a premium for faster delivery, making delivery time a vital factor. The proposed approach successfully minimized the DCT to meet these expectations by considering real-world scenarios, actual network topology, and a reasonable number of sharing points. Notably, the model required no physical transfer nodes, allowing drivers to share parcels efficiently without the need for temporary storage. Moreover, isochrone maps were employed to strike a balance between the count and dispersion of sharing points, ensuring their accessibility within a reasonable time. Using Google OR-Tools as a routing solver facilitated solving VRP instances in large-scale scenarios within a reasonable timeframe. In summary, our approach addressed these key objectives:

- Minimizing DCT to prioritize timely services.
- Implementing a simultaneous sharing process without the need for physical transfer nodes.
- Incorporating actual network topology and traffic considerations.
- Utilizing isochrone maps to determine the reasonable count and distribution of sharing points.
- Leveraging metaheuristic methods for efficient problem-solving in large real-world scenarios.
- Adding flexibility to the sharing process by considering customer nodes as additional potential sharing points.

The model's flexibility allows users to adjust parameters and coefficients to tailor the solution for different objectives. Whether emphasizing total delivery time or minimizing driving time for drivers, the model can adapt to varying delivery scenarios. For example, as expressed in Scenario II, by assigning different model parameters to model coefficients, the model was given priority to minimize the total delivery time rather than the DCT.

The proposed practical approach presents a flexible and efficient solution for determining the OSP in fast delivery services. The model offers valuable insights to enhance fast delivery operations by considering real-world constraints and time-sensitive requirements.



REFERENCES

- [1] T. Reardon, F. Tomatis, and E. Pedersen, “Evolution of food e-commerce during the covid-19 pandemic,” 2023.
- [2] “Shared delivery service for sme’s using the uber model.” <https://www.climatecolab.org/contests/2018/helping-small-medium-businesses-go-low-carbon-ontario/phase/1319074/proposal/1334498>, Accessed: 2024-10-13.
- [3] M. W. Savelsbergh and M. Sol, “The general pickup and delivery problem,” *Transportation science*, vol. 29, no. 1, pp. 17–29, 1995.
- [4] “Pyomo.” <http://www.pyomo.org/about>. Accessed: 2024-12-10.
- [5] M. Guajardo and M. Rönnqvist, “A review on cost allocation methods in collaborative transportation,” *International transactions in operational research*, vol. 23, no. 3, pp. 371–392, 2016.
- [6] F. Cruijssen, O. Bräysy, W. Dullaert, H. Fleuren, and M. Salomon, “Joint route planning under varying market conditions,” *International Journal of Physical Distribution & Logistics Management*, vol. 37, no. 4, pp. 287–304, 2007.
- [7] J. Los, F. Schulte, M. Gansterer, R. F. Hartl, M. T. Spaan, and R. R. Negenborn, “Large-scale collaborative vehicle routing,” *Annals of Operations Research*, pp. 1–33, 2022.
- [8] Q. Zhang, Z. Wang, M. Huang, Y. Yu, and S.-C. Fang, “Heterogeneous multi-depot collaborative vehicle routing problem,” *Transportation Research Part B: Methodological*, vol. 160, pp. 1–20, 2022.
- [9] C. E. Cortés, M. Matamala, and C. Contardo, “The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method,” *European Journal of Operational Research*, vol. 200, no. 3, pp. 711–724, 2010.
- [10] R. Masson, F. Lehuédé, and O. Péton, “Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers,” *Operations Research Letters*, vol. 41, no. 3, pp. 211–215, 2013.
- [11] G. Ciobotaru and S. Chankov, “Towards a taxonomy of crowdsourced delivery business models,” *International Journal of Physical Distribution & Logistics Management*, vol. 51, no. 5, pp. 460–485, 2021.
- [12] C. Archetti, M. Savelsbergh, and M. G. Speranza, “The vehicle routing problem with occasional drivers,” *European Journal of Operational Research*, vol. 254, no. 2, pp. 472–480, 2016.

- [13] V. Carbone, A. Rouquet, and C. Roussat, “The rise of crowd logistics: a new way to co-create logistics value,” *Journal of Business Logistics*, vol. 38, no. 4, pp. 238–252, 2017.
- [14] Z. Lyu and A. J. Yu, “The pickup and delivery problem with transshipments: Critical review of two existing models and a new formulation,” *European Journal of Operational Research*, vol. 305, no. 1, pp. 260–270, 2023.
- [15] A. Sampaio, M. Savelsbergh, L. P. Veelenturf, and T. Van Woensel, “Delivery systems with crowd-sourced drivers: A pickup and delivery problem with transfers,” *Networks*, vol. 76, no. 2, pp. 232–255, 2020.
- [16] A. Rais, F. Alvelos, and M. S. Carvalho, “New mixed integer-programming model for the pickup-and-delivery problem with transshipment,” *European Journal of Operational Research*, vol. 235, no. 3, pp. 530–539, 2014.
- [17] J. S. Shang and C. K. Cuff, “Multicriteria pickup and delivery problem with transfer opportunity,” *Computers & Industrial Engineering*, vol. 30, no. 4, pp. 631–645, 1996.
- [18] S. Mitrović-Minić and G. Laporte, “The pickup and delivery problem with time windows and transshipment,” *INFOR: Information Systems and Operational Research*, vol. 44, no. 3, pp. 217–227, 2006.
- [19] R. Masson, F. Lehuédé, and O. Péton, “The dial-a-ride problem with transfers,” *Computers & Operations Research*, vol. 41, pp. 12–23, 2014.
- [20] Z. Fu and J. Y. Chow, “The pickup and delivery problem with synchronized en-route transfers for microtransit planning,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 157, p. 102562, 2022.
- [21] Y. Hou, W. Zhong, L. Su, K. Hulme, A. W. Sadek, and C. Qiao, “Taset: Improving the efficiency of electric taxis with transfer-allowed rideshare,” *IEEE transactions on vehicular technology*, vol. 65, no. 12, pp. 9518–9528, 2016.
- [22] M. A. N. Andini, Y. Satria, and H. Burhan, “Dynamic pickup and delivery problem with transfer in ridesharing to reduce congestion,” in *Journal of Physics: Conference Series*, vol. 1218, p. 012010, IOP Publishing, 2019.
- [23] D. Wang, Q. Wang, Y. Yin, and T. Cheng, “Optimization of ride-sharing with passenger transfer via deep reinforcement learning,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 172, p. 103080, 2023.
- [24] M. Mahmoudi, J. Chen, T. Shi, Y. Zhang, and X. Zhou, “A cumulative service state representation for the pickup and delivery problem with transfers,” *Transportation Research Part B: Methodological*, vol. 129, pp. 351–380, 2019.

- [25] S. Voigt and H. Kuhn, “Crowdsourced logistics: The pickup and delivery problem with transshipments and occasional drivers,” *Networks*, vol. 79, no. 3, pp. 403–426, 2022.
- [26] J. H. Holland, “Genetic algorithms and the optimal allocation of trials,” *SIAM journal on computing*, vol. 2, no. 2, pp. 88–105, 1973.
- [27] S. Karakatič and V. Podgorelec, “A survey of genetic algorithms for solving multi depot vehicle routing problem,” *Applied Soft Computing*, vol. 27, pp. 519–532, 2015.
- [28] P. Graglia, N. Stark, H. Alfonso, and C. Salto, “A comparison of recombination operators for capacitate vehicle routing problem,” in *XIV Congreso Argentino de Ciencias de la Computación*, 2008.
- [29] J. Ochelska-Mierzejewska, A. Poniszewska-Maraña, and W. Maraña, “Selected genetic algorithms for vehicle routing problem solving,” *Electronics*, vol. 10, no. 24, p. 3147, 2021.
- [30] “Google or-tools.” <https://developers.google.com/optimization>. Accessed: 2024-11-06.
- [31] T. Cuvelier, F. Didier, V. Furnon, S. Gay, S. Mohajeri, and L. Perron, “Or-tools’ vehicle routing solver: a generic constraint-programming solver with heuristic search for routing problems,” in *24e congrès annuel de la société française de recherche opérationnelle et d’aide à la décision*, 2023.
- [32] “Google maps platform.” <https://developers.google.com/maps>. Accessed: 2024-11-01.
- [33] “Mapbox.” <https://www.mapbox.com/>. Accessed: 2025-02-01.