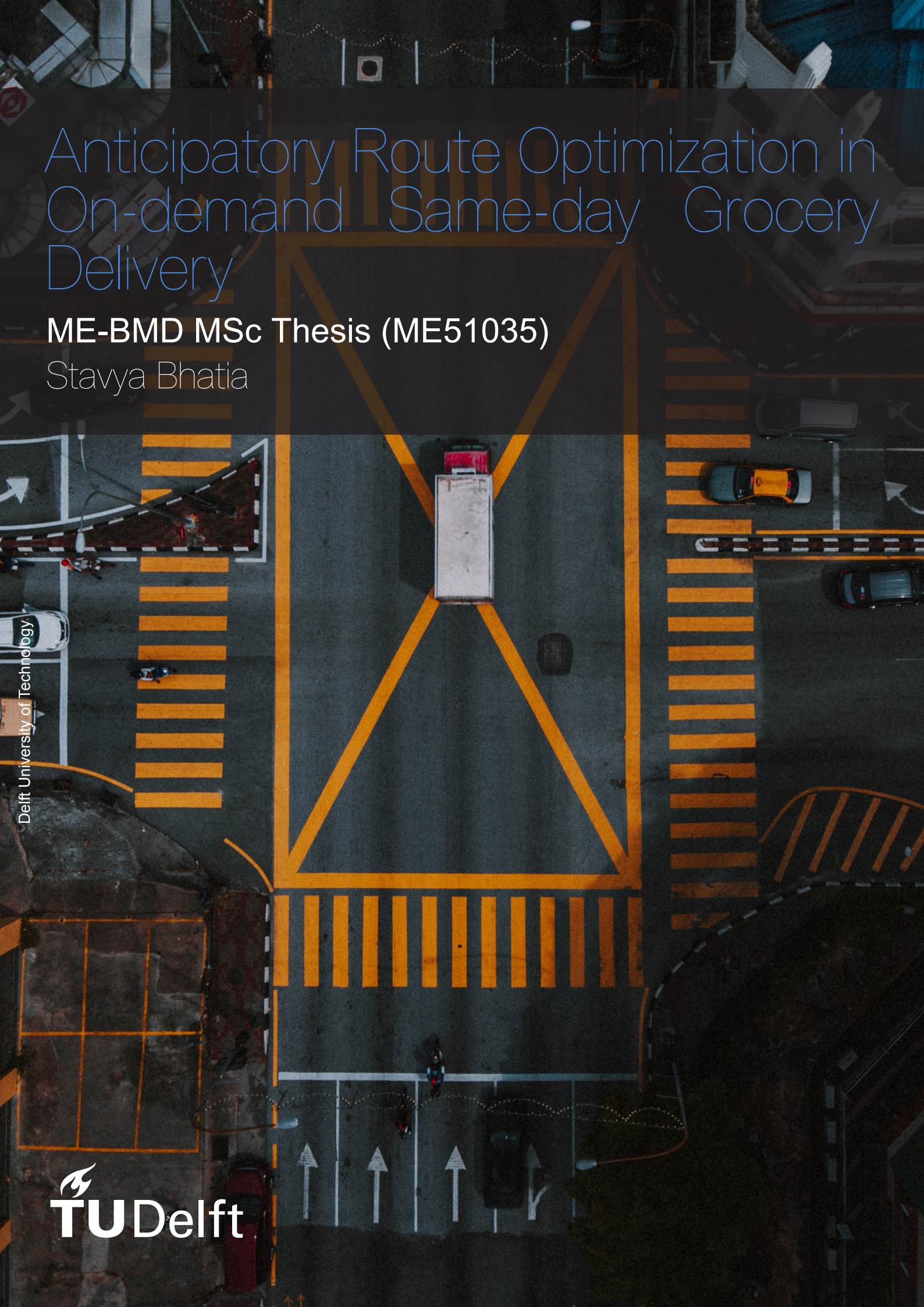
Anticipatory Route Optimization in On-demand Same-day Grocery Delivery

ME-BMD MSc Thesis (ME51035)

Stavya Bhatia



Delft University of Technology

TU Delft

Anticipatory Route Optimization in On-demand Same-day Grocery Delivery

by

Stavya Bhatia

Supervisor:	Dr. Ir. Javier Alonso-Mora
PhD Supervisor:	Maximilian Kronmueller
Institution:	Delft University of Technology
Place:	Faculty of Mechanical, Maritime and Material Engineering, Delft
Project Duration:	July, 2021 - March, 2022

TU Delft

Abstract

The recent decade has seen exponential growth in technology and information processing. This has enabled a paradigm shift in several logistical operations such as the introduction of on-demand same-day delivery of groceries.

One of the most relevant challenges in dynamic same-day pickup & delivery systems is related to uncertainty about future orders. Lack of knowledge about emerging orders results in the selection of routes that are optimal till such time additional information is not available and sub-optimal thereafter. This induces a mismatch between the routes of the vehicles in the future and the origins of the emerging orders. In our thesis, we introduce simple anticipatory techniques that solve this problem and can scale to large problem instances of thousands of orders. In particular, our techniques utilize endogenous properties of the problem to affect both how vehicles are assigned to orders, and how to route vehicles to serve those orders. One of our techniques introduces rewards that reduce the cost of assignment between a vehicle and a group of orders if the vehicle is routed towards a favourable zone. A favourable zone can be a region with more number of orders that can be picked up at its nearest depots, or a region whose distance to the nearest depots is lower than others, etc. Another technique penalizes assignment between a vehicle and a group of orders if the vehicle is routed away from a high-demand zone and vice versa. We propose, formally discuss and experimentally evaluate several formulations of both rewards adjustment and adjustment with penalty + rewards.

We test our techniques in combination with the state-of-the-art Vehicle Group Analysis (VGA) framework in Amsterdam for a fleet of 10 vehicles and up to 3600 grocery orders. We further conduct extensive computation tests with varying hours of service under different conditions and compare the performance of our methods with the original VGA method. We identify that our most promising anticipatory technique can reduce the number of rejections in the busiest of demand scenarios by up to 1% of total demand. The increase in orders served comes at a cost of marginally increased distance travelled by the fleet of vehicles. Additionally, we note that the value of rejections saved increases by up to 5% when the system is not working up to its maximum capacity and allowing for greater scope for anticipation. Furthermore, our results underpin the strength and weaknesses of each anticipatory technique and highlight the importance of studying anticipation under a wide range of demand scenarios.

Acknowledgements

The journey to the completion of my thesis has been a testing period- starting off with recovering from major knee surgery to working remotely continents apart. All this would not have been possible without the patience, empathy, and generosity of my Ph.D. supervisor Maximilian Kronmueller. I will particularly miss our coffee chats that almost always extended into long discussions, not only about my thesis but also about the many ideas I came up with within the vehicle routing domain.

Further, this thesis would have been incomplete without inputs from Andres Fielbaum Schnitzler. Not only is my work inspired by his study, but also unapologetic-ally builds from his ideas and viewpoints of anticipation within the VGA framework.

I would also like to thank my supervisor Javier Alonso-Mora. Our meetings left a profound impact on how I approached my thesis and hopefully resulted in producing work that extends to further discovery in anticipation and vehicle routing.

I'd also like to thank all the Ph.D.'s and fellow MSc students at AIR lab, Delft. Spending lunches discussing our work and playing board games kept me motivated to do better every day.

Last but not the least, I'd like to thank my family without whose effort I'd never have gotten this opportunity to enrich myself with such engaging experiences.

Stavya Bhatia
Delft, March 2022

Contents

Acronyms

Abbreviation	Definition
VRP	Vehicle routing problem
DVRP	Dynamic vehicle routing problem
SDVRP	Stochastic dynamic vehicle routing problem
CAGR	Cumulative aggregate growth rate
IT	Information yechnology
SDD	Same day delivery
SDPD	Same day pickup and delivery
VGA	Vehicle group analysis
AFMP	Ambulance fleet ,anagement
DARP	Dial a ride problem
CI	Cheapest insertion
MSA	Multiple scenario approach
MPA	Multiple plan approach
ALNS	Adaptive large neighborhood search
CBH	Cost benefit heuristic
ADP	Approximate dynamic programming
UO	Unknown orders
IO	Ignored orders
LO	Loaded orders
PO	Picked orders
DO	Delivered orders
OSM	Open street map
CV	Candidate vehicle
CTV	Candidate trip vehicle
ILP	Integer linear program
FN	First node
LN	Last node
CD	Closest depot
KPI	Key performance index
CVRP	Capacitated vehicle routing problem
TW	Time windoow
DD	Delivery deadline
VFA	Value function approximation
PFA	Policy function approximation
MVF	Modeling value function
AVI	Approximate value iteration
PA	Perfect anticipation
P+R	Penalty + Reward
WF	Wait-First
DEH	Double edge horizon

List of Figures

List of Tables

1 Introduction

Today, the advent of high-speed internet, instant downloads, live-streaming, etc. have all conditioned the modern customer to expect instant gratification. This, in turn, has impacted customer expectations when it comes to how we perceive last-mile delivery: the domain of delivering physical goods, transportation of people, and various kinds of at-home services. Since the Covid-19 pandemic, the demand for goods and services at one's doorstep is continuously increasing [51]. At the same time, this increase in demand is combined with the hopes of achieving the same level of gratification as one would at a brick-and-mortar store, i.e, customers expect near-instant deliveries[17]. The fact that a quarter of the customers aged 18-34 abandon their online orders if their products are not shipped within the same day stresses this need for more efficient delivery operations [44].

As such, the growing nature of demand has made logistics imperative to the manufacturing, e-commerce, food, and public transportation industries. This can also be observed by the $10.6 trillion expenditure on logistics in 2020 alone, of which 28% was solely for last-mile distribution infrastructure [47]. Furthermore, the market size of last-mile deliveries is expected to rise at a CAGR of 8.9%. This indicates an increase from $39.57 Billion in 2020 to $66 Billion in 2026 and is largely attributed to the expansion of e-commerce [46].

Another domain influenced by last-mile deliveries is the grocery industry. Companies like Gorillas, Flink, Blink-it, Zepto, Amazon, DoorDash, and many more [44][30][66][27] realise this potential and have invested heavily in infrastructure, technology, and labour force to offer faster and in some cases under-10 minute deliveries. The main focus of these companies remains to attract a greater share of the global online grocery consumer market. A market that is poised to grow by $600 Billion between 2020-2024 [34].

As a result, the importance of an efficient distribution strategy that not only reduces the growing operational cost but also increases the level of service to their customers is no longer just an active area of interest but an immediate necessity to meet the growing demand.

From a technological standpoint, various advancements within IT and telecommunication, with growing amounts of data, offer opportunities to meet the rising expectation of instant gratification by customers. In particular, increased computing power allows these companies to use routing algorithms that can generate re-planned solutions in real-time thereby incorporating recent information [19]. These changes can then be communicated effectively to individual drivers enabling same-day delivery (SDD). However, in order to truly achieve sustainable and profitable grocery delivery within minutes of a request, reactive planning that simply responds to current information may not be enough to achieve effective routing for the rapidly moving fleet of vehicles. Anticipation, i.e, incorporating information about the future is essential to avoid ineffective decisions [18].

To illustrate the benefit of anticipation, Figure 1.1 highlights the route taken by a vehicle under a myopic algorithm and compares it with its anticipatory counterpart at different time intervals. In this example, all orders must be delivered within a total operation period of 80 seconds. Further, the time required to

travel between two locations is given along the edges. Figure 1.1a represents the original route plan (black arrows) by both the anticipatory and the myopic routing techniques at $T = 0seconds$. Orders 1 and 2 are known at $T = 0seconds$ and result in a planned route that picks up both orders from depot D1 and delivers them to their respective destinations. The estimated time at the end of delivering order 2 is 60 seconds. Further, Figures 1.1b and 1.1c represent the route plans by the myopic and anticipatory routing techniques at $T = 25seconds$ respectively. Order 3 is not known at $T = 25seconds$ as its time of request is $t = 30seconds$. With the unchanged demand, the myopic route plans continue with the initial plan of delivering both orders by 60 seconds. On the other hand, the anticipatory technique can explicitly predict the possible occurrence of orders in the future. This is indicated by the red basket denoting order 3 in Figure 1.1c. As a result, the anticipatory technique redirects the vehicle as per a new route plan (green arrows). According to this new plan the anticipatory technique takes a minor detour towards depot D2 and is en route to deliver order 2 by the end of 65 seconds. Figure 1.1d and Figure 1.1e further highlight the evolution of the route plan for the myopic and anticipatory algorithms at $T = 35seconds$. At this time order 3 is known. The myopic algorithm delivers order 1 and is en route to deliver order 2. However, in order to deliver order 3, it has to return to depot D2 to collect the order. It is unable to do this as the time required to reach depot D2 exceeds the operational period of 80 seconds. On the other hand, for the anticipatory routing technique, the vehicle is already located at the depot D2 and can deliver both order 3 and order 2 within the operational period of 80 seconds. As a result, the anticipatory technique is able to deliver additional orders within the same time period.

As illustrated, incorporation of anticipation within the routing algorithms can lead to better customer service and reduce the burden of rising operational costs. However, anticipation itself can be a major challenge in the context of routing[56]- sophisticated techniques suffer from a limited computation time and simpler techniques are usually unable to adequately handle large real-world scenarios. Therefore, there is a need to find a computationally inexpensive anticipatory framework that can improve performance for large real-world scenarios.

1.1. Research Objective

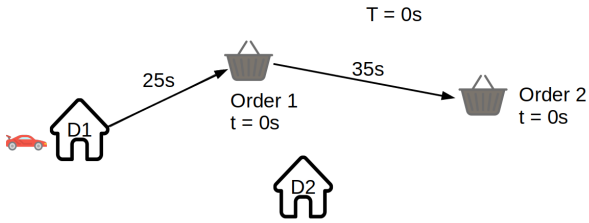
The objective of this work is to develop anticipatory techniques to enhance a state-of-the-art dynamic routing algorithm for a large-scale on-demand grocery delivery service. In addition, we develop solutions for situations that can deliver orders very fast in a densely populated setting with multiple warehouses. The primary performance indicators include the total distance traveled by the fleet of vehicles, the overall delay in delivery of orders, and the number of ignored orders. The framework operates under a set of service constraints that have been adopted from previous works. The goals identified include:

- Propose an anticipatory framework that has the potential to improve the state-of-the-art VGA algorithm.
- Evaluate whether the proposed anticipatory technique outperforms the base approach on a wide variety of problem instances.
- Analyse the circumstances in which the proposed anticipatory technique has a greater impact on performance improvement.
- Establish a benchmark for maximum potential performance improvement when deploying anticipatory approaches over the base framework.

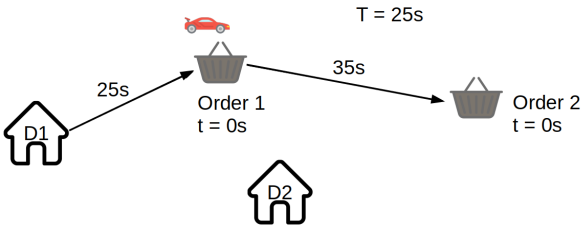
1.2. Contribution Statement

The contributions in this work expand on earlier published works for on-demand SDD operations and anticipatory routing [28][15][2]. We do this by re-implementing the state-of-the-art with suitable feature enhancements. More specifically, we contribute by:

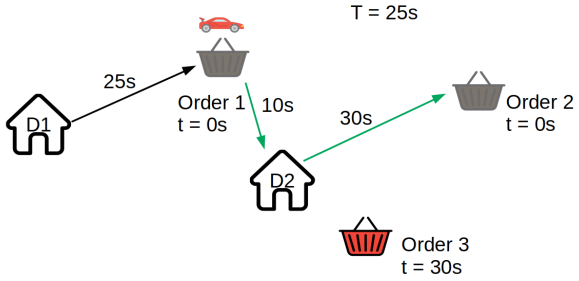
- Proposing and investigating several anticipatory techniques in combination with the state of the art Vehicle Group Assignment (VGA) framework designed for the context of on-demand same-day grocery delivery.
- Improving the performance of the VGA framework through anticipation.



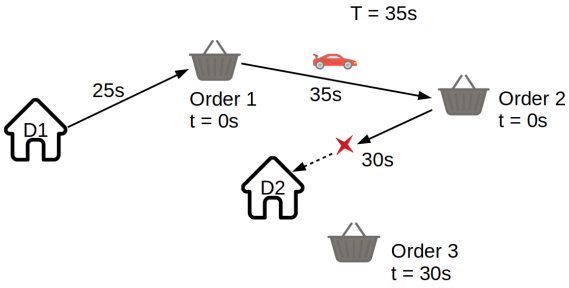
(a) Initial routing scenario at $T = 0seconds$.



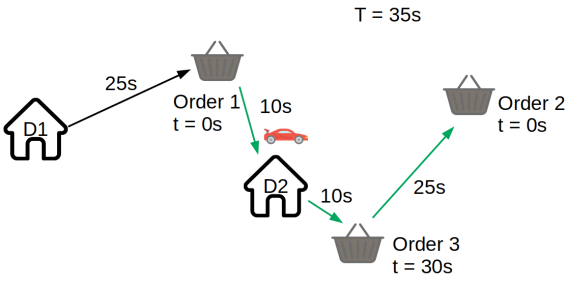
(b) Myopic routing at $T = 25seconds$.



(c) Anticipatory routing at $T = 25seconds$.



(d) Myopic routing at $T = 35seconds$.



(e) Anticipatory routing at $T = 35seconds$. The basket of order 3 changes color from red to grey as it is known and no longer anticipated

Figure 1.1: An illustrative example of routes taken by a vehicle under myopic and anticipatory systems at different time intervals. All orders must be delivered within a total operation period of 80 seconds. Further, the time required to travel between two locations is given along the edges. Figure 1.1a represents the original route plan (black arrows) by both the anticipatory and the myopic routing techniques at $T = 0seconds$. Orders 1 and 2 are known at $T = 0seconds$ and result in a planned route which picks up both orders from depot D1 and delivers them to their respective destinations. The estimated time at the end of delivering orders 2 is 60 seconds. Further, Figures 1.1b and 1.1c represent the route plans by the myopic and anticipatory routing techniques at $T = 25seconds$ respectively. Order 3 is not known at $T = 25seconds$ as its time of request is $t = 30seconds$. With the unchanged demand, the myopic route plans continue with the initial plan of delivering order 2 by 60 seconds. On the other hand, the anticipatory technique can explicitly predict the possible occurrence of an order in the future. This is indicated by the red basket denoting order 3 in Figure 1.1c. As a result, the anticipatory technique redirects the vehicle as per a new route plan (green arrows). According to this new plan, the anticipatory technique takes a minor detour towards depot D2 and is en route to deliver order 2 by the end of 65 seconds. Figure 1.1d and Figure 1.1e further highlight the evolution of the route plan for the myopic and anticipatory algorithms at $T = 35seconds$. At this time order 3 is known. The myopic algorithm delivers order 1 and is en route to deliver order 2. However, in order to deliver order 3, it has to return to depot D2 to collect the order. It is unable to do this as the time required to reach depot D2 exceeds the operational period of 80 seconds. On the other hand, for the anticipatory routing technique, the vehicle is already located at the depot D2 and can deliver both order 3 and order 2 within the operational period of 80 seconds. As a result, the anticipatory technique is able to deliver additional orders within the same time period.

- Developing an independent framework for data set generation to generate multiple reproducible data sets for analyzing the proposed anticipatory techniques.
- Identifying the impact of different spatial & temporal demand characteristics on the anticipatory potential of the proposed techniques.

1.3. Thesis Overview

This work is composed as follows. First, in Chapter 2, an overview of the existing literature on vehicle routing is presented. Chapter 3 defines the problem context and the state-of-the-art VGA algorithm to provide the reader with both an intuitive as well as a mathematical understanding of the topics. Chapter 4 further focuses on the most promising anticipatory techniques and discusses several modifications that were proposed to influence VGA's decision-making. Further Chapter 5 focuses on several demand instances that have been prepared for the purpose of comparing results and investigating the effect of the data. In Chapter 6, the proposed methods are evaluated and a comparative study is highlighted. Finally, in Chapter 7, conclusions and recommendations for future work are presented.

2

Related Works

The related literature is split into several sections. The first section introduces the Vehicle Routing Problems (VRP). It further illustrates different types of VRPs and the attributes that define a VRP. The next section covers initial work regarding SDD problems. This is further expanded into more recent works involving anticipation in the next section. The sections following anticipation cover literature pertinent to our problem formulation and research objective. These include multiple-depot vehicle routing, work regarding different demand scenario generation, & most importantly methodically related work.

2.1. Vehicle Routing Problems

On-demand same-day grocery delivery problems belong to a well-studied family of problems named Vehicle Routing Problems. VRP's were first introduced by Dantzig and Ramser *et al.*[12] in 1959. Their work aimed at finding an optimal truck-dispatching solution for efficient problem delivery of gas to gas stations. However, VRPs can further be generalized as a set of problems whose goal is to find optimal route plans for a fleet of vehicles so as to visit several sets of customers from one or multiple depots. A solution to a VRP is illustrated in Figure 2.1

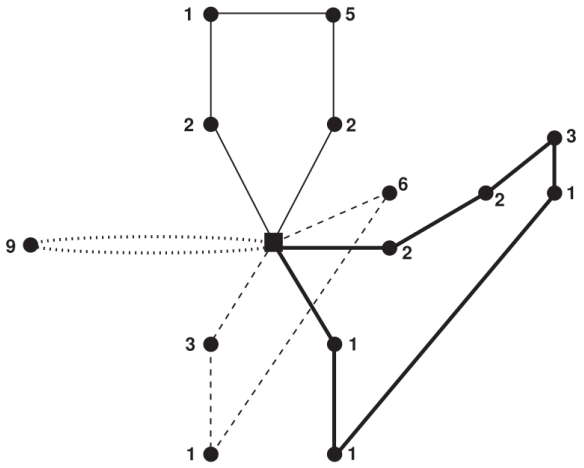


Figure 2.1: Illustration of a VRP: The problem includes 14 customer locations and their respective demand quantities. These customers are to be served by 4 vehicles with capacities of 10 each. Different line styles represent individual vehicle route plans[29].

VRPs themselves is a vast field of research and can be classified into four main sub-categories of problems based on information availability and information certainty. This is covered in detail by Schorpp

et al.[54].

Classification on Information availability: Problems are either categorized as static or dynamic.

In the static case, all necessary information needed to solve the problem is available beforehand. Several approaches such as ant colony optimization, integer linear programming, tabu search heuristic, branch and bound algorithm, large neighborhood search, simulated annealing, etc. have been explored extensively for solving the static variant of the problem and can be studied further in the work by Toth *et al.*[58].

On the other hand, in the dynamic variant information is only made available as time progresses. Limited information at any point in time impacts the performance. This is because even if the routes planned are optimal given the current information, they may result in sub-optimal routes when future information is made available. The DVRP variant in itself comprises a broad class of problems including applications in several logistical domains such as Dial-a-ride problem (DARP)[10][11][8], Courier services[40], Grocery delivery, Ambulance fleet management problem (AFMP)[69], etc. A comprehensive overview of such problems is given by Pillac *et al.*[39] and Psaraftis *et al.*[42]. It is important to note that each of the logistical contexts mentioned has distinctive attributes that define the problem being solved. For instance:

- A retail-based delivery can be defined as a fleet of vehicles that receive requests that are to be picked from any warehouse and delivered to their respective destination. These problems typically optimize for the number of orders served, but the distance traveled and customer lateness are also considered in some studies.
- Dial A Ride Problem refers to a problem wherein individuals need to be transported from any pickup point to any destination within a stipulated time period. Instead of distributing agents from a single depot to many different locations, it involves no depot and agents that are both picked up and dropped off. The urgency of serving customers is the main goal and hence, these problems typically focus on optimizing the arrival times.

.

Depending on the logistical context, attributes that formally define a DVRP are varied. These attributes in turn impact decision-making when developing a solution framework. For the sake of complete comprehension, it is imperative to formalize the different attributes that make up a DVRP. Adapting from the taxonomy by Psaraftis *et al.*[42], we categorize the different attributes that are used to classify a DVRP in Figure 2.2. A detailed description of the relevant attributes is further available in Appendix A.

Classification on information certainty: This type of classification splits the problem into deterministic and stochastic. When all information required for the purpose of solving a VRP has no uncertainty associated with it, the problem is considered deterministic. On the contrary, a stochastic problem deals with uncertainties due to one or many factors. These include uncertainty in the demand quantity of a request, uncertainty in the occurrence of a request itself, or uncertainty in the travel times of the vehicles. The stochastic variant of vehicle routing problems and their solution are covered in great detail in the survey by Ritzinger *et al.*[49].

2.2. Same Day Delivery

SDD operations are a particular case of VRP's characterized by deliveries that must occur within the same day of the order being placed. In these types of problems, orders are placed while the fleet of vehicles is operational and already executing a planned route. This means that at the time of planning and execution-only partial information about the orders is available. In a special case of SDD, all requests need to be picked up from their appropriate pick-up locations before being delivered. These are called same-day pickup and delivery problems (SDPD). These problems constraint the sequence in which different locations need to be reached, thereby, adding to the complexity of the problem.

SDD problems can belong to two main VRP variants- dynamic vehicle routing problems (DVRP) or stochastic dynamic vehicle routing problems (SDVRP). An SDVRP problem is an extension of a DVRP

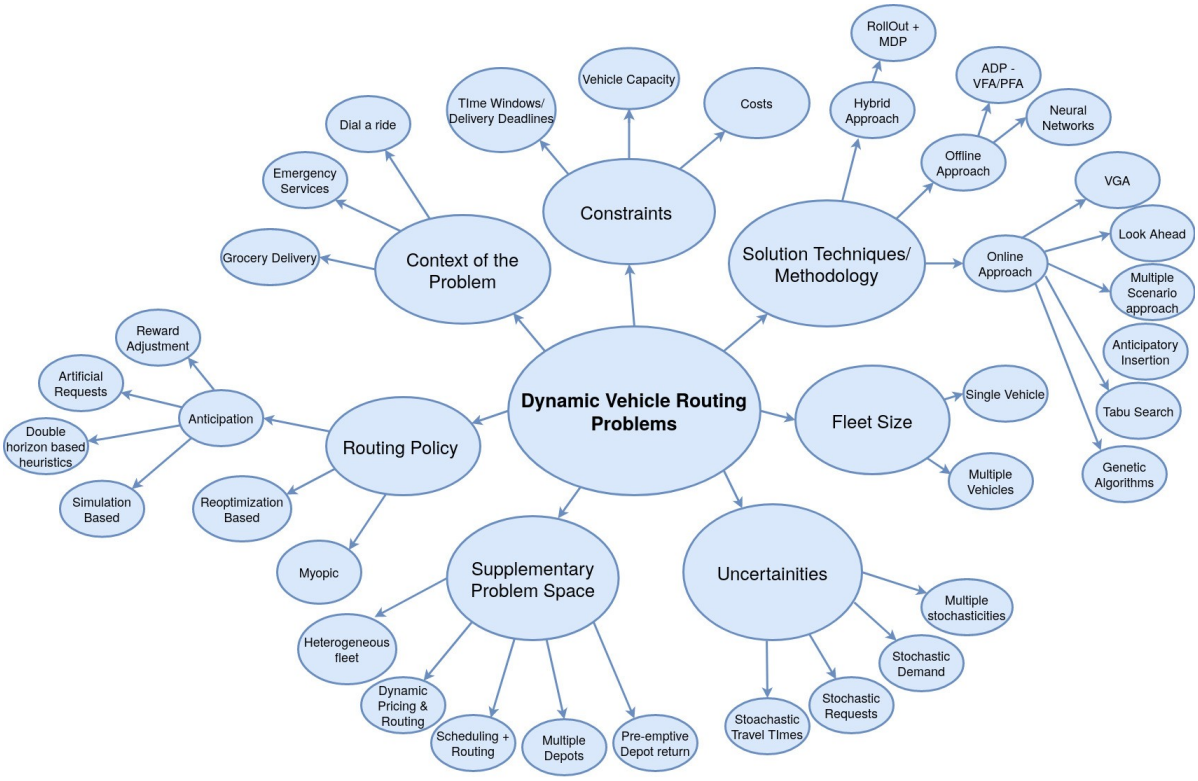


Figure 2.2: DVRP taxonomy: The figure highlights all attributes and solution frameworks that describe a DVRP. Each attribute can in turn take different values depending on the problem being solved and its corresponding solution approach.

problem where one or more types of uncertainties are also considered in the solution framework. These uncertainties are described in more detail in Appendix A.

One of the first contributions to SDD was the study by Gendreau *et al.*[18]. They developed a solution framework for the logistical context of courier services wherein a fleet of vehicles collect couriers from different locations and bring them to a depot. Their approach adapted a background tabu search heuristic [21][22], initially developed for the static routing problem, to a dynamic one with stochastic requests. At the occurrence of a new request, all tabu search heuristics are paused and the routes from the best solution of each parallel search are added to an adaptive memory[50]. The framework then decides whether to accept or reject the new request based on the feasibility of the new request in each solution. If the request is feasible for even one solution, it is accepted, and the corresponding feasible solutions are updated. The solutions which are incompatible are removed. Post updating, the tabu search is repeated till the next decision instance.

Additionally, Ichoua *et al.*[24] further built on the approach by Gendreau *et al.*[18] by allowing vehicles to divert to better routes while they were already en route to a particular destination. While the approach improved performance, it also posed technical challenges associated with identifying a high quality & feasible diversion in extremely short time spans.

Another notable contribution is that of the Multiple Plan Approach(MPA) by Bent & Hentenryck *et al.*[7]. The principal idea behind the MPA generalized the approach by Gendreau *et al.*[18] by making it independent of the search procedure used to generate solutions. During the execution phase, the MPA generates feasible routing solutions by means of a local search technique and stores them in memory. At the occurrence of a new request MPA checks whether the request can be incorporated within any of its generated solutions and if not, rejects the request. Although, if the request is feasible within any of the plans, it is accepted. Once accepted, all solutions in the memory that cannot incorporate the request are removed. Even amongst the remaining solutions, the one solution that incorporates the request best according to an objective is considered as the distinguished plan - the plan to be executed in reality which is chosen using a decision function. Once the request is assigned new routing plans are again generated which are in accordance with the current distinguished plan[6]. As plans are continuously generated, the distinguished plan is also adapted regularly.

In contrast to these works, our methodology works by generating an exhaustive list of possible trips that each vehicle can take. It does so using a search functionality. Next, our method then assigns plans based on the results of an ILP solver. This is in contrast to the consensus function or other heuristics needed due to multiple scenarios in the above studies. Furthermore, unlike the above approaches, our solution can handle multiple new requests at any decision instance.

SDD has gained major prominence in the last decade owing to advancements in computational capabilities [49]. However, these problems are NP-hard [16]. Because of this exact algorithms such as branch & bound, ant-colony optimization, large-neighbourhood search, & genetic algorithms are rarely able to solve large problem instances. In most cases, the incorporation of approximate techniques such as the ones covered in this Section, coupled with heuristic and probabilistic anticipatory approaches have helped achieve satisfactory performance. These are discussed further in the subsequent section.

2.3. Anticipation in Same Day Delivery

Anticipation in SDD can be classified into four broad categories- Non-Reactive Implicit, Non-Reactive Explicit, Reactive Implicit & Reactive Explicit. A summary of these classifications is illustrated in Table 2.1. Further, a detailed description of each category is provided in Appendix B.

2.3.1. Non-Reactive Implicit Anticipation

Waiting strategies are perhaps the most common and simplest of the anticipatory techniques applied in the SDD literature. They are primarily based on the abstract idea that by waiting at a specific location one can assimilate more information before taking any further planning decisions. This will in turn benefit the overall routing provided the waiting does not hamper with the deadlines of the current route. Several studies exist on waiting strategies and are briefly explained below.

Minic *et al.*[37] conducted an empirical study of 4 heuristic waiting strategies- wait first, drive first, dy-

Table 2.1: Classification of Anticipation: A summary of what each category of anticipation signifies.

	Reactive	Non-Reactive
Implicit	• No Stochasticity • Future decisions	• No future decision. • No stochasticity
Explicit	• Future decisions • Stochastic	• No future decision • Stochasticity

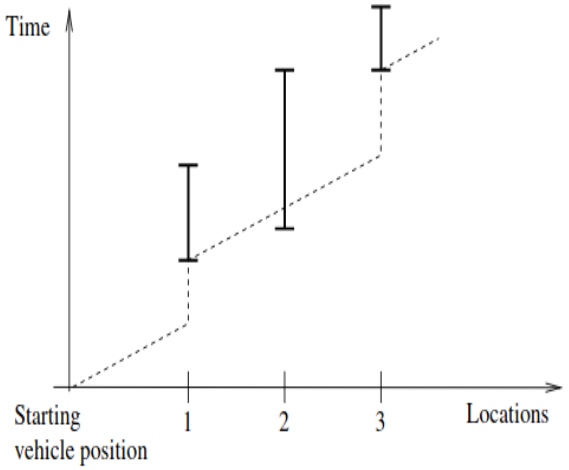
namic waiting, and advanced dynamic waiting for a dynamic pickup and delivery problem with time windows. These are illustrated by means of Figure 2.3. They tested the strategies for a total of 1,000 new requests. They further utilized the cheapest insertion[52] or tabu search heuristic and achieved overall improvements over the base algorithm.

Another study by Branke *et al.*[9] highlighted specific locations at which a vehicle is made to wait and how this impacted overall solution quality. The overall duration of waiting in each case is equal to the maximum duration a vehicle can wait without requiring a change in planning or the number of orders served. This is known as the slack time of the vehicle. Their strategies include waiting at the depot, waiting at a location farthest from the depot, waiting at every customer location with the same average waiting time, and waiting at each customer with a waiting time proportional to the distance traveled. In addition, they also explored evenly distributed waiting times for customer requests that are to be serviced when the time to reach the depot is less than the available slack time. They compared each of these approaches for 1,000 new requesting customers and also compared the solution with evolutionary algorithms. Their work specifically highlighted that waiting at the depot for as long as possible is not a suitable approach. For the other strategies, however, the performance is highly case-specific.

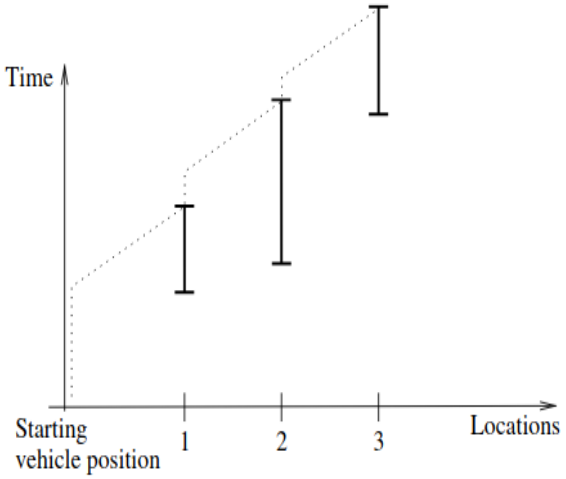
In a nutshell, waiting strategies work by simply delaying the decision-making to a future point in time. This delay allows the decision system to accumulate more information and make better routing decisions. In contrast, our anticipatory approaches do not delay decision-making. Rather, they exploit characteristics of either the underlying demand or other properties of the environment to make modifications to the intrinsic value (cost) of a trip. In most cases, our techniques modify the current routing plan. Furthermore, our approach works on a rolling horizon basis, i.e, decisions are made at a fixed cadence and for a group of orders. On the other hand, waiting-based approaches tend to re-plan at every new request. We also explore and discuss the implications of deploying waiting strategies on our solution framework in Appendix F.

Other studies incorporate heuristic anticipation by exploiting assumptions of what a good decision policy should look like. For instance, Kalina *et al.*[26] modified the Cheapest Insertion heuristic[52] specifically for the vehicle routing problems with time windows. This modification introduces elements to the cost structure based on the interactions between individual time window constraints caused by their respective widths and placements within the vehicle's route. They name this the Slackness Savings heuristic and empirically prove the addition of route flexibility in the insertion cost anticipates future customer requests implicitly. While our work incorporates a very similar modification to the cost function, we also introduce other factors such as a high penalty for every ignored order at a given time step. Furthermore, our anticipatory techniques also tend to utilize heuristic rules to modify the overall cost in addition to defining a more involved cost term.

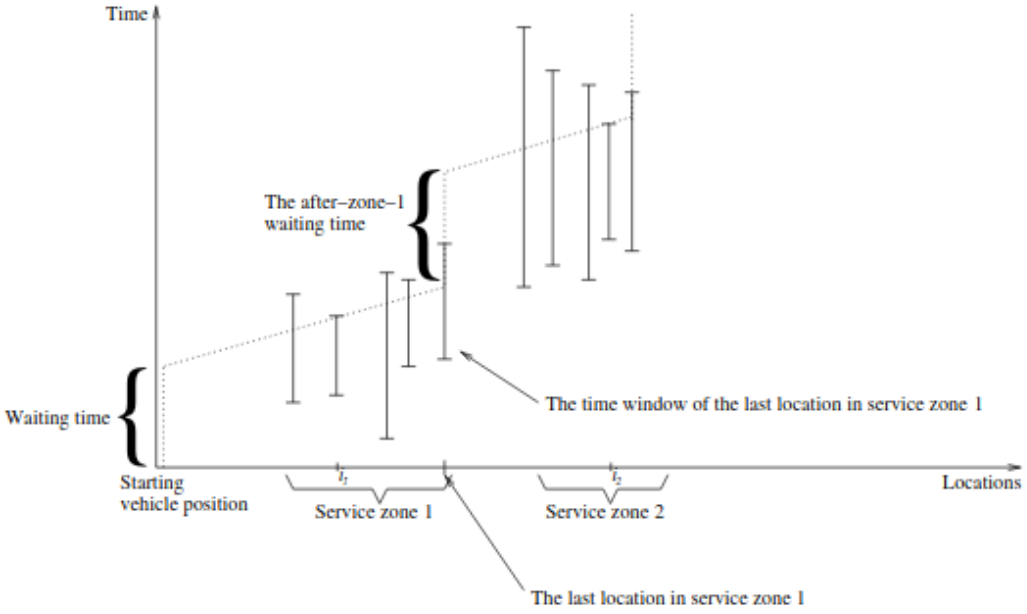
Minic *et al.*[36] further expanded on their waiting strategy research by developing a new anticipatory routing framework coined the Double Edge Horizon (DEH). The framework optimizes a two-fold goal- a short-term goal and a long-term goal. They set their short-term goal to reduce the total traveled distance and set the long-term goal to maintain routes that are flexible enough to include future requests.



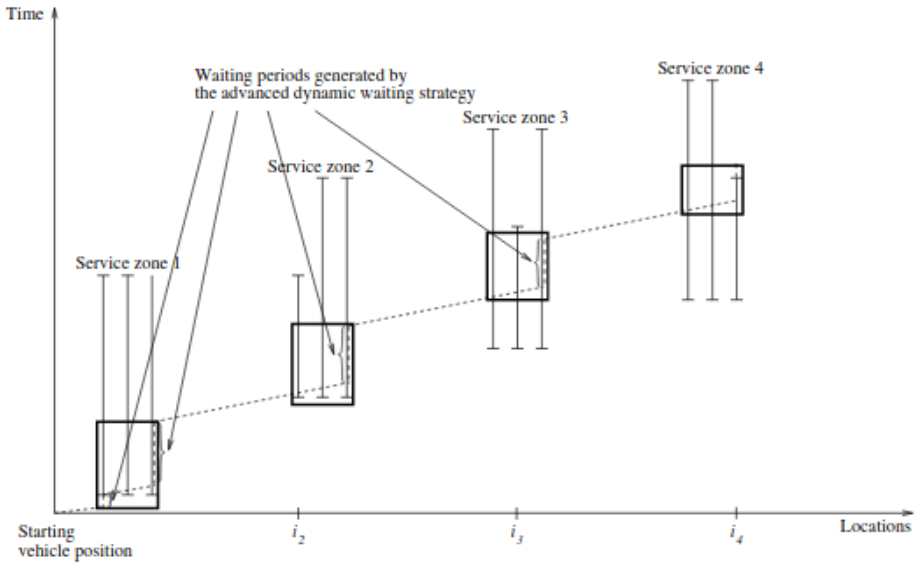
(a) Drive-First Waiting strategy



(b) Wait-First Waiting strategy



(c) Dynamic Waiting strategy



(d) Advanced Dynamic Waiting strategy

Figure 2.3: Heuristic Waiting Strategies [37]: 2.3a highlights the drive-first waiting strategy where-in the vehicle always moves to the next destination and waits if the vehicle arrival is before the earliest arrival time. 2.3b highlights the wait-first strategy where-in the vehicle waits at the past delivery location as long as it can deliver all known orders within the constraints of time. 2.3c expands on wait-first and drive-first by dividing orders into zones and implementing drive-first within a zone and wait-fist between zones. 2.3d enhances the dynamic waiting strategy by waiting only a fraction of the total available time depending on certain characteristics.

They achieved this by modifying the cost function in such a way that one component represents the increase in travel distance whereas the other component represents the reduction in slack time. Their work highlighted an improvement in routing performance but also noted a reduction in the degree of improvement as the requests increased. Our work differs from that of Minic's *et al.*[36] in two respects. At first, their problem serves orders based on Time Windows (TW), i.e., orders are served between two specific time periods in the future. In contrast, our problem works on the delivery deadline (DD) model, where orders need to be delivered as soon as they are requested and within a certain period. Second, the formulation of their problem allows them to plan on longer horizons of 4 hours. On the other hand, the planning horizon for our work is only 8 minutes. These differences, help us draw useful insights on the type of anticipatory approaches that will complement our solution framework. These are briefly covered in Appendix F.

2.3.2. Non-Reactive Explicit Anticipation

Azi *et al.*[3] studied the DVRP Problem with delivery deadlines of one hour. His work focused on accepting or rejecting a request as soon as it occurred and could only be assigned to vehicles that were waiting at the depot at the time of request occurrence. The success of his work can be attributed to making immediate decisions due to anticipation using scenario sampling.

Bent & Hentenryck *et al.*[7] further expanded their work on the MPA with the Multiple Scenario Approach (MSA). MSA extends on the MPA by including scenarios of possible future requests when generating new routing solutions. The future requests are sampled from either a probability distribution or historical data. Once the new solutions are obtained, the sampled requests are removed. Routes are selected by means of a consensus function wherein a route most similar to all other obtained routes is chosen.

Building on the work by Bent & Hentenryck *et al.*[7], Voccia *et al.*[66] compared the performance of the MSA algorithm with a fleet of 3 vehicles and for different request sizes ranging from 100-800 based on the data instances given by Solomon *et al.*[55]. They additionally leveraged waiting for strategies at the depot. They also analyzed the effect of the simulation horizon over the number of sample scenarios, indicating that the sample scenarios required increase quickly as the sampling horizon increases.

Another study on Non-Reactive Explicit anticipation is the anticipatory insertion technique developed by Ghiani *et al.*[19] for the single-vehicle routing problem. The principle idea of AI is to utilize the expected future requests to alter existing routes prior to solving. AI often incorporates a secondary heuristic decision over stochastic requests. For instance, Ghiani used an adaptation of the center of gravity LW strategy by Thomas *et al.*[57] on the combination of real and stochastic requests as the anticipatory technique in his study.

Anticipatory Insertion was further formalized in a recent study by Lochem *et al.*[32][31]. Within their method, historical requests were clustered in advance in order to exploit the inherent patterns in demand and then compute a predicted number of requests for each cluster. The predicted requests of each request type that were generated were then combined along with actual requests before the start of the operation. During the operation, any placed request was compared with the pool of artificial request types using a classifier. If the request was found comparable to any of the artificial request types, an artificial request from the cluster was removed and the real request was inserted in its place. This method was known as the add, replace, and remove technique. Further, an initial solution was constructed by routing using a cluster insertion heuristic where clustered requests were inserted into a single route. Finally, the initial solution was optimized using a solver involving ALNS and other heuristics. The interested reader is referred to their thesis work[32] for more information.

One fundamental difference between the anticipatory techniques explored in our study and the ones discussed in this section so far is that our approaches do not add to the computational burden of the algorithm. This is because, unlike the above-described approaches, our techniques do not identify multiple scenarios by explicitly sampling probable future requests. Rather, we focus on using implicit properties of existing demand for planning routes. This implicit nature allows us to iterate over large demand situations rather quickly.

Furthermore, Ichoua *et al.*[25] continued building on their previous work [24]. Their approach worked by dividing the map into independent sub-regions and then utilizing a policy to decide whether a vehicle should wait in a sub-region or serve the next customer. The decision to wait was made if the probability

of future requests in the corresponding sub-region was higher than a threshold value. This threshold value was computed using a PFA. One of the anticipatory techniques explored in our study also draws on the concept of sub-regions. It further determines whether a vehicle should remain in the sub-region or move to other demand regions depending on a threshold value. However, unlike Ichoua *et al.*[25], our work utilizes heuristics to determine a suitable threshold value.

Van Hemert & Poutre *et al.*[23] proposed an evolutionary algorithm that solved an SDVRP. Their study divided customer requests into so-called fruitful regions. These fruitful regions are basically clusters of customer locations that are likely to require service in the future. One of the anticipatory techniques explored in our study exploits the concept of fruitful regions for adjusting routing decisions towards promising regions.

Ulmer *et al.*[60] developed a Cost-Benefit Heuristic for customer assignments for same-day courier deliveries. Cost-Benefit Heuristic (CBH) analyses whether a customer request should be accepted depending on the location of the customer with respect to the vehicle. It does so by analyzing the relative gain of delivering a new request vs the relative cost associated with the delivery. They compared their methodology with other approaches and show comparable performance at much lesser computational demands. Our approaches use similar heuristic rules as CBH to make quick and flexible decisions however, there are some notable differences. CBH utilizes a CI heuristic to identify the route plan for each new order and then applies its CBH heuristic policy to decide whether the plan to serve the new order should be implemented. Our approach, on the other hand, first uses the anticipatory heuristic policy to modify the value of every pre-computed trip comprising of several orders. Then, the ultimate decision plan is implemented by an ILP solver. This allows us to evaluate hundreds or more route plans instantly. Further, CBH has shown to work on 1 vehicle and a handful of orders. On the contrary, our problem comprises of 10 vehicles and handles thousands of orders.

2.3.3. Reactive Explicit Anticipation

Most studies on this type of anticipation formalize the DVRP as a Markov Decision Process (MDP) and solve it using the Approximate Dynamic Programming method (ADP). ADP-based anticipatory techniques are sophisticated frameworks that simulate several future scenarios offline to identify the value of every possible decision the algorithm can encounter online. This allows them to make robust decisions keeping in mind the future impact. Unlike ADPs, our anticipatory techniques are limited to implicit anticipatory heuristics that only estimate the future impact. As a result, our approaches do not require an offline computation step to estimate the value of a decision. Furthermore, given the explicit nature of computation, ADPs are not known to scale to larger problem instances. This grossly limits their scope. The interested reader is referred to Appendix C for a brief description of ADP and its underlying definitions.

Meisel *et al.*[35] conducted one of the first studies to deploy approximate anticipatory techniques for a DVRP problem with late customer requests. They did this by implementing an offline AVI & MVF to evaluate the post-decision states (refer Appendix C for brief description). This allowed them to achieve reasonable spatial-temporal coverage for decision-making. ADP-based methods suffer from state-space dimensionality owing to the limits of computation and hence, their study is only performed on 49 customer locations.

Ulmer *et al.*[61] studied the SDD problem for a single-vehicle scenario. Their work comprised of a good inclusion of dynamic requests as they were able to overcome some of the limitations of dimensionality common with previous works including ADPs. They achieved this by aggregation and partitioning of the state-space [59]. The authors proposed a method building on approximate dynamic programming to evaluate which subset of requests to accept. The decision scheme comprised of an aggregated state-space of only the current time and the free time budget. The free time budget is considered as the time remaining after the vehicle serves all current customers. The computational analysis from the study indicated that this method provides superior results regardless of the degree of dynamism or the service area. However, as ATB only considers temporal parameters, it remains sensitive to how customers are distributed within the service area. In particular, ATB did not perform well in the case of asymmetrically clustered request locations.

Building on their previous work, Ulmer *et al.* enabled a solution that allowed the vehicle to return to

the depot even before existing requests are served [63]. As per our knowledge, this is so far the only study allowing for preemptive depot returns in SDD. By extensive computational tests, they showed that pre-emptive depot returns improved the obtained solutions in most cases. Our work also utilizes a similar construct of allowing returns to a depot even before a route plan is completed. The difference between our work and Ulmer *et al.* is that we do not incorporate anticipation to make decisions on early returns to a depot. However, we focus on much larger problem instances than the ones covered by Ulmer *et al.*.

Ulmer and Thomas[62] studied the SDD Problem with a heterogeneous fleet. Their fleet consisted of a set of drones and a set of road-based delivery vehicles. The focus of the problem was to find out whether a dynamic customer request should be served by a drone or a vehicle. Using a PFA they developed a heuristic decision-making policy that maximized the served requests. Their results showed that vehicles were better suited to serve high-density urban districts while drones work better for more distant and less populated areas. This was primarily because drones routes are independent of roads and therefore from traffic. Contrary to this study, our work focuses on a homogeneous fleet of vehicles.

2.4. Multiple Depot Vehicle Routing

As per our knowledge, limited work has been done in the context of multiple depots for DVRP's. Multiple depots add to the complexity of the problem as they add to the decision scheme by increasing the possibilities from where a request can be picked up. Most literature focuses on splitting the dynamic multiple depot problem into multiple single depot problems where-in each request is assigned to a single depot based on the proximity to the customer destination. This can be observed in the works by Yu *et al.*[33] and Xu *et al.*[68]. Our work differs from their approach as we include the decision of pickup location within the routing decision itself. A comprehensive overview of multiple depot vehicle routing has also been conducted by Montoya *et al.*[38].

Work on Demand Distributions

A solution framework can be considered robust only if it consistently performs well on a range of demand situations. As a result, generating multiple demand distributions for the sake of evaluating solution methodologies is pertinent to VRPs. In this respect, we identify two noteworthy works by Solomon *et al.*[55] and Van Lon *et al.*[65] where the generation of demand distribution scenarios was given paramount importance.

Solomon *et al.* generated 6 notable data sets, each that differed in their spatial-temporal characteristics as well as the scheduling horizon and length of time windows. They further assessed the performance of different heuristic methodologies on each instance and identified the heuristic methodology that consistently outperformed others for all demand patterns.

Van Lon *et al.* characterized demand distributions in terms of self-defined dynamic and urgency-based measures and identified how different temporal distributions- Gaussian, Poisson, and uniform vary in terms of these measures. They then generated distributions that varied only in terms of their urgency and/or dynamism keeping all other stochastic parameters fixed. They further tested the performance of cheapest insertion on the different distributions and concluded urgency and dynamism must be treated separately as characteristics of the data. Our work on generating data sets is largely inspired by these two studies.

2.5. Methodically Related Work

The anticipatory routing framework proposed in our study primarily builds on two works- the on-demand SDD by Kronmueller *et al.*[28] and the anticipatory routing methods for ride-pooling services by Fielbaum *et al.*[15].

The study of Kronmueller *et al.*[28] in turn built upon the work of Alonso-Mora *et al.*[2] and Alonso-Mora *et al.*[1]. The authors proposed a routing & assignment procedure focused on ride-sharing for a large-scale metropolitan area. Their ride-sharing method was called Vehicle-Group Assignment (VGA) Method and with enough computation time, their method could solve large-scale real-world instances of thousands of vehicles, in an any-time optimal manner. This was possible as the algorithm was

specifically able to decompose the problem of trip generation and vehicle assignment. Kronmueller *et al.*[28] translated their work into a retail-based SDD context. Their work is discussed in detail in Chapter 3.

Further, the study by Fielbaum *et al.*[15] extended the work by Alonso-Mora *et al.*[2] and Alonso-Mora *et al.*[1]. In particular, Fielbaum *et al.*[15] introduced anticipatory routing techniques to improve the performance of the VGA algorithm in the context of DARP. Their anticipatory techniques- artificial requests and adjustment using rewards - work by modifying the decision scheme within the VGA framework. While artificial requests affect trips by generating artificial future orders at specific nodes, rewards adjustment adjust trip costs by rewarding trips that direct vehicles to specific nodes. The number of artificial orders and rewards in each method was determined by means of several generation and rejection rates and the performance across each rate was explored. Additionally, they also compared performance when different specific nodes were considered. Overall, the results from the numerical analysis highlighted that incorporation of adjustment using rewards within the VGA framework result in a lower number of rejections with marginally higher delays. On the other hand, relatively lower delays with reduced service rates were observed for artificial requests when compared with the base algorithm. Our work devises modifications to the anticipatory technique - adjustment with reward - to suit the context of an on-demand grocery delivery problem. A detailed explanation of the adjustment with rewards approach is provided in Chapter 3. Our modifications are further explained in Chapter 4.

3

Preliminaries

At the time of conducting this thesis, the VGA framework proved capable of generating a healthy service rate for close to 10,000 orders with a fleet of just 30 vehicles. Thus, the algorithm holds great promise to formalize into practical solutions for real-life delivery applications. As a result, to further explore potential enhancement avenues, VGA was re-implemented for this thesis. In this chapter, the problem is formulated more formally in Section 3.1. Next, an overview of the implemented VGA algorithm is given in Section 3.2.

3.1. Problem Formulation

The problem we face is a SDPD problem where goods need to be delivered to customer locations as fast as possible from one of many depots via a fleet of vehicles. Further, the fleet of vehicles must work under a set of constraints and must aim to minimize a given objective function. For our work, the objective function is a combination of the total distance traveled by the fleet, the delay in delivery to each customer, and the total ignored orders at the end of the operation. The algorithm we build to solve this focuses on two main aspects- which of the vehicles serve which of the orders and the sequence in which the orders are assigned to a vehicle. In Section 3.1.1, we first define different elements of the problem. Further in Section 3.1.2, we describe the problem at hand. We cover the dynamics of the problem in Section 3.1.3.

3.1.1. Definitions

As the problem under consideration is exactly the one solved in [28], we adopt several of our notations and definitions from their work. Table 3.1 provides a summary of the notations considered. We further adapt their definitions wherever necessary to suit our work.

Environment: The problem environment is a weighted & directed graph $G = (N, A)$ that represents the road network used by vehicles. N represents a set of nodes of the graph and A represents the arcs that connect two nodes together. It is important to note that the graph is only sparsely connected, i.e., all nodes are not connected to each other. The weight of the arcs represents the travel times between two connected nodes. For the purpose of simplicity, we assume that travel times between a pair of connected nodes is always constant. Further, specific nodes of the graph are assigned as depots where the vehicles arrive to pick up the goods for placed orders. We assume that the inventory of goods is infinite in all of the depot nodes. The depot nodes are defined as $D = \{d_1, d_2, ..., d_h\} \subset N$.

Vehicle Fleet: The vehicle fleet $V = \{v_1, v_2, ..., v_m\}$ is defined by m identical vehicles that can pick up and deliver goods at specific nodes of the graph. A single-vehicle v_i can be completely defined at any time t by its current location $l_{i,t}$, the loaded orders $LO_{i,t}$ on the vehicle, and its planned route $\pi_{v_i}(t)$ representing the sequence in which it shall pick up and deliver orders. Further, the maximum capacity of orders that can simultaneously be on the vehicle is constrained by a value denoted by c.

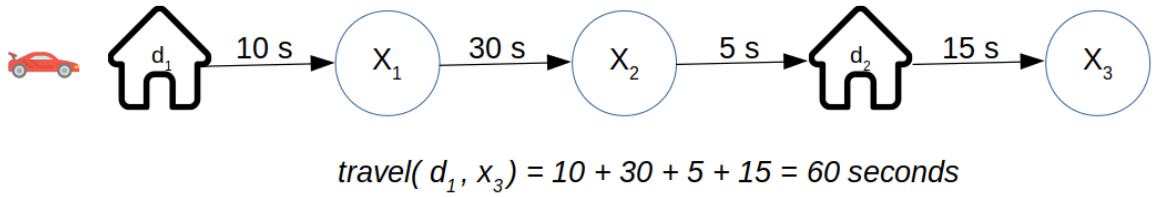


Figure 3.1: Travel time between depot node d_1 and destination node x_3: Travel time is the sum of individual weights of all the arcs along the path.

Demand: The set of orders $O = \{o_1, o_2, ..., o_n\}$ makes up the demand of unique orders during the operation period. In our work, we do not drill down into the elements of each independent order but assume that all orders are of the same size, set to a value of one. As a result, a unique order o_i is completely defined by the time at which it is placed t_i and the destination g_i at which it needs to be delivered. The pickup location of the order is not determined by the placed order itself. Rather, it is determined as a result of the routing algorithm. The demand is distinguishable into five sub-categories- UO_t, PO_t, LO_t, DO_t & IO_t. An order belongs to different categories depending on the point of time. UO_t represents the unknown orders that have not been placed yet i.e the time of simulation has not reached the time at which the request will be placed $(t < t_i)$. PO_t represents the set of orders that have been requested but not been loaded onto any of the vehicles. LO_t represents the set of orders that have been loaded onto a vehicle and set out for delivery. DO_t represents the set of completed deliveries. Finally, IO_t represents the set of orders that were either never assigned to a vehicle or could not complete delivery during the operation period. Orders are typically considered ignored when they can not satisfy the given delivery time constraints. Note that we consider a completely dynamic problem. This means that at time $t = 0$, all orders are unknown $UO_0 = O$ whereas at time $t = \tau$ all orders are either delivered or ignored $DO_\tau \cup IO_\tau = O$.

Times:

Each vehicle follows an independent route plan, i.e. vehicle's travel along arcs between any two nodes x_1 and x_2. The travel time between these nodes is given by the function $travel(x_1, x_2)$. Note that the graph is directed. This means that $travel(x_1, x_2)$ does not equal $travel(x_2, x_1)$ as the vehicle may be required to take a different path to reach x_1 from x_2. This is representative of one-way roads often found in urban environments. Furthermore, travel between any two locations is equal to the sum of the weights of all the arcs that need to be traversed between the two locations. This is further illustrated in Figure 3.1.

During operation, the vehicles not only traverse between nodes but also pick up and deliver goods. Loading goods at the depot as well as delivering orders at the customer location requires additional time that must be taken into account. For our work, we assume a constant loading and service time represented by δ_{load} and $\delta_{service}$ respectively.

An order o_i is picked up at time $t_{pick,i}$ and delivered to its destination at $t_{drop,i}$. The earliest possible drop-off (optimal drop-off) of an order is denoted by $t_{ideal,i}$. This optimal time to drop off is determined by considering a virtual idle vehicle that is already located at the nearest pickup depot denoted by $d_{best,i}$, and starts serving the order immediately after the request is placed. Mathematically, this means $t_{ideal,i} = t_i + \delta_{load} + travel(d_{best,i}, g_i) + \delta_{service}$. The actual delivery time can be higher than the optimal drop-off time as vehicles may already be serving other requests and may not be idle. Further, even if a vehicle is idle, it may not be located at $d_{best,i}$. The difference between the time it takes for the vehicle to actually drop off an order and the ideal drop-off time of that order is known as a delay. This is denoted mathematically as $\Theta_i = t_{drop,i} - t_{ideal,i} >= 0$. One of the constraints in the system is that of a maximum permissible delay for each order δ_{delay}. The latest possible drop off time hence can be computed as $t_{drop,max,i} = t_{ideal,i} + \delta_{delay}$. The maximum permissible delay is pre-determined by the operator and remains constant during the course of the operation. An illustration for all points in time for an order is

illustrated in Figure 3.2.

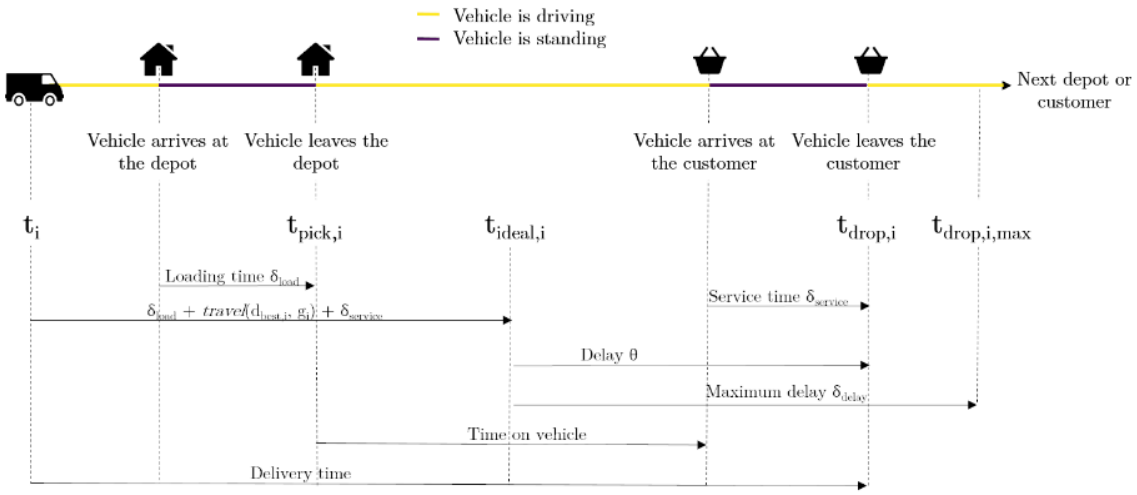


Figure 3.2: Visualization of the different time spans for one order. For an ideal delivery, a vehicle would be required already at the corresponding depot, the moment the order gets placed[28]

Table 3.1: Explanation of variables required for problem formulation

Variable	Explanation
Environment	
$G = (N, A)$	Directed graph representing the network
N	Node, representing a specific location
A	Arcs, connection between two nodes
D	Set of depots $D = \{d_1, d_2, ..., d_h\}$
d_i	Location of depot i
Vehicle Fleet	
v_i	Single Vehicle $v_i = \{t, l_{i,t}, LO_{i,t}, \pi_{v_i}(t)\}$
$l_{i,t}$	Current location of vehicle v_i at time t
$LO_{i,t}$	Loaded orders of vehicle v_i at time t
V	Fleet of m vehicles $V = \{v_1, v_2, ..., v_m\}$
c	Maximum capacity of a vehicle
Demand	
o_i	Single order $o_i = \{t_i, g_i\}$
t_i	Time order i gets known
g_i	Goal location of order i

Table 3.1: Explanation of variables required for problem formulation (contd.)

Variable	Explanation
Demand	
UO_t	Unknown orders- yet to be made known to the system at time t
PO_t	Placed orders- orders that are known but not loaded onto a vehicle at time t
LO_t	Loaded orders- Orders that are loaded by a vehicle at time t
DO_t	Delivered orders- Orders that are already delivered at time t
IO_t	Ignored orders- Orders that do not get delivered at time t
O	Set of all n orders $O = \{o_1, o_2, ..., o_n\} = UO \cup PO \cup LO \cup DO \cup IO$
Times	
t	Current time
τ	End of working day
δ_{load}	Loading time of one order at the depot
$\delta_{service}$	Service time of one order at its destination
$travel(x_1, x_2)$	Travel time between two locations x_1 and x_2
δ_{delay}	Maximum delay until an order has to be delivered
$t_{pick,i}$	Pickup time of order o_i
$t_{drop,i}$	Drop off time of order o_i
$t_{drop,i,max}$	Maximum drop off time of order i
$t_{ideal,i}$	Ideal delivery time of order o_i
Miscellaneous	
$d_{best,i}$	The closest depot to an orders destination g_i

3.1.2. Problem Statement

The problem we focus on is a Dynamic Multiple Depot Vehicle Routing Problem with Pickups and Deliveries. To elaborate, consider a direct weighted graph $G = (N, A)$, a set of depots D, where orders can be picked up, and a fleet of vehicles V in an initial state that is described by the location of the vehicles and the number of loaded orders on each vehicle $(l_{i,0}, LO_{i,0})$. The initial location of the vehicles corresponds to any of the depots and are empty $(LO_{i,0} = \emptyset) \forall v_j \in V$. Further, the operation during which the fleet is operational is indicated by times $[0, \tau]$. Orders are revealed at time $t = [0, \tau - \delta_\tau]$ where δ_τ is the time span during which no more orders are allowed to be placed. The objective is to find an optimal assignment Ω of orders $o_i \in O$ to vehicles $v_j \in V$ in such a way that:

- Each individual feasible vehicle route π_{v_j} from Ω must determine the orders that are to be picked up, the sequence in which they are to be picked up and the sequence in which they are to be delivered.
- It must find these feasible routes by minimizing an objective function subject to certain constraints. This function is described in Equation 3.1. The constraints include the maximum capacity c of the vehicle, the maximum delay δ_{delay} before which an independent order must be delivered and the end of the simulation period τ.
- Orders that cannot be delivered by any vehicle within the constraints of time $(\theta_i \leq \delta_{delay})$ or before the end of the operation period τ are ignored.
- Orders that can be delivered within the time constraints of δ_{delay} may not be assigned a vehicle. This happens if no suitable trips involving such orders are found or other better trips are available to the vehicle.

For our work, J is defined as the overall cost over the entire time horizon. It is expressed in time units as the cost of assigning a set of requests, known as a trip T_i to their respective vehicle v_j with updated routes π_{v_j}. Further, a penalty α is also added for ignored requests. This can be considered as charges incurred for hiring a third-party delivery agent to fulfill deliveries of ignored orders. The overall cost over the full-time horizon is indicated by Equation 3.1.

$$J = \sum_{(v_j, T_i, \pi_{v_j}) \in ga} \gamma(v_j, T_i, \pi_{v_j}) + \sum_{O_i \in IO} \alpha \tag{3.1}$$

where $\gamma(v_j, T_i, \pi_{v_j})$ is the cost of assigning vehicle v_j to trip T_i for the route π_{v_j}. Equation 3.1 represents the overall objective function at $t = \tau$. Generally, the cost function and constraints are universal and could be changed to fit other requirements. As a result, our problem combines several NP-hard problems, including the capacitated vehicle routing problem [45], the multi-depot vehicle routing problem [38], and dynamic optimization, while looking at large fleet sizes.

3.1.3. Problem Dynamics

The problem statement covered in the previous section does not explain how the problem evolves with time. At first, the problem evolves during the operation period $t \in [0, \tau]$ in fixed intervals denoted by $\psi = \{t_1, t_2, ...t_K\}$. At each interval, the state S_k at $t_k \in \psi$ of the problem is characterised by the time itself, the vehicle's fleet state V_{t_k} characterized by $t_k, (l_{j,t_k}, LO_{j,t_k}) \forall v_j \in V$, and the set of known but not yet loaded orders PO_{t_k}. This is highlighted in Equation 3.2:

$$S_k = (t_k, V_{t_k}, PO_{t_k}) \tag{3.2}$$

The transition from the current state S_k to a future state S_{k+1} involves the transition of the vehicle's fleet and the transition of the set of known but not loaded orders. The vehicle state transitions as per the trip each vehicle is assigned. This will be covered in Section 3.2. Further, the set of known but not loaded orders transitions due to the discovery of new requests as time progresses from t_k to t_{k+1} and loading of previous orders as per the trips $T_i \in T$ assigned to the vehicles. Figure 3.3 depicts an overview:

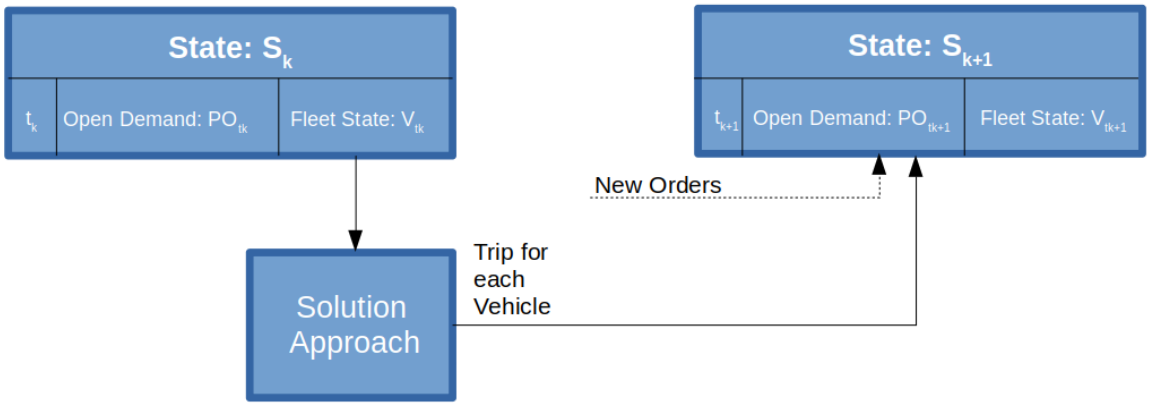


Figure 3.3: Dynamics of transition between two states: The transition of the vehicle fleet is known, however, due to new unknown requests, the transition of open orders is partly unknown. The solution approach takes the input state and outputs the assignment of a trip for each vehicle [28]

3.2. Method

The work in this thesis builds on the work of a previously developed solution algorithm. The algorithm is coined the Vehicle Group Analysis (VGA) method by the authors and is known for its capability to handle large problem instances [2][15][28]. In the following Section, we will elaborate on how the algorithm solves the problem discussed in Section 3.1.

3.2.1. Overview

The VGA algorithm works by solving the problem in a rolling-horizon style i.e, at a particular instance of time $t_k \in \psi$, an optimal solution is obtained. The simulation is then propagated forward till the next instance $t_{k+1} \in \psi$ wherein another solution is obtained by VGA. This process repeats till the end of the simulation period $t_K \in \psi$. Further, the decision intervals at which the VGA framework is implemented are constant and can be denoted by δt where $t_k = k \cdot \delta t$. This results in $K = \tau/\delta t$ decisions from the start till the end of the operation. Solving this K times results in the overall solution of the problem formulated previously.

The process of optimal decisions by the VGA can be divided into four steps. In the first step, a decision needs to be made about which depots can act as a picking point for an order. This is elaborated in Section 3.2.2. Once all possible order-pickup locations are obtained, potential trips need to be formed by grouping orders and the current vehicle location. The trip generation process is explained in a step-wise manner of 2 steps. Each step is discussed in detail in Section 3.2.3 and 3.2.4respectively. Furthermore, amongst the many trips possible for a vehicle only one must be selected and in such a way that one order can be delivered by one vehicle only. This is elaborated in Section 3.2.5. Further, we briefly discuss how does the methodology handles the case when vehicles are idle in Section 3.2.6. Next, the time propagation is briefly covered in Section 3.2.7. An overview of the base method is illustrated in Figure 3.4.

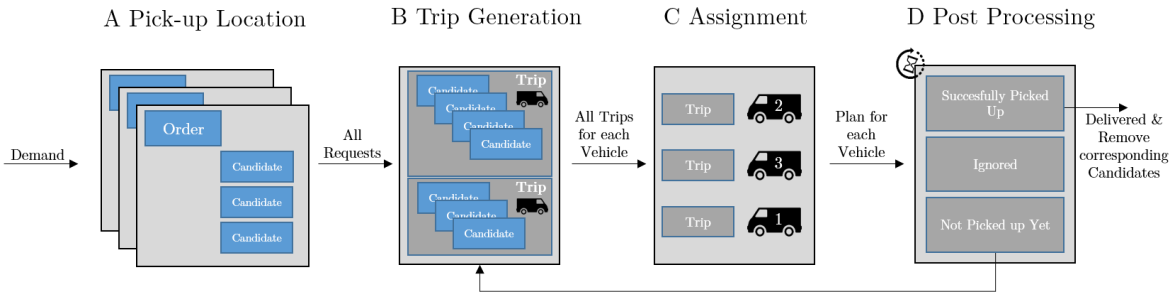


Figure 3.4: Schematic overview of the base method. "Step A assigns a number of potential pick-up locations to each order. During step B individual candidates are combined to feasible trips. Step C performs an assignment of trips and individual vehicles. With step D we ensure that the structure of the problem and the solution stays stable. The clock in step D symbolizes the propagation of time between states[28]."

As the base method does not explicitly consider future states of the problem, it is myopic in its approach. However, in our solution framework, we additionally incorporate an anticipatory technique prior to the assignment decision. The anticipatory techniques explored are briefly discussed in Chapter 4.

3.2.2. Pick-ups

When a new order occurs, the system receives a destination where the order is to be delivered and a time at which it was requested. This is in line with on-demand grocery delivery where the customer is not concerned with the pickup of the order itself. Thus, a decision regarding which depot should be used to pick up the order needs to be made. In theory, any depot d_l can act as a pickup location as long as the order can be delivered within the time constraint: $travel(d_l, g_i) < travel(d_{best}, g_i) + \delta_{delay}$. This implies that the travel time to a destination from a depot should be lesser than the sum of the travel time from the closest depot and the maximum permissible delay. All depots $d_l \in D$ that satisfy this constraint are considered as feasible options. To describe an order along with its feasible depot locations, a term candidate $c_i \in C$ of the order $o_i \in O$ is introduced.

A candidate is a tuple containing an order o_i and its associated pickup location, i.e., $c_i = (o_i, p_i) = (t_i, g_i, p_i)$. An order can have several candidates associated with it depending on the number of feasible pickup locations. In [28], a heuristic is implemented to limit the number of candidates for each order to a threshold value x. This implies the x closest pickup locations that satisfy the above constraint make up the candidates of the order. Limiting the number of candidates per order limits the number of potential trips for each vehicle, and hence, reduces computation time. Further, by computing feasible depots of all nodes N of the graph, candidates for every order can be determined offline. This is because candidates of an order only depend on the destination location and hence remain constant.

This additionally reduces the online computation time.

3.2.3. Candidate-Vehicle Graph

Each trip $T_i \in T$ can be defined as an ordered sequence of locations to pick up and deliver orders and is executed by a vehicle. The generation of trips is a two-step process. At first, the candidates are individually checked for the feasibility of delivery by a vehicle. The results are stored in the CV graph. In the subsequent step, trips are formed by identifying candidates that can be feasibly clubbed together on the same vehicle. This is illustrated in the next section.

The Candidate-Vehicle graph is an undirected graph comprising of two types of nodes- one for vehicles nodes and another for candidates. Two types of edges exist. These are vehicle-candidate and candidate-candidate. As such, an edge is added between a vehicle and a candidate when the candidate can be feasibly delivered by the vehicle. Further, an edge between two candidates implies two different orders that can be combined and delivered by the same vehicle. Such an edge is added between two candidates of different orders that share a common pickup p_i only when a virtual vehicle located at p_i can deliver both the orders feasibly. As a result, we check which vehicles can serve a given order and if two orders could be combined in a single trip, independent of the vehicle. This is further illustrated in step 1 of Figure 3.5.

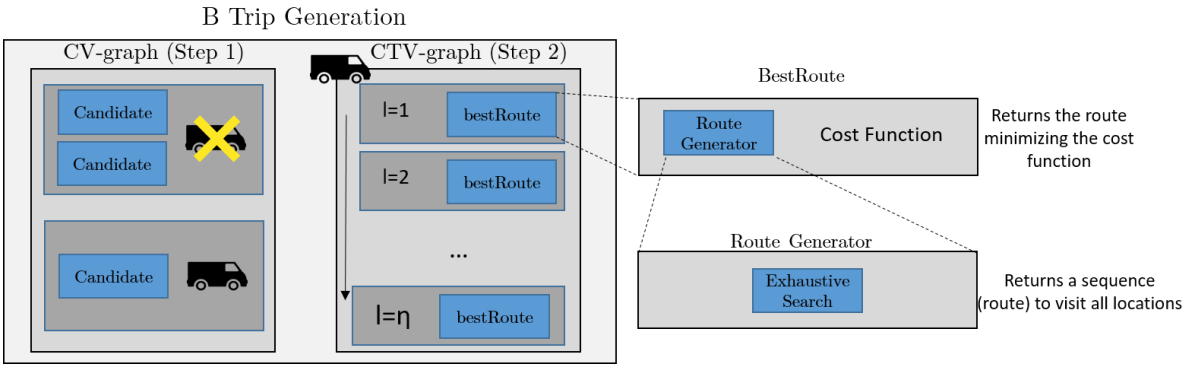


Figure 3.5: Schematic overview of trip generation[28].

3.2.4. Candidate-Trip-Vehicle Graph

As a next step to the CV graph, the Candidate-Trip-Vehicle graph is constructed one vehicle at a time. The CTV graph again has two types of nodes- trip nodes and vehicle nodes. A vertex is added for every trip that can be feasibly delivered by at least one of the vehicles. As such, all vehicles that can serve the trip are added as unique vertices $e(T_i, v_j)$. The vertex of this trip is in turn connected to all the candidates of that trip $e(c_i, T_j)$. The computation of a trip starts with 1 candidate that is already available in the CV graph and adds other candidates iteratively up to a maximum size η set by the operator. The CTV graph generation is computationally expensive and calculations for a vehicle stop when a predefined time $\rho_{max,CTV}$ has elapsed. All complete trips generated up to this point are considered for assignment [28].

Note that in the CTV graph, for a trip containing a pool of candidates and for a particular vehicle there can be many possible routes with which the candidates can be served. Each route must follow a different sequence in which the candidates of a trip are picked up and delivered. A simple illustration of this is provided in Appendix D. In our case, the formation of individual routes is achieved by the function $RouteGenerator$. $RouteGenerator$ performs an exhaustive search over the possible sequence of locations in which a vehicle can complete its deliveries. Additionally, for a vehicle that already contains prior parcels, the $RouteGenerator$ also includes the destinations of those parcels in the exhaustive search. The pre-empty capabilities in our solution, henceforth, increase the number of possible routes as new candidates can be picked up before prior parcel deliveries. Hence, for any trip-vehicle combination, there can be more than one way in which the trip can be executed. However, for every feasible vehicle and trip combination, only the optimal route and the associated cost are stored. These are computed by a function $bestRoute$ that iterates through the feasible routes of each trip-vehicle combination and

chooses a route that minimizes the given objective function for the trip. This cost of visiting a sequence of locations of a Trip T_i by vehicle v_j is given by Equation 3.3 and the overview of the CTV graph is illustrated in step 2 of Figure 3.5.

$$\gamma(T_i, v_j, \pi_{v_j}) = (1-\beta) \cdot \sum_{o_i \in T_i} \theta_i + \beta \cdot (TravelTime_i) \tag{3.3}$$

where $\gamma(T_i, v_j)$ represents the cost function, β represents the convex weight, and θ_i represents the delay in delivery.

3.2.5. Assignment of Trips to Vehicles

With the generation of the complete CTV graph, all feasible trips and the vehicles that can serve them are computed. However, a decision still needs to be made about which of the trips must be assigned to which of the vehicles to obtain an optimal routing solution. This decision is made by formulating the assignment as an integer linear program that is solved incrementally.

At first, a greedy solution is implemented that selects trips in a hierarchical fashion- Trips with the higher number of served candidates are prioritized over trips with a lower number of served candidates. If two or more trips serve the same number of candidates, then the trip that minimizes the objective function $\gamma(v_j, T_i, \pi_{v_j})$ gets a preference.

Next, the initial greedy solution Ω_{greedy} is used to initialize the ILP for obtaining the optimal assignment Ω of trips to vehicles. The notations for the ILP are defined in Table 3.2 [28]. Further, the mathematical formulation for the ILP itself is depicted in Algorithm 1 [28].

The number of binary variables in the ILP is equal to the number of candidates plus the number of edges between trips and vehicles. In the worst case, all vehicles can serve all candidates individually and all candidates can be combined in trips. However, this is unlikely. Equation 3.4 represents the objective function of a single state that must be minimized by the ILP. It includes the cost of the trips that are assigned to the vehicles and the cost of candidates that are rejected. Further, Equation 3.5 highlights the constraints that only one trip should be assigned to a vehicle. Next, Equation 3.6 represents the second constraint which indicates that either an order is assigned to at most one vehicle or is rejected. It furthermore ensures that only one candidate belonging to the same order is chosen. Finally, either the vehicle delivers the assigned orders or remains idle, i.e, it is not get assigned to any trip and remains idle.

Table 3.2: Explanation of variables required for Assignment Decision

Variable	Explanation
$\epsilon_{i,j} \in \{0,1\}$	Binary decision variables for every edge between a vehicle j and a trip i.
$\chi_k \in \{0,1\}$	Binary variable defined for every candidate $c_k \in C$. The value of 1 indicates a candidate that is not part of any assignment.
$e(T_i, v_j)$	Edge of CTV graph connecting Trip T_i and vehicle v_j.
ϵ_{TV}	Set of $\{i,j\}$ indices for all edges in $e(T_i, v_j)$.
$I^T_{V=j}$	Set of trips that can be serviced by vehicle j.
$I^V_{T=i}$	Set of vehicles that can service a trip i.
$I^T_{C=k}$	Set of Trips that contain a candidate k.
χ	Set of all variables $\chi \in \{\epsilon_{i,j}, \chi_k \| \forall i,j \in \epsilon_{TV}, k \in C\}$
$I^C_{O=h}$	Set of candidates that belong to order h.
$\gamma_{i,j}$	Cost associated with edge $e(T_i, v_j)$.
γ_{ko}	Cost of rejecting a request.

Algorithm 1 Optimal Assignment

Input: Greedy assignment of trips to vehicles Ω_{greedy}
Output: Optimal assignment of trips to vehicles Ω
begin
Initialize with Ω_{greedy};

Solve;

$$\sum_{optim} = \arg\min_{\chi} \sum_{i,j \in \epsilon_{TV}} \gamma_{i,j}\epsilon_{i,j} + \sum_{k \in 1,..,n} \gamma_{ko}\chi_k \tag{3.4}$$

with constraints-

$$\sum_{i \in I^T_{v_j}} \epsilon_{i,j} \leq 1 \qquad \forall v_j \in V \tag{3.5}$$

$$\sum_{I^C_{O=h}} \sum_{I^T_{C=k}} \sum_{I^V_{T=i}} \epsilon_{i,j} + \chi_k = 1 \qquad \forall o_h \in O \tag{3.6}$$

return Ω
end

3.2.6. Rebalancing

After the completion of the assignment step, there may be a scenario where one or more vehicles are not assigned any trips and do not have prior parcels to deliver. This could be because the vehicles may be located far away from the depots of the candidates, or other vehicles are serving the nearby candidates at a lower overall cost. As a result, such vehicles remain idle. To ensure that the idle vehicles are located at the best possible location for the next decision interval, rebalancing of idle vehicles is carried out. Rebalancing in itself is an anticipatory approach focused on only idle vehicles.

In our work, rebalancing of all idle vehicles is done by sending them to the closest possible depot location. This ensures that the vehicles are already located at pickup locations for the next decision interval ensuring reduced waiting times and delivery times of orders. It is important to note that more advanced rebalancing approaches have been studied previously. One such work is the rebalancing strategy for ride-sharing by Waller *et al.*[67]. However, our work focuses on extremely busy scenarios where the system is working close to its maximum capacity. As a result, the scope of rebalancing in our work is limited.

3.2.7. Time-Propagation

At the end of the assignment of any decision interval $t_k \in \psi$, an additional time propagation step is implemented. This step simulates the time and all other elements affected by it till the next decision interval $t_{k+1} \in \psi$. During this step the following events occur:

- The vehicle propagates according to the plan determined in the previous decision interval
- Parcels that are delivered are removed from the vehicles
- New orders are loaded onto the vehicle as parcels
- Orders that are ignored are recorded and removed from the trip generation process

Post the time-propagation, the above steps are repeated as per the problem dynamics covered in Section 3.1.3. In addition, minor changes were made to the base method introduced by Kronmueller *et al.* [28]. This is covered in Appendix E. Finally, additional notations used to completely define the solution methodology are covered in Table 3.3.

Table 3.3: Explanation of variables required for Solution Framework

Variable	Explanation
t_k	Time at decision k
$\psi = \{t_1, t_2, ..., t_K\}$	Time instants for re-optimization
$\gamma(v_j, T_i, \pi_{v_j})$	Cost Function for assigning trip T_i to vehicle v_j with updated route π_{v_j}
J	Overall Cost Function
α	Penalty for not delivering an order
β	Weighted factor for the cost function
p_i	Pickup location of the order i- independent of the order itself
c_i	One candidate of order o_i where $c_i = (o_i, p_j)$
$\pi_{v_j(t)}$	Planned route of vehicle v_j at time t
Π_{v_j}	Route followed by vehicle v_j throughout the operation period
Ω	Assignment of all routes to vehicles without anticipation
C	List of all candidates of orders O

4

Anticipation by Introducing Rewards & Penalties

Any solution methodology to a DVRP suffers from incomplete information when making routing decisions. As myopic solutions aim to identify optimal assignments for existing orders at any given decision step, they result in sub-optimal service performance whenever new information about the demand is made available. Research has shown that anticipatory routing methods improve service quality over myopic solutions [49]. In general, anticipation works by utilizing heuristic, analytical or probabilistic techniques such as machine learning, data mining, threshold-based decision making, etc, in two possible ways:

- predicting and incorporating future unknown orders in the trip computation allowing for better routing decisions taking the future into account. As such, these techniques deal with exogenous demand.
- using decision rules to increase route flexibility of a trip to be more robust for any future scenarios. Such techniques do not deal with exogenous demand for performance improvement.

The incorporation of an anticipatory technique within a solution framework is illustrated by means of Figure 4.1. As such, the vehicle state, emerging orders, overall trip generation (CV & CTV graph) and assignment process can be abstracted as the assignment procedure P. The anticipatory technique selected incorporates further information and works on top of the pre-defined assignment procedure. This abstraction clarifies the boundaries between the anticipatory technique and the assignment procedure, allowing us to explore several forms of anticipation for the same assignment procedure.

In this study, we modify and deploy anticipatory techniques that deal with non-exogenous demand and were first introduced by Fielbaum *et al.*[15]. These techniques alter how the VGA system assigns trips to vehicles i.e, influence assignment Ω to Ω_A. In particular, our techniques make modifications to the assignment procedure by introducing rewards and/or penalties to the objective cost function at every single stage of the problem, i.e, assignment $\omega_{A,k}$ at a single decision instance $t_k \in \psi$. To compare the performance across each approach and its myopic counterpart, we utilize the overall cost metric defined by Equation 3.1. One must note that the cost obtained in Equation 3.1 is a-posterior and can only be determined at the end of the operation. A simple addition of cost obtained at each interval of ψ will not compute this overall cost. This is because intermediate stages deal with the re-assignment of orders at future intervals. However, this cost provides a sound benchmark for determining the improvement of the anticipatory technique over the base (myopic) algorithm.

In addition to adjustment using rewards & penalties, we also explored alternative anticipatory techniques from existing literature. These include several waiting approaches[9][37], the double edge horizon technique by Minic *et al.*[36], and a trip trimming method inspired by the slackness-savings approach first proposed by Kalina *et al.*[26]. However, we identified through our preliminary analysis, that these approaches did not compliment the rolling horizon nature of VGA. As a result, we do not

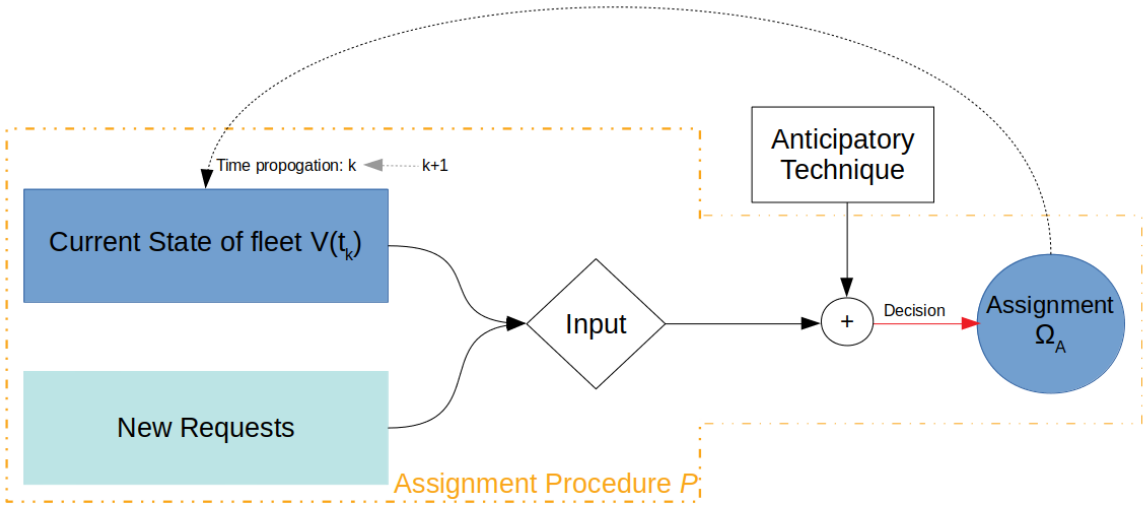


Figure 4.1: A diagram synthesizing the assignment process during the period of operation. The state of the fleet, and new orders behave as inputs to the decision system (assignment system). The anticipatory technique makes alterations to information from the input. This altered information is then fed to the numerical solver for assignment and routing decisions. Finally, the time propagates forward to the next decision interval $k+1$. Together the state of the system, newly emerging orders, and the assignment system form the assignment procedure P. The anticipatory technique works on top of P and the two are independent of each other. The blue background represents elements determined by the system and the green background represents elements that may depend on exogenous information.

discuss these in detail and the interested reader is referred to Appendix F & Appendix G for a detailed discussion and results respectively.

The rest of the Chapter is structured as follows. In Section 4.1, we introduce the basic concept of assignments with rewards. In Section 4.2, we introduce a penalty term to the assignments with rewards and the necessary modifications incorporated. Finally, we propose different variations to computing rewards & penalty adjustment values in Section 4.3.

4.1. Introduction to Assignment with Rewards

The anticipatory technique of assignments by introducing rewards was first proposed by Fielbaum *et al.*[15] for DARP. In our thesis, we modify the framework and translate the approach to the context of on-demand grocery delivery.

At each decision interval of ψ, rewards adjustment alters assignments by modifying the cost associated with feasible trips. We do this by altering the cost function γ in Equation 3.1. This modification is illustrated in Equation 4.1. Recall that $\gamma(v_j, T_i, \pi_{v_j})$ is the original cost of inserting trip T_i into vehicle v_j if the route is π_{v_j}. The rewards value R is introduced to modify this cost and achieve anticipatory routing. The design of the rewards value R is discussed in detail in Section 4.3. Further, a tuning parameter Θ controls how much weight is given to the reward.

$$\gamma_A(v_j, T_i, \pi_{v_j}) = \gamma(v_j, T_i, \pi_{v_j}) - \Theta \cdot R(v_j, T_i, \pi_{v_j}) \tag{4.1}$$

The anticipatory technique impacts the system in a dual manner. The first impact of this cost modification is the way in which the $bestRoute$ function [28] chooses a route for a given vehicle-trip combination. With adjustments to the costs from γ to γ_A, a different sequence of picking up and delivering orders may end up having the lowest cost. As a result, the route selection for each vehicle-trip combination might be altered. The second and more important impact is the change in assignments $\omega_{A,k}$ at any decision interval k by the ILP solver. The shift in assignments from ω_k to $\omega_{A,k}$ is also due to a change in costs from γ to γ_A that impacts each vehicle-trip combination differently. The complete adjustment of the assignment procedure is provided by a pseudo-code showcased in Algorithm 2 [15]. According

to the algorithm, the assignment procedure P gets influenced by the cost modification of the rewards adjustment technique. This is indicated in bullets two and three.

Algorithm 2 Reward Adjustment over assignment procedure P

Input: $G = (N, A)$, $C = \{c_1, c2, ...c_n\}$, $V = \{v_1, v_2, ..., v_m\}$, R, $CTVgraph$, $\psi = \{t_1, t_2, ...t_K\}$
Output: Ω_A
begin
- Compute feasible matches between trips $T_i \in T$ and vehicles v_j using the original assignment procedure P
- Re-compute the cost of routes π_{v_j} using the adjustment $\gamma_A(v_j, T_i, \pi_{v_j}) = \gamma(v_j, T_i, \pi_{v_j}) - \Theta \cdot R(v_j, T_i, \pi_{v_j})$ for every $e(T_i, v_j)$
- For every $e(T_i, v_j)$ determine the optimal route using γ_A.
- Implement the ILP solver to find the optimal assignments ω_{Ak} with γ_A as the objective function
- Update the vehicle itinerary to the next decision interval in ψ

return ω_{Ak}
end

One important variable to note is the weight of the rewards Θ. The value of Θ is tuned by the operator and determines the impact of the reward value on the system. If the value of Θ is too low, the impact on the system may be negligible. On the other hand, a very high value of Θ will result in a greater impact of rewards on the costs. This may degrade the performance due to a greater focus on preparing for the future than the present. Once the value of Θ is tuned, it remains constant and affects all trips at all decision intervals in the same manner. Therefore, altering the decision across trips is purely due to the reward value R which varies as a function of the vehicle-trip-route combination.

How the costs are modified depends on the manner in which the reward value R is designed. For instance, let us consider the value of a reward that takes a higher value when a greater number of rejected orders correspond to the nodes of the trip. Such a rewards value influences the VGA framework to favor trips that historically have a higher demand along the route. The question then arises on how to define the value of the rewards R? In our work, we propose several specifications of defining R. In a nutshell, all rewards value depend on some characteristic of one or more nodes of the route π_{v_j}. Two questions naturally arise:

- What node/nodes to look at?
- What characteristic of the node to use for computing the rewards value R?

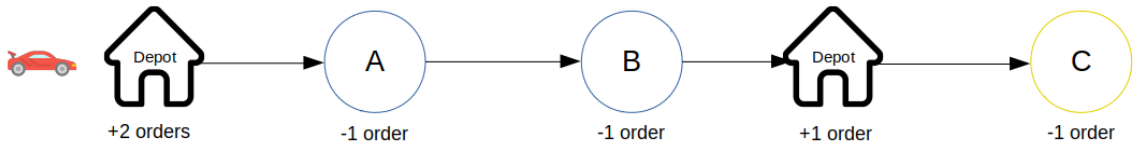
To answer the first question, our rewards value is subject to one node of the route π_{v_j}:

- The Last Node, $LN(\pi_{v_j})$, or
- The Last Depot, $LD(\pi_{v_j})$ of the route π_{v_j}

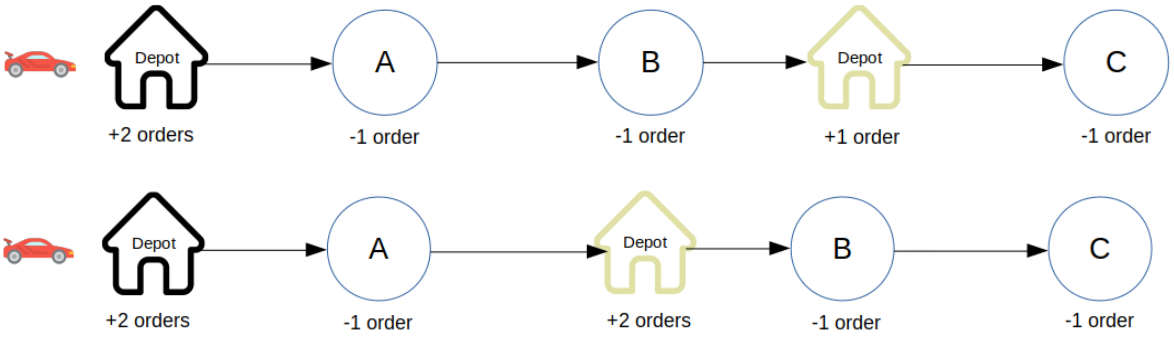
The Last Node, $LN(\pi_{v_j})$ of a route represents the final delivery destination at which the route culminates. On the other hand, the Last Depot, $LD(\pi_{v_j})$ of a route represents the depot at which the vehicle collects the last set of orders for the route without making a return to any other depot. An example of these nodes is provided in Figure 4.2. The Figure 4.2a illustrates the concept of the last node for a given route. Further, Figure 4.2b highlights the last depots of two independent routes.

Next, given the considered node is the last node $LN(\pi_{v_j})$, the reward value is typically a function of one or more of the following characteristics:

- Number of orders placed within the last 8 minutes that have their destinations corresponding to the node under consideration.
- Distance between the node and the closest depot CD of the node.
- Normalised average distance between x closest depots of the node and the node itself. The value x is determined by the operator and is equivalent to the number of depots considered for pickup for an order. Further, the normalization metric is the maximum distance between all nodes of the graph and all their x considered depots.
- Average of the candidates with pickup locations corresponding to all x closest depots of the node. The candidates considered are for orders placed in the last 8 minutes.



(a) An illustration of a vehicle route's nodes. The planned route of the vehicle collects parcels from the first depot and delivers them to two delivery locations A and B. Further, the vehicle collects another order from the second depot and delivers it at location C. Location C marks the last node of the vehicle route ($LN(\pi_{v_j}) = C$) for the decision interval as no other deliveries are scheduled at the given time step.



(b) An illustration of a vehicle route's nodes. The planned route of the vehicle collects parcels from the first depot and delivers them to its subsequent delivery nodes. The vehicle again returns to a different depot to pick up more orders and deliver them to its subsequent nodes. As no other pickups in the route are scheduled post the second depot (yellow), the second depot of the route in both cases represents the $LD(\pi_{v_j})$ last depot of the route.

Figure 4.2: Examples for Last Node and Last Depot of a Route

Furthermore, when the considered node is the last depot $LD(\pi_{v_j})$, we consider the number of candidates with pickups location as the depot itself to be the value of the rewards. These adjustment approaches are covered in more detail and described mathematically in Section 4.3.

Our proposed assignment with rewards shares some similar advantages and disadvantages as those covered by Fielbaum *et al.*[15]. First, the anticipatory technique does not increase the computational burden of the overall algorithm. This is because the system deals with the same number of orders as the myopic algorithm at the time of trip generation. As a result, if the myopic algorithm works effectively with regard to the computational burden, then it will continue to do so even after the incorporation of the anticipatory framework. Second, the impact of anticipation at an individual vehicle-trip-route level is clear. For instance, for rewards values based on orders at the last depot, the system will direct vehicles towards trips with a higher number of order pickups at the last depot of the route. However, unlike Fielbaum *et al.*[15], it is hard to assess the impact of anticipation on the entire system in a clear-cut manner. As a result, identifying the scenarios in which one should expect anticipation to perform well for our case is non-trivial. Similar to Fielbaum *et al.*, another disadvantage of our technique is that it operates at an individual level, i.e., the anticipatory technique impacts each vehicle-trip combination. As a result, it is hard to assess how many of the vehicles are directed towards a particular region. This may result in a greater imbalance between vehicles and demand regions because of an aggressive anticipatory technique. Further, the value of Θ needs to be determined carefully. This is because the larger the value of Θ, the greater the number of vehicles affected and hence, the greater the impact of anticipation.

4.2. Addition of a Penalty Term to Adjustment Techniques

In this section, we introduce an additional penalty term to the adjustment method discussed in Section 4.1. The basic principle of adding a penalty term was to prevent vehicles from going into unfavorable locations by penalizing their corresponding trips. Depending on certain underlying criteria, the penalty term introduced replaces the reward term for each vehicle-trip-route combination . These criteria are discussed independently for every variation to the technique in Section 4.3. A generalized formulation

that highlights an overall modification to the cost γ in Equation 3.1 is shown in Equation 4.2.

$$\gamma_A(v_j, T_i, \pi_{v_j}) = \gamma(v_j, T_i, \pi_{v_j}) - \begin{cases} \chi \cdot Pe(v_j, T_i, \pi_{v_j}) & \text{If underlying criteria are met (Section 4.3)} \\ \kappa \cdot R(v_j, T_i, \pi_{v_j}) & \text{If underlying criteria are not met (Section 4.3)} \end{cases} \tag{4.2}$$

where $Pe(v_j, T_i, \pi_{v_j})$ is the penalty associated with the trip-vehicle combination, χ, and κ are the weights associated with the penalty and rewards terms of the assignments including a penalty. Note that the tuning weight κ for the value of the rewards is different than Θ introduced in Section 4.1. This is deliberately chosen for two reasons. First, a separate tuning parameter allows a clear distinction between the adjustment with rewards techniques and the adjustment with penalty techniques that include an additional rewards term. Second, as the value of the rewards computed in adjustment with penalty is principally different than the one computed in adjustment with rewards, a different tuning factor range is applied.

In addition to the nodes and characteristics considered in Section 4.1, we also explore alternative computations of the penalty and rewards adjustment terms. Generally, these terms are still characteristics of one or more nodes of the system, however, the selection of nodes and characteristics considered vary. This is because adjustments based on penalty require additional nodes to estimate a penalty value. A simple classification for the nodes considered and their characteristics are provided below:

- First node $FN(\pi_{v_j})$ and last node $LN(\pi_{v_j})$ of the route: Penalty & rewards value are computed by first evaluating the number of candidate destinations within a certain threshold distance of the $FN(\pi_{v_j})$. The region within this threshold distance (or travel time) is called the vicinity of the node. If the value computed exceeds a certain pre-defined threshold, the penalty metric is computed on the basis of the $LN(\pi_{v_j})$. On the other hand, if the value computed falls short of the pre-defined threshold, the value of a reward is computed on the basis of the $LN(\pi_{v_j})$. This is explained clearly by means of an illustration in Figure 4.3.
- First node $FN(\pi_{v_j})$ and all other nodes $n \in \pi_{v_j} \vee n \notin FN(\pi_{v_j})$ of the route: Methods with this type of node selection follow a similar hierarchical approach to computing penalty and rewards values as the first approach. The only difference is that instead of computing penalty at the last node, it is computed for all nodes $n_o \in n$ that lie outside the vicinity region of the first node. This is also illustrated in Figure 4.4.

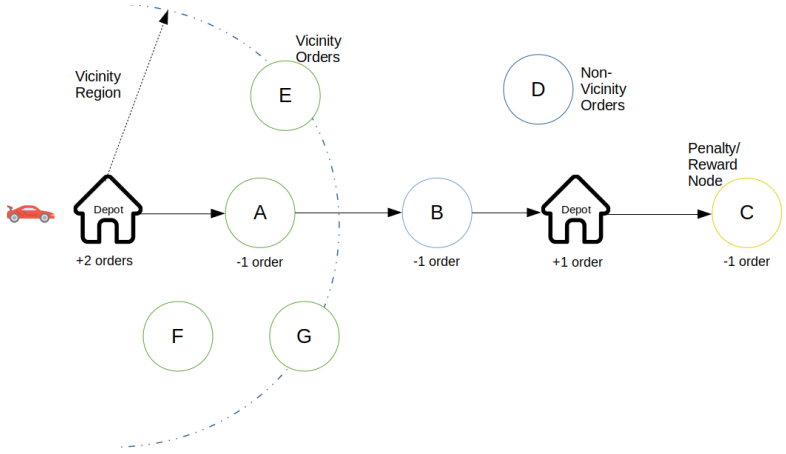
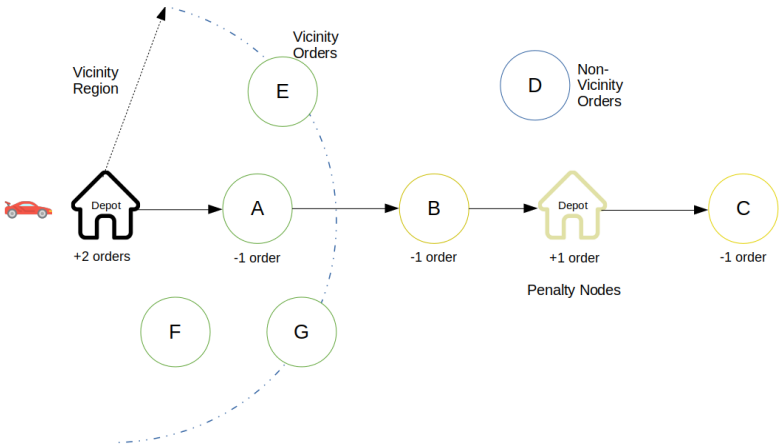
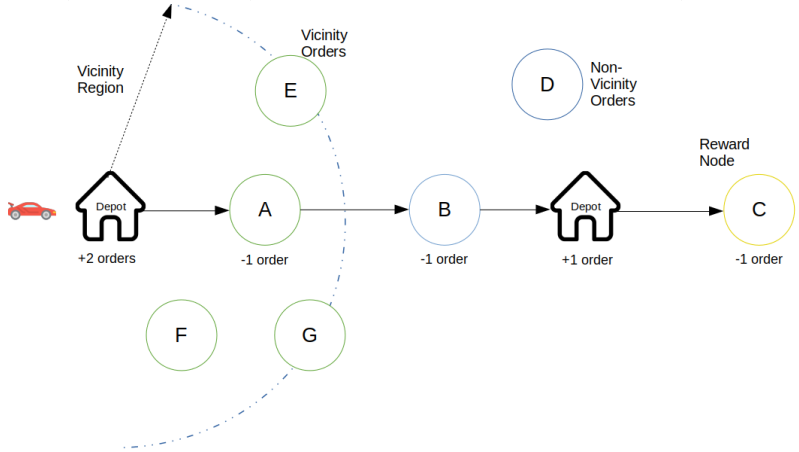


Figure 4.3: Example of Penalty + Reward Adjustment Techniques: Illustration of First Node & Last Node method of Penalty + Reward Computation. The green circles represent vicinity nodes, whereas the yellow circles represent the node at which penalty/reward is computed.



(a) Illustration of First Node & All Nodes method Penalty Computation. The green circles represent vicinity nodes, whereas the yellow circles represent the node at which the penalty is computed.



(b) Illustration of First Node & All Nodes method Reward Computation. The green circles represent vicinity nodes, whereas the yellow circles represent the node at which reward is computed.

Figure 4.4: Examples of Nodes considered in Penalty + Reward Adjustment Techniques(Contd.)

4.3. Proposed Rewards & Penalty Adjustment Techniques

In this section, we mathematically define each of the adjustment based techniques discussed in Section 4.1 and 4.2. At first the two most promising techniques are discussed. This is followed by brief introductions to alternative formulations.

4.3.1. Distance-Based Reward Adjustment: x Closest Depots

Under this method, the rewards value R is computed as a function of the average distance between the x nearest depots and the last node $LN(\pi_{v_j})$ of the route. The method influences assignments of vehicles towards trips that end in locations closer to depots. The value of a reward is computer as per Equation 4.3.

$$R(v_j, T_i, \pi_{v_j}) = \Theta \cdot (1 - \frac{1}{x} \sum_{n \in De} \frac{D_n}{D_{max}}) \tag{4.3}$$

where D_n is the distance of each closest depots n from the $LN(\pi_{v_j})$, De represents the depot nodes considered, x is an operator defined value representing the threshold number of closest depots or pickup locations, and D_{max} represents the maximum distance between all the graph nodes and their respective x depots.

The value computed in Equation 4.3 is normalised with the maximum distance. This limits the value of the term D_n/D_{max} between zero and one. This is done to mitigate the complexity of selecting a tuning

parameter Θ due to the wide range of distances between nodes and x closest depots. Further, the normalized value is subtracted from one to ensure that nearer average distances have a higher reward impact and vice versa.

Intuitively, the adjustment value obtained from Equation 4.3 indicates how quickly a vehicle can travel from the last node of its route to its x closest depots. In particular, a higher reward value indicates that the last node of the vehicle route is closer to its feasible depots and vice versa. By rewarding routes that are closer to depots in a vehicle-trip combination and by specifically assigning such trips to vehicles, we expedite the pickup of new candidates. This is because, with lesser travel times to pickup locations, a vehicle can quickly reach the pickup location for future assigned orders. As a result, it can end up delivering those orders faster. This faster pickup and in turn faster delivery has the potential to free up a vehicle's time to deliver more orders in the same operation time, thereby providing anticipatory benefit to the underlying routing framework.

A potential pitfall of the rewards adjustment method could be that the reward value constantly aims to minimize the distance between the last node of a trip and depots. As a result, in situations where depots are located farther away from high demand regions, the anticipatory approach may influence VGA to assign trips that end closer to depots even if they are located farther away from high-demand zones. This could be counter-intuitive and may result in degraded performance.

Figure 4.5 attempts to provide a clearer understanding of the anticipatory technique. According to the figure, two feasible trips starting from the same node and with the same number of orders can be served by one vehicle. Note that Trip 1 is overlapped by Trip 2 for the route that is common to both. The order destinations are indicated by dark blue nodes for both trips. Each trip roughly travels the same distance and has a comparable cost metric. Trip 1 ends at a location located further away from all three closest depots $(x = 3)$. The termination node of Trip 2 is visibly closer to its depots than Trip 1 and therefore its average distance is lower. As a result, Trip 2 is rewarded more than Trip 1.

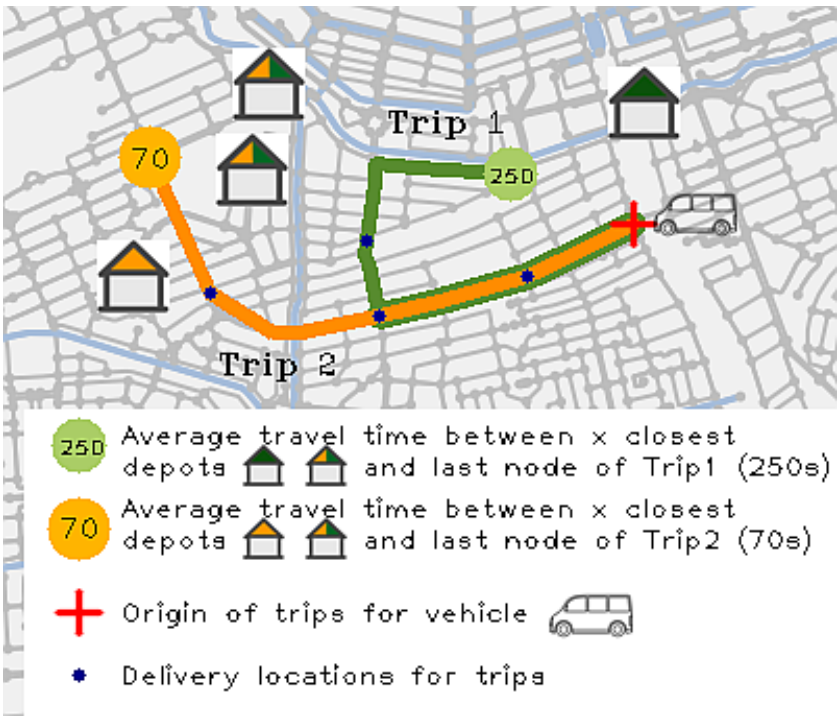


Figure 4.5: Distance-Based Reward Adjustment- x Closest Depots: Two feasible trips starting from the same node and serving the same number of orders can be assigned to one vehicle. Each trip roughly travels the same distance and has a comparable cost metric. Trip 1 ends at a location located further away from all three closest depots $(x = 3)$. The termination node of Trip 2 is visibly closer to its depots than Trip 1 and therefore its average distance is lower. As a result, Trip 2 is rewarded more than Trip 1.

Having understood the impact of the underlying adjustment technique, the question arises on how the

rewards are incorporated within the VGA frameworks assignment procedure P? This is highlighted in Algorithm 3. For every route of a trip-vehicle combination, the rewards value R is computed. Then, the cost of each route is adjusted. Finally, the route with the lowest cost is selected as the route for the vehicle-trip combination and stored in memory. This is repeated till all edges of the vehicle- trip combination are computed. Results obtained from this algorithm are then supplied to the ILP solver for assignments.

Algorithm 3 Rewards Adjustment Value Cost Computation

Input: $G = (N, A)$, $CTVgraph$, β, Θ
Output: Updated Routes and Cost for $CTVgraph$
begin
  for *each* $e(T_i, v_j)$ *in* $CTVgraph$ **do**
    Set $\gamma_{temp} = \infty, \pi_{temp}$

    Find all possible π_{v_r} in each $e(T_i, v_j)$
    while π_{v_r} *in* $e(T_i, v_j)$ **do**
      Obtain $LN(\pi_{v_r})$

      Compute adjustment value,

      $R(\pi_{v_r}) = (1 - \frac{1}{x} \sum_{n \in De_{LN}} \frac{D_n}{D_{max}})$

      Compute cost of route,

      $\gamma(v_r, T_i, \pi_{v_r}) = (1 - \beta) \cdot \sum_{o_i \in T_i} \theta_i + \beta \cdot (TravelTime_i) - \theta \cdot R(\pi_{v_r})$

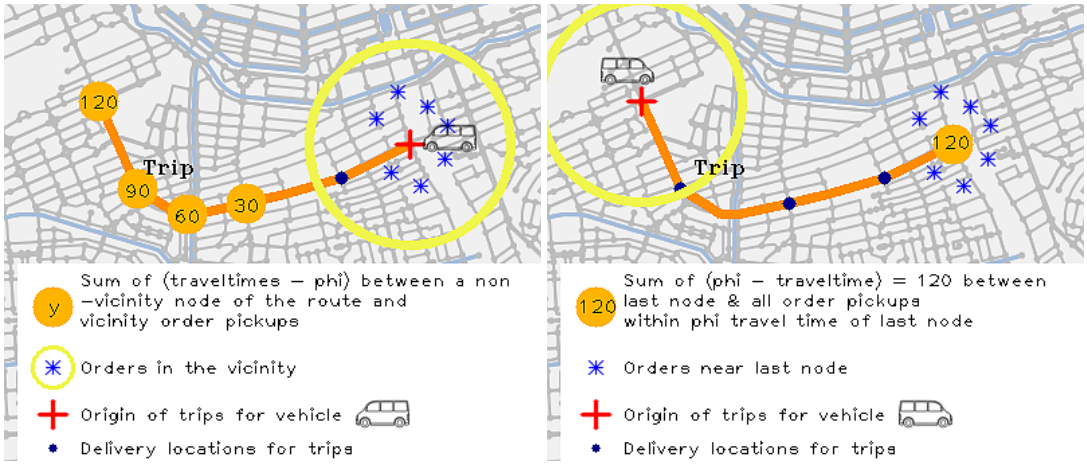
      if $\gamma(v_r, T_i, \pi_{v_r}) < \gamma_{temp}$ **then**
        $\gamma_{temp} = \gamma(v_r, T_i, \pi_{v_r})$

        $\pi_{temp} = \pi_{v_r}$
      end
    end
    $\pi_{v_j} = \pi_{temp}$

    $\gamma_A(v_j, T_i, \pi_{v_j}) = \gamma_{temp}$
    return $\pi_{v_j}, \gamma_A(v_j, T_i, \pi_{v_j})$
  end
end

4.3.2. Candidates-Based Penalty + Rewards

In this approach, trips that direct a vehicle from a higher demand sector to a lower demand sector are penalized whereas trips that take a vehicle from a lower demand sector to a higher demand sector are rewarded. The penalty and rewards value for this technique are a function of more than one node of the trip and utilize a hierarchical computation. To clarify, the algorithm first selects the first node $FN(\pi_{v_j})$ of the trip to determine the number of candidates in its vicinity $C_v \in C$. In our case, vicinity is defined as a region within a threshold distance ϕ. Hence, $C_{v,FN}$ includes all candidates whose pickup locations $pickup(C)$ are within ϕ seconds from the first node. Further, when the percentage of candidates in the vicinity to all candidates on the graph is higher than a threshold value ϵ, the sector is deemed as a high-demand sector. The approach then iterates through all nodes of the trip to ascertain how many of the nodes $N_o(\pi_{v_j})$ lie greater than ϕ distance outside the high demand sector. It then penalizes the trip for each $N_o(\pi_{v_j})$ node. The value of the penalty term is a function of the distance between each $n \in N_o(\pi_{v_j})$ to the candidate pickups that are in the vicinity of the first node $FN(\pi_{v_j})$. This is mathematically defined in Equation 4.4 and depicted in Figure 4.6a.



(a) Illustration of Penalties **(b)** Illustration of Rewards

Figure 4.6: Candidates-Based Penalty + Rewards: Sub-figure 4.6a illustrates the situation of penalties. According to the figure, the first node of the vehicle-trip-route is in a region of high-demand (where order pickups in vicinity > $\epsilon \cdot$ total order pickups). Since, the vehicle-trip-route leaves the vicinity region, it is penalised. This penalty value is equivalent to the summed value of individual non-vicinity node penalties. These non-vicinity node penalties are a function of travel times between the corresponding node (large orange bubbles) and each order in the vicinity of the first node. Alternately, when the vehicle is not in a high demand region, as in Sub-figure 4.6b, the vehicle-trip-route is rewarded as the vehicle moves towards the last node with several order pickups within ϕ distance. The value of the reward is a function of the travel time between the last node and all nearby order pickups.

$$Pe(v_j, T_i, \pi_{v_j}) = \sum_{i,p_j} (TT_{ip_j} - \phi) \qquad \{\forall i \in N_o(\pi_{v_j}); p_j \in pickup(C_{v,FN})\} \tag{4.4}$$

Alternately, if the vehicle is not in a high-demand sector, we compute the value of a reward on the basis of the number of candidates with pickups $pickups(C_{v,LN})$ within the vicinity of the last node $LN(\pi_{v_j})$ of the trip. This is mathematically defined in Equation 4.5 and illustrated in Figure 4.6b.

$$R(v_j, T_i, \pi_{v_j}) = \sum_{p_j} (\phi - TT_{ip_j}) \qquad \{\forall p_j \in pickup(C_{v,LN}); i = LN(\pi_{v_j})\} \tag{4.5}$$

Intuitively, a non-zero penalty value obtained by Equation 4.4 indicates that the vehicle-trip under consideration leaves a high-demand sector and the magnitude of the adjustment value provides directional evidence of how far the vehicle has traversed out of this high demand sector. Alternately, a non-zero reward term indicates a vehicle-trip combination where in the vehicle is entering a sector with higher demand. The magnitude of the adjustment indicates the average distance between the last node of the trip and all the pickup locations of orders in its vicinity.

Incorporating the adjustment terms in cost influences the system in two ways. At first, it influences certain vehicles to remain in high-demand regions and deliver orders to their full capacity. Second, it influences other vehicles to be re-directed towards similar or other areas of high demand. Intuitively, this means that the method limits the possibility of a vehicle ending up in a location where demand may be scarce in the future. It does so by nudging the vehicle towards regions of substantial current demand. It is important to note that adjustment with penalty + reward is based on the underlying assumption that high demand zones are always better than low demand zones. A step-by-step incorporation of the penalty and reward terms are highlighted in Algorithm 4.3.2.

Algorithm 4 Penalty and Reward Adjustment Value Cost Computation

Input: $G = (N, A)$, $CTVgraph$, β, Θ, χ, ϵ, $\{c_1, c_2, ..c_n\} \in C$
Output: Updated Routes and Cost for $CTVgraph$
begin
  for *each* $e(T_i, v_j)$ *in* $CTVgraph$ **do**
    Set $\gamma_{temp} = \infty, \pi_{temp}$

    Find all possible π_{v_r} in each $e(T_i, v_j)$
    while π_{v_r} *in* $e(T_i, v_j)$ **do**
      Obtain $FN(\pi_{v_r})$

      Identify candidates in vicinity, $C_{v,FN}$
      $C_{v,FN} \subset \{c_1, c_2, .., c_i\} \qquad \{\forall c_i \in C \vee travel(FN(\pi_{v_r}), p_i) <= \phi;\ \ p_i \neq FN(\pi_{v_r})\}$

      Identify out of sector nodes of the trip,
      $N_o(\pi_{v_r}) \subset N(\pi_{v_r}) \qquad \{\forall n \in N(\pi_{v_r}) \vee travel(FN(\pi_{v_r}), n) > \phi\}$
      Check if High Demand Sector and Penalise for leaving,
      if $|C_{v,FN}|/|C| > \epsilon$ **then**

$$Pe(\pi_{v_r}) = \sum_{i,p_j} (TT_{ip_j} - \phi) \qquad \{\forall i \in N_o(\pi_{v_j}); \forall p_j \in pickup(C_{v,FN})\} \tag{4.6}$$

        Compute new cost of route,
        $\gamma(v_r, T_i, \pi_{v_r}) = (1 - \beta) \cdot \sum_{o_i \in T_i} \theta_i + \beta \cdot (TravelTime_i) + \chi \cdot Pe(\pi_{v_r})$
      end
      else

$$R(\pi_{v_r}) = \sum_{p_j} (\phi - TT_{ip_j}) \qquad \{i = LN(\pi_{v_j}); \forall p_j \in pickup(C_{v,LN})\} \tag{4.7}$$

        Compute new cost of route,
        $\gamma(v_r, T_i, \pi_{v_r}) = (1 - \beta) \cdot \sum_{o_i \in T_i} \theta_i + \beta \cdot (TravelTime_i) - \kappa \cdot R(\pi_{v_r})$
      end

      if $\gamma(v_r, T_i, \pi_{v_r}) < \gamma_{temp}$ **then**
        $\gamma_{temp} = \gamma(v_r, T_i, \pi_{v_r})$

        $\pi_{temp} = \pi_{v_r}$
      end
    end
    $\pi_{v_j} = \pi_{temp}$

    $\gamma_A(v_j, T_i, \pi_{v_j}) = \gamma_{temp}$
    return $\pi_{v_j}, \gamma_A(v_j, T_i, \pi_{v_j})$
  end
end

4.3.3. Other Adjustment Techniques

In addition to the approaches discussed in Section 4.3.1 and Section 4.3.2, alternate rewards and penalty adjustment techniques were also explored. These approaches were only tested during preliminary simulations and were discarded because of relatively worse performance. They are briefly discussed below:

Orders or Candidates-Based Rewards Adjustment

Orders at Last Node: In this approach, the rewards value is computed as the number of candidate destinations generated at the last node $LN(\pi_{v_j})$ of the vehicle-trip combination during the last 8 minutes of simulation. This is mathematically highlighted in the Equation 4.8 and illustrated in Figure 4.7a.

$$R(v_j, T_i, \pi_{v_j}) = |c_i| \qquad \{\forall c_i \in C \vee g_i = LN(\pi_{v_j}); (t_k - t_i) \leq 480\} \tag{4.8}$$

Candidates at Last Depot: The rewards value is computed as the number of candidates with pickup location corresponding to the last depot $LD(\pi_{v_j})$ of the vehicle-trip combination during the last 8 minutes of simulation. This is mathematically highlighted in the Equation 4.9 and illustrated in Figure 4.7b.

$$R(v_j, T_i, \pi_{v_j}) = |c_i| \qquad \{\forall c_i \in C \vee p_i = LD(\pi_{v_j}); (t_k - t_i) \leq 480\} \tag{4.9}$$

Average of Candidates at x Closest Depots from Last Node: In this rewards type definition, all x closest depots De associated with the last destination node of the given vehicle-trip combination are considered. The rewards value is computed as the average of the total candidate pickups generated at the corresponding depots. Further, only candidates that were requested in the last 8 minutes were used in computation. This is shown in Equation 4.10 and illustrated in Figure 4.7c.

$$R(v_j, T_i, \pi_{v_j}) = \frac{|c_i|}{x} \qquad \{\forall c_i \in C \vee p_i \in De\} \tag{4.10}$$

Maximum of Candidates at x Closest Depots from Last Node: In this rewards type definition, all x closest depots De associated with the last destination node of the given vehicle-trip combination are considered. The rewards value is computed as the maximum of the total candidate pickups generated at each of the considered depots. Further, only candidates that were requested in the last 8 minutes were considered for computation keeping in mind the constraints of the maximum delay. This is shown in Equation 4.11 and illustrated in Figure 4.7d.

$$R(v_j, T_i, \pi_{v_j}) = max_{De}|c_i| \qquad \{\forall c_i \in C \vee p_i \in De;\} \tag{4.11}$$

Candidates at Closest Depot from Last Node

In this rewards adjustment type, candidate pickups generated at a depot located closest (CD) to the last destination node $LN(\pi_{v_j})$ of the vehicle-trip combination are considered for computing the value of the reward. This is highlighted in Equation 4.12 and Figure 4.7e.

$$R(v_j, T_i, \pi_{v_j}) = |c_i| \qquad \{\forall c_i \in C \vee p_i = CD\} \tag{4.12}$$

Distance-Based Rewards Adjustment

Distance of Closest Depot to Last Node: In this adjustment approach, the distance of the closest depot CD from the last node $LN(\pi_{v_j})$ of the vehicle-trip combination is used for computing the value of the reward. This is also illustrated in Figure 4.7f.

$$R(v_j, T_i, \pi_{v_j}) = (1 - \frac{D_{CD}}{D_{max}}) \tag{4.13}$$

Convex Combination of Candidates and Distance of x Closest Depots to Last Node

In this rewards adjustment type, the value of the rewards is computed as a convex combination of the average distance to x depots and the average quantity of candidates at the corresponding depots. The depots De considered are the x closest depots to the last node of the scheduled vehicle-trip combination. This is represented in Equation 4.14 and illustrated in Figure 4.8a

$$R(v_j, T_i, \pi_{v_j}) = \delta \cdot (1 - \frac{1}{x} \sum_{n \in De} \frac{D_n}{D_{max}}) + (1 - \delta) \cdot (\frac{|c_i|}{x}) \qquad \{\forall c_i \in C \vee p_i \in De\} \tag{4.14}$$

where δ represents the weight pertaining to the distance-candidate relevance in the equation.

Candidates-Based Penalisation

Penalization with Constant Term: In this method of adjustment using a penalty, the objective is to restrict vehicles to regions of substantially high demand. It does so by adjusting the cost with a constant penalty term. The penalty is applied on the basis of the first node-last node approach discussed in Section 4.2. This means that the vehicle is penalized for ending a route outside the high-demand sector without taking into account the impact of the distance it moves away from this region. The penalty term is explained in Equation 4.16 and illustrated in Figure 4.8b.

$$Pe(v_j, T_i, \pi_{v_j}) = \sum_{p_i} \rho \qquad \{\forall p_i \in pickup(C_{v,FN})\} \tag{4.15}$$

where ρ is a constant penalty term.

This approach is similar to the method discussed in Section 4.3.2. However, unlike the method discussed in Section 4.3.2, this method does not reward vehicles for moving to more suitable locations. Further, instead of using a penalization term that is a function of the distance between order pick-up locations and the last node of the trip, this approach makes use of a constant penalty term. Furthermore, for computing the penalty we do not consider all nodes of the trip, but only the last node.

Penalization with Travel Time Metric: Building on the penalization approach with a constant term, this approach penalizes the cost in exactly the same manner with the only difference being the selection of a relative penalty term instead of a constant term. This means that in addition to penalizing a vehicle for ending a route outside the high-demand sector, the method also focuses on how far the vehicle has moved from the high-demand sector. This is explained in Equation 4.16 and illustrated in Figure 4.8c.

$$Pe(v_j, T_i, \pi_{v_j}) = \sum_{p_i} (TT_{np_i} - \phi) \qquad \{n = LN(\pi_{v_j}); \forall p_i \in pickup(C_{v,FN})\} \tag{4.16}$$

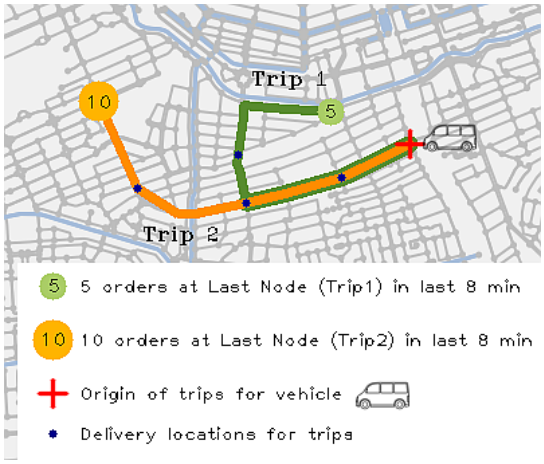
Zone-Based Adjustment (Penalty + Rewards)

The Zone-Based Adjustment anticipatory approach is inspired by the work of Ichoua *et al.*[25]. Similar to the approach taken by [25] in their study, we divide the graph into M geographical zones. We do this using an Integer Linear Program that divides the graph into zones of relatively similar sizes where each node is reachable by its center within $t_M = 100$ seconds. The formulation of sub-regions is detailed further in Chapter 5. However, unlike [25], we do not determine whether the vehicle should wait or move to the next destination based on a probability factor dependent on the time at which orders are placed. Rather, we adjust the cost by means of a penalty or reward term depending on the vehicle's movement. To do this, we first compute the relative frequency of orders in each zone for every time iteration. If the vehicle moves from a zone of lower relative frequency to a higher relative frequency, it is rewarded. On the other hand, if the vehicle moves from a zone of higher relative frequency to a zone of lower relative frequency, it is penalized. Finally, if the vehicle remains in the same zone, no alterations to vehicle-trip costs are made. The penalization and rewards are computed as per Equations 4.17 and 4.18 respectively.

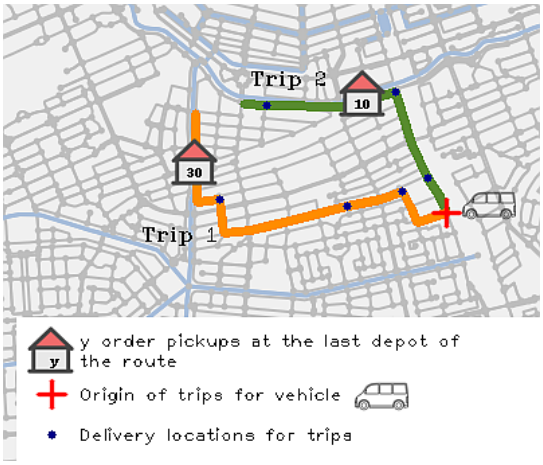
$$Pe(v_j, T_i, \pi_{v_j}) = -\rho \qquad P(Z_D) < P(Z_O) \tag{4.17}$$

$$R(v_j, T_i, \pi_{v_j}) = \rho \qquad P(Z_O) < P(Z_D) \tag{4.18}$$

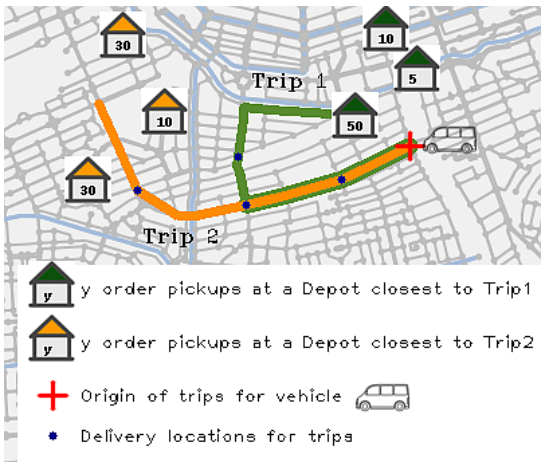
where $P(Z_D)$ and $P(Z_O)$ are relative frequencies of destination zone Z_D and origin zone Z_O respectively. Further, ρ represents a constant penalty-reward term that is empirically determined.



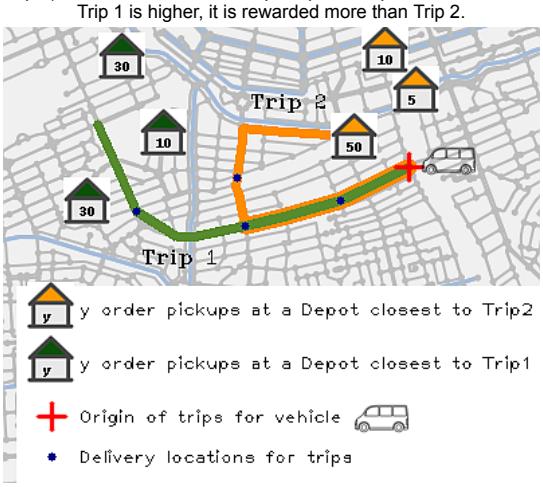
(a) Candidates-based Rewards Adjustment - At Last Node: Trip 2 has a higher number of orders placed at its last node in the last 8 minutes than Trip 1 (5 orders for Trip 1, and 10 orders for Trip 2). Therefore, Trip 2 is rewarded more than Trip 1.



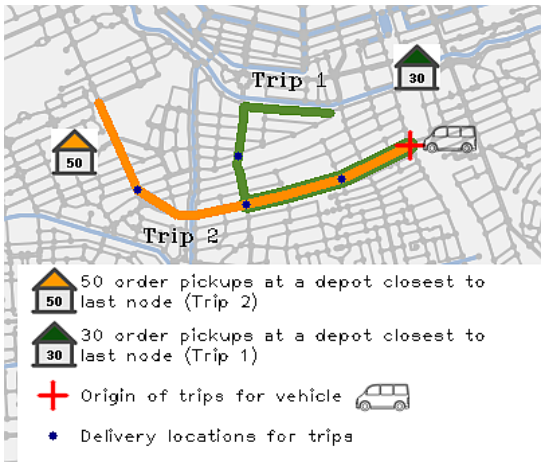
(b) Candidates-Based Rewards Adjustment - At Last Depot: The last depot before the termination of both trips has different pick-up orders as shown in their icons (10 for Trip 2, and 30 for Trip 1). As the number of order pickups for depot associated with Trip 1 is higher, it is rewarded more than Trip 2.



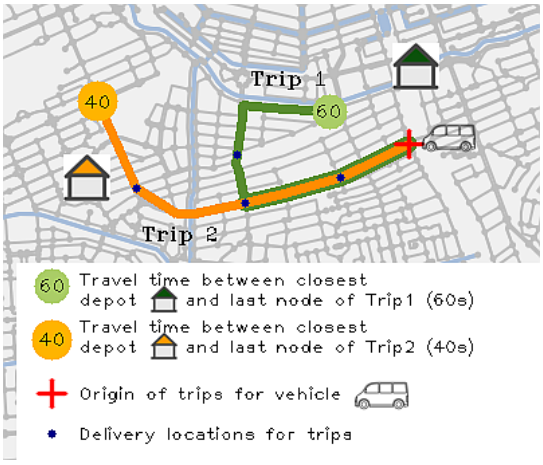
(c) Candidates-Based Rewards Adjustment - Average of Candidates at x Closest Depots of Last Node: Both Trips have 3 unique closest depots. Trip 1 ends near depots with 5, 10, & 50 order pickups whereas Trip 2 ends near the closest depots with 30 order pickups each. The average number of candidates for Trip 1 is, therefore, 21.67 & that of Trip 2 is 30. As a result, Trip 2 is rewarded more than Trip 1.



(d) Candidates-Based Rewards Adjustment - Maximum of Candidates at x Closest Depots of Last Node: Both Trips have 3 unique closest depots. Trip 1 ends near depots with 5, 10, & 50 order pickups whereas Trip 2 ends near the closest depots with 30 order pickups each. As maximum order pickups are associated with the depot closer to Trip 2, it receives a higher reward than Trip 1.

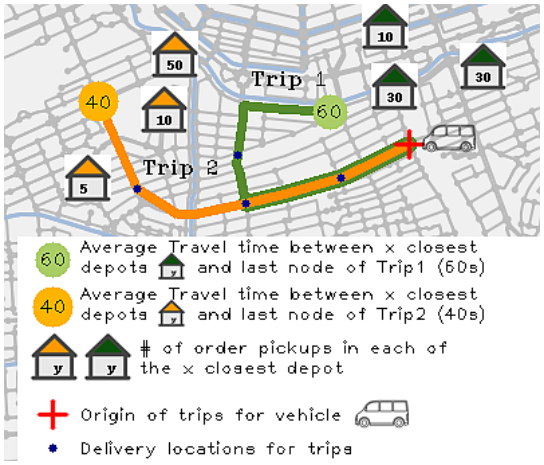


(e) Candidates-Based Rewards Adjustment - Candidates at the closest depot from the last node: Rewards are computed proportionally to order pickups at the closest depot. As order pickups at the depot closest to Trip 2 are higher than that of Trip 1, Trip 2 is rewarded more than Trip 1.

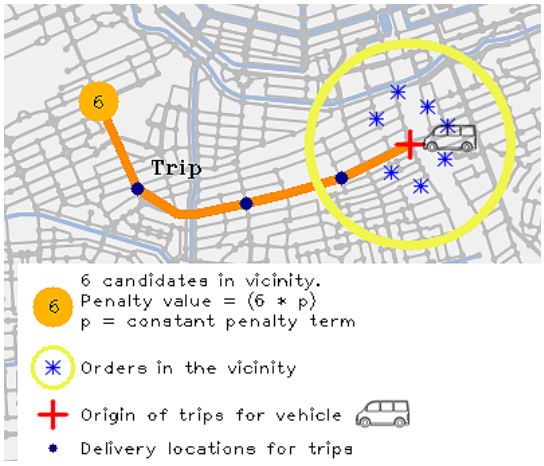


(f) Distance-based Rewards Adjustment - Distance of closest depot from the last node: Rewards are computed inversely proportional to the distance between the closest depot and last node of each trip. The last node of Trip 2 is 40 seconds away from its closest depot. On the other hand, the last node of Trip 1 is 60 seconds away from its closest depot. Hence, Trip 2 is rewarded more than Trip 1.

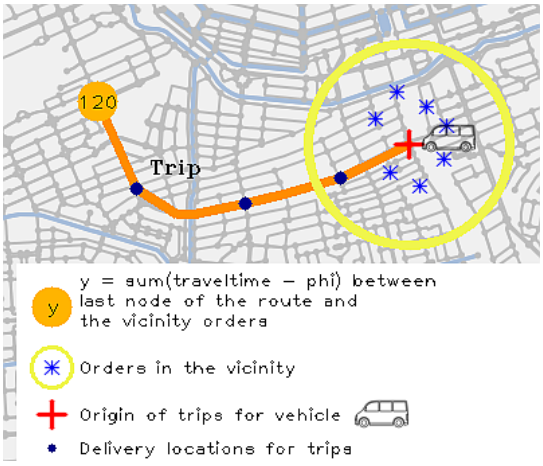
Figure 4.7: Other Adjustment Techniques: All sub-figures have two Trips 1 and 2 that serve the same number of orders and have the same cost. Each sub-figure represents an independent adjustment technique that utilizes unique information to modify the costs. These are explained in individual sub-figures.



(a) Convex Combination of Candidates and Distance of x Closest Depots to Last Node: The value of rewards is computed as a convex combination of two values- the first is a function of average travel time between the last node of the vehicle-trip-route and its x closest depots, and the second is the average number of order pickups at the same x depots. In this case, Trip 1 has an average travel time of 60s and an average order pickup of 23.33 (30+10+30/3). On the other hand, Trip 2 has an average travel time of 40s, and 21.57 order pickups at corresponding depots. As Trip 1 has a higher travel time than Trip 2, and a higher number of average order pickups, it is not explicit which Trip will be rewarded more. The weight of the convex combination will determine which of the average travel time and order pickups will be prioritized for rewards. If full priority is given to travel time, Trip 2 will be rewarded more than Trip 1. On the other hand, if full priority is given to order pickups, Trip 1 will be rewarded more than Trip 2.



(b) Candidate Based Penalty- Penalization with constant term: The first node of the vehicle-trip-route is in a region of high-demand (where order pickups in vicinity > $\epsilon \cdot$ total order pickups). Since the vehicle-trip-route leaves the vicinity region, it is penalized. This penalty value is equivalent to a constant value ρ summed repeatedly till the number of repetitions is equal to the number of order pickups in the vicinity of the first node of the vehicle-trip-route.



(c) Candidate Based Penalty- Penalization with travel time metric: According to the figure, the first node of the vehicle-trip-route is in a region of high-demand (where order pickups in vicinity > $\epsilon \cdot$ total order pickups). Since the vehicle-trip-route leaves the vicinity region, it is penalized. This penalty value is equivalent to the penalty at the last node, which is a function of travel times between the corresponding node (large orange bubble) and each order in the vicinity of the first node.

Figure 4.8: Other Adjustment Techniques (contd.): All sub-figures have two Trips 1 and 2 that serve the same number of orders and have the same cost. Each sub-figure represents an independent adjustment technique that utilizes unique information to modify the costs. These are explained in individual sub-figures.

5

Demand Distribution & Instance Setup

The real potential of any anticipatory technique can only be ascertained if it offers consistent performance enhancement over the myopic simulation on a diverse set of situations. This is owing to the fact that no two days or even hours of grocery delivery operations have the same demand distribution. Hence, it is imperative for the anticipatory approach to be evaluated on different demand distributions to confirm the merits and limits of the approach.

At the time of conducting this study, to the best of our knowledge, no open-source real demand data sets were available for testing the limits and capabilities of our solution methodology. Further, popular simulation instances in literature [55][61][28] were limited in terms of the number of orders and the unique spatial-temporal characteristics they offered. It, therefore, became crucial to generate our own unique data set instances to evaluate the performance of our proposed approaches. For this purpose, a total of 85 unique data sets were created majorly differing in their spatial-temporal characteristics. A temporal characteristic pertains to the distribution of demand across time whereas a spatial characteristic represents the distribution of demand over the graph. In addition, other forms of stochasticity were also introduced in the instance set up to limit any performance bias. In the following sections, we first study the temporal characteristics of demand. Second, we capture the spatial aspects of demand. In the next section, we cover the generation of five unique data sets that mimick real-world data. Finally, we explore external factors that add more stochasticity to the instance setup.

5.1. Temporal Characteristics

The temporal characteristic of a demand distribution focuses on the way new orders are placed across time during a simulation period.

Most studies in literature model the demand on one of the following four random distributions- non-homogeneous Poisson, homogeneous Poisson, Gaussian, or Uniform. Consequently, the question arises which temporal patterns are suitable for our use case.

To identify the suitability of a temporal distribution, we refer to the work done by Van Lon *et al.*[65]. To characterize the nature of each temporal distribution, Van Lon *et al.*, distinguishes them on the basis of a self-defined measure called dynamism. Their definition of dynamism refers to the degree of change of continuous activity. Intuitively, this means that a highly dynamic scenario changes continuously whereas a less dynamic scenario changes occasionally. This is illustrated by means of Figure 5.1. Additionally, a complete explanation of dynamism is provided in Appendix H.

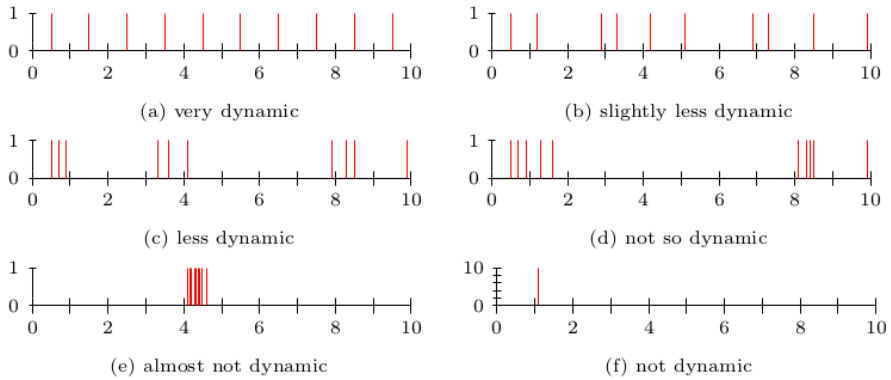


Figure 5.1: Degree of Dynamism: Visualization of order arrival times, each red bar indicates an event in which a new order is announced. The figures (a) to (f) are presented in decreasing order of dynamism. In (a) the events have equal interarrival times and are nicely distributed over the period, in (b) and (c) we see that changes occur less frequently. In (d) and (e) all events arrive in one or two batches making it less continuous and therefore less dynamic. In (f) all 10 events arrive at the same time resulting in a scenario with no dynamism [65].

From their work, they identify that Uniform and Gaussian distributions are typically more dynamic than their Poisson counterparts. This is illustrated in Figure 5.2. We re-implement their work using test data sets and identify that our results highly correlate with their findings. This is highlighted in Figure 5.3.

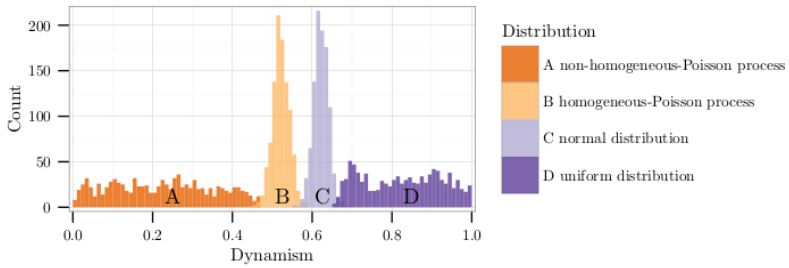


Figure 5.2: Degree of Dynamism for the four commonly produced temporal distributions [65]

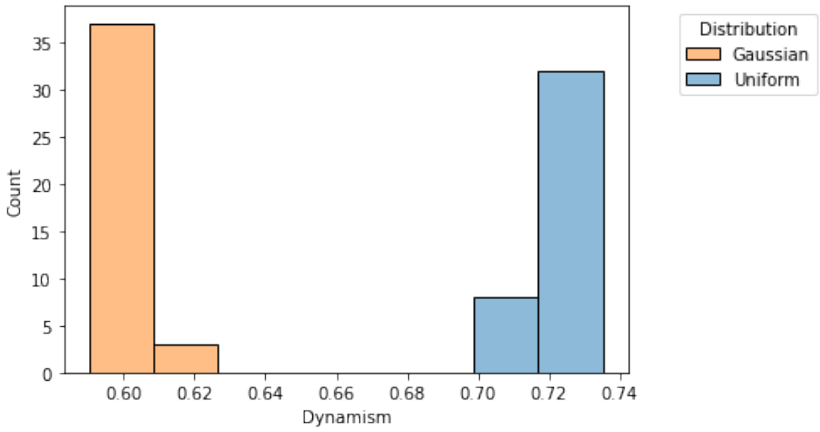
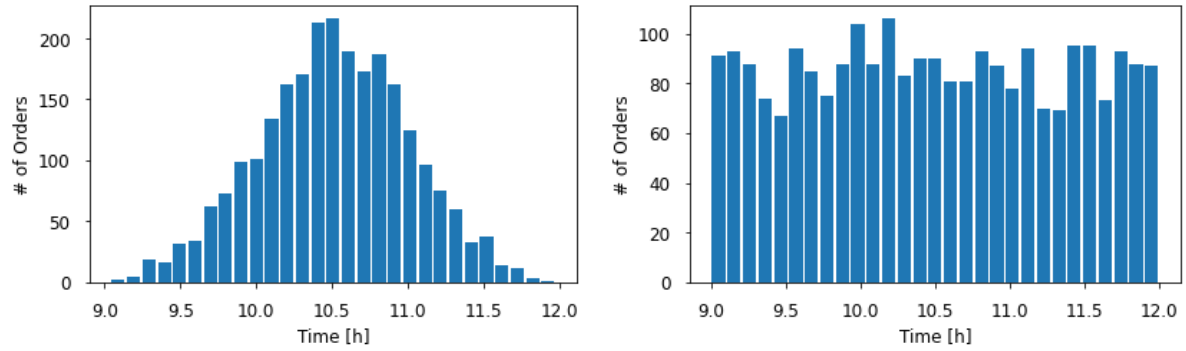


Figure 5.3: Degree of Dynamism for test instances

As the focus of our study is to build robust anticipatory techniques for fast-moving on-demand grocery deliveries, we aim to conduct our experiments on highly dynamic demand situations. As a result, modeling our demand distributions as Uniform and Gaussian temporal distributions over Poisson distributions

seems to be an appropriate choice. Using this rationale, we generate demand distributions for 80 data instances with 40 instances distributed Uniformly, and 40 instances distributed as a pure Gaussian. From the generated data instances, one instance from each of the temporal distribution categories is illustrated in Figures 5.4a and 5.4b.



(a) Gaussian Temporal Distribution

(b) Uniform Temporal Distribution

Figure 5.4: Temporal Demand Patterns: Distribution of orders with time. Each bar represents the orders accumulated in a period of 6 minutes

5.2. Spatial Characteristics

The spatial characteristic of the demand pertains to how the orders' destinations are spatially distributed on the graph network during a simulation period.

To the best of our knowledge, most literature deploys a uniform distribution of demand across the entire graph region with only a few notable exceptions [61][55] where demand is also clustered into two or more distributions. Furthermore, limited information is available about how the demand is distributed within each cluster. In our work, we identify two independent factors to characterize spatial distributions. These are-

- Whether the demand is clustered in specific sub-regions of the graph?
- How the demand is distributed in each of its specific clusters or the entire graph?

We discuss the clustering of demand in Section 5.2.1 whereas the nature of the distribution is classified further in Section 5.2.2.

5.2.1. Clustered vs Non-clustered Distributions

Demand for grocery delivery in an urban setting is not always evenly distributed. Some regions are densely populated while others have sparsely located residences. Furthermore, some regions tend to utilize online on-demand deliveries more than others. For instance, areas with a higher millennial population are more likely to order online than other demographics [43]. Keeping this in mind, having the demand spread throughout the map may not necessarily represent a true picture of demand in an urban setting.

To model this uneven distribution of demand we generate two separate demand models- clustered distributions and non-clustered distributions. Clustered distributions generate order destinations at specific sub-regions of the graph creating an unevenly distributed demand pattern. On the other hand, non-clustered regions can be considered as a 0 clustered distribution where the demand is distributed throughout the entire graph network. In our work, clustered data set instances are divided into clusters of 3, 5, and 7 sub-regions of the graph. This is done for the sake of adding more generality to our simulation instances.

To generate suitable clusters of sub-regions within the graph network, we make use of an integer linear program similar to the one used by Waller *et al.*[67]. The notations for the integer linear program are highlighted in Table 5.1 and the ILP itself is illustrated by means of Algorithm 5. At first, a reachability matrix $R(i,j)$ is prepared from the travel time matrix $\tau(i,j)$. The reachability matrix comprises all

origin nodes along rows and destination nodes along columns. A value of 1 across origin node i and destination node j indicates that j is reachable by i. A value of 0 indicates otherwise. Reachability can be defined as a binary term indicating whether the time required to travel between two nodes is lesser than a threshold time period Γ. Furthermore, a binary variable x is defined where $x_i = 1$ if vertex i is used as a cluster center. Equation 5.1minimizes the number of nodes in the graph given the constraint that every destination j must be reachable by at least one cluster center i. This is highlighted by Equation 5.2. The solution to the ILP solver indicates the minimum number of cluster centers required to cover the entire graph in terms of reachability. The nodes corresponding to cluster centers represent the center of the sub-region and their reachable destinations form the nodes for the sub-regions of the graph. This is indicated by equation 5.3. Note that in case a node is reachable by two cluster centers, a random assignment of the node to the sub-regions takes place. A solution by the ILP is illustrated in Figure 5.5.

Table 5.1: Explanation of variables for sub-region computation

Variable	Explanation
$\tau(i,j)$	Travel time matrix indicating time taken to travel from node i to j
$g1, g2, ..., g_n$	n sub-regions or clusters of the Graph G
Γ	Reachability threshold. All distances beyond this threshold are considered unreachable. We empirically determine a suitable value to be 150 seconds.

Algorithm 5 Sub-region computation in Graph using ILP

Input: $G = (N, A)$, $\tau(i,j)$
Output: $\{g1, g2, ..., g_n\} \in G$
begin
$R(i,j) \leftarrow \tau(i,j) < \Gamma$

Solve;

$$\sum_{optim} = argmin_x \sum_{i \in N} x_i \tag{5.1}$$

with constraints-

$$\sum_{i \in N} (R(i,j) * x_i) >= 1 \qquad \forall j \in N \tag{5.2}$$

for $i \vee x_i \leftarrow 1$ **do**

$$g_i \leftarrow g_i, j \qquad \forall j \in R(i,j) > 0 \tag{5.3}$$

end
return $g_1, g_2, ..g_n$
end

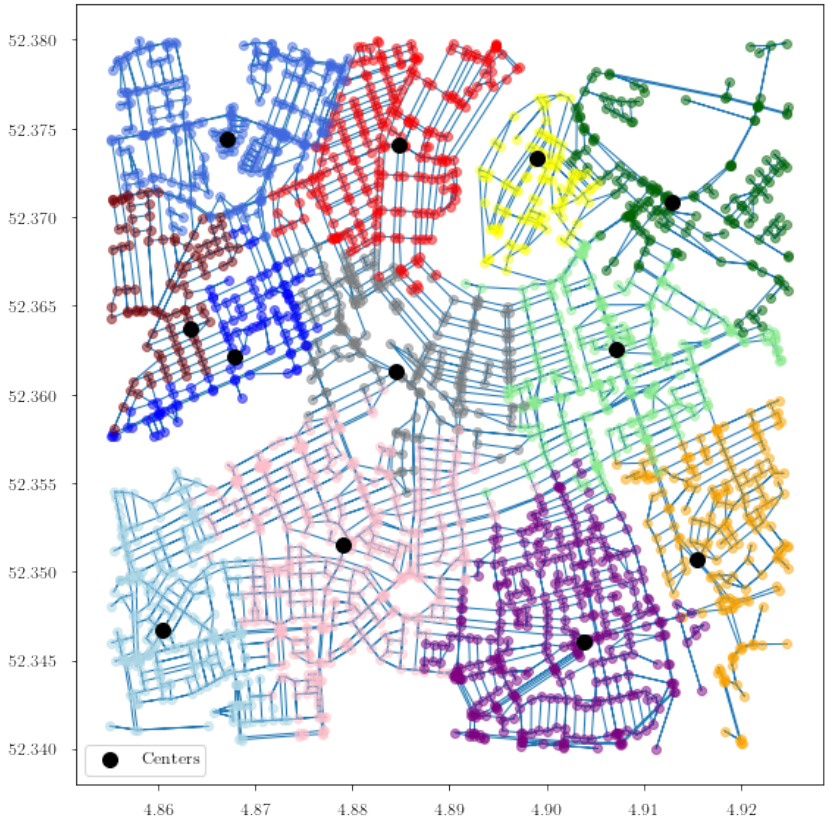
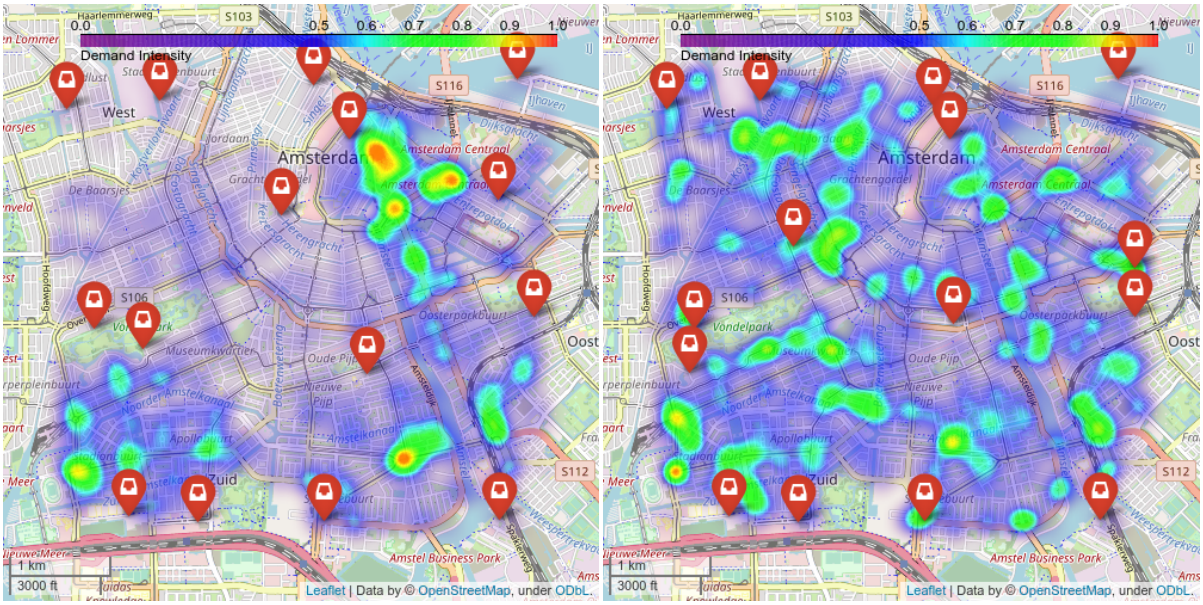


Figure 5.5: Sub-regions/Clusters obtained from the ILP solver: A total of 12 sub-regions are identified. Each unique color represents a sub-region. The black nodes represent the cluster centers.

Once all the sub-regions of the graphs are obtained n sub-regions are selected at random, where n can be 3, 5, or 7. Further, a random demand factor between 0.2 and 1 is determined for each of the selected sub-regions. Each value of the demand factor indicates the ratio of total orders that are to be generated for its corresponding sub-region. Additional noise is also to be added to the generated demand. For this purpose, a noise value is defined that represents the share of total demand that will be generated anywhere on the graph irrespective of the clusters. To generate and append the noise, a standard noise value of 0.15 is selected and appended to the list of demand factors. These demand factors are then normalized, to sum up to a total of 1. Figure 5.6 highlights two demand distributions with clustered demand and non-clustered demand respectively.



(a) Clustered Demand Distribution with 3 clusters

(b) Non-clustered Demand Distribution

Figure 5.6: Spatial Demand Distribution on the Graph: Comparison between clustered demand patterns and non-clustered demand patterns. The clusters represent sub-regions of the Graph where the majority of the orders are generated. Sub-region computation is done by means of the ILP solver. Red colors represent zones with the highest demand. The intensity of demand diminishes as color intensity reduces from red to blue.

5.2.2. Uniform vs Gaussian Distributions

In addition to the clustering of demand into sub-regions, the nature in which orders are spread within each sub-region may also impact the performance of the proposed approach. As a result, we model spatial demand using two separate distributions techniques- Gaussian normal distribution and Uniform distribution.

Although the generation of a Uniform distribution is fairly simple, deployment of a true Gaussian distribution on a 2-dimensional graph proved to be non-trivial. Several intuitions were developed for modeling a complete Gaussian distribution. These are coined displacement to the center method, index method, and independent axis method. Amongst these, the independent axis method proved to be robust and was selected for generating Gaussian distributions. We introduce the independent axis method in this section by means of Algorithm 6. The other methods are covered in Appendix I for the interested reader.

In the independent axis method, the mean and standard deviation are computed independently for latitudes and longitudes of all the nodes in the graph/sub-region. In the subsequent steps, two independent Gaussian distributions are generated using the mean and standard deviation of longitudes and latitudes respectively. The independent distributions are then combined to form the complete coordinates of the distribution. Then for each coordinate from the list of generated coordinates, the closest node of the graph is determined and stored as the station corresponding to the order. Figures 5.7a and 5.7b illustrate examples of generated Uniform and Gaussian spatial distributions respectively. According to the figure, the Gaussian distribution has orders distributed around the cluster center which in this case is the center of the graph. On the other hand, uniform distribution has no such patterns and the orders are arranged throughout the graph region.

Algorithm 6 Generation of Spatial Demand using Independent Axis Method

input: num_orders, $G = (N, A)$
output: $demand\_nodes$
begin
Computing Gaussian Distribution:
$list_{Longitude} \leftarrow N \cdot longitudes$
$mean_x \leftarrow mean(list_{Longitude})$
$std_x \leftarrow std(list_{Longitude})$
$list_{Latitude} \leftarrow N \cdot latitudes$
$mean_y \leftarrow mean(list_{Latitude})$
$std_y \leftarrow std(list_{Latitude})$
$coord_x \leftarrow Gauss(mean_x, std_x, num\_orders)$
$coord_y \leftarrow Gauss(mean_y, std_y, num\_orders)$
$coord \leftarrow combine(coord_x, coord_y)$
Finding nearest Node on Graph:
for $destination \in coord$ **do**
$dist_{max} \leftarrow \infty$
for $node \in N$ **do**
$dlat \leftarrow destination.longitude - node.longitude$
$dlon \leftarrow destination.latitude - node.latitude$

$$a = sin(dlat/2)^2 + cos(node \cdot latitude) * cos(destination \cdot latitude) * sin(dlon/2)^2 \quad (5.4)$$

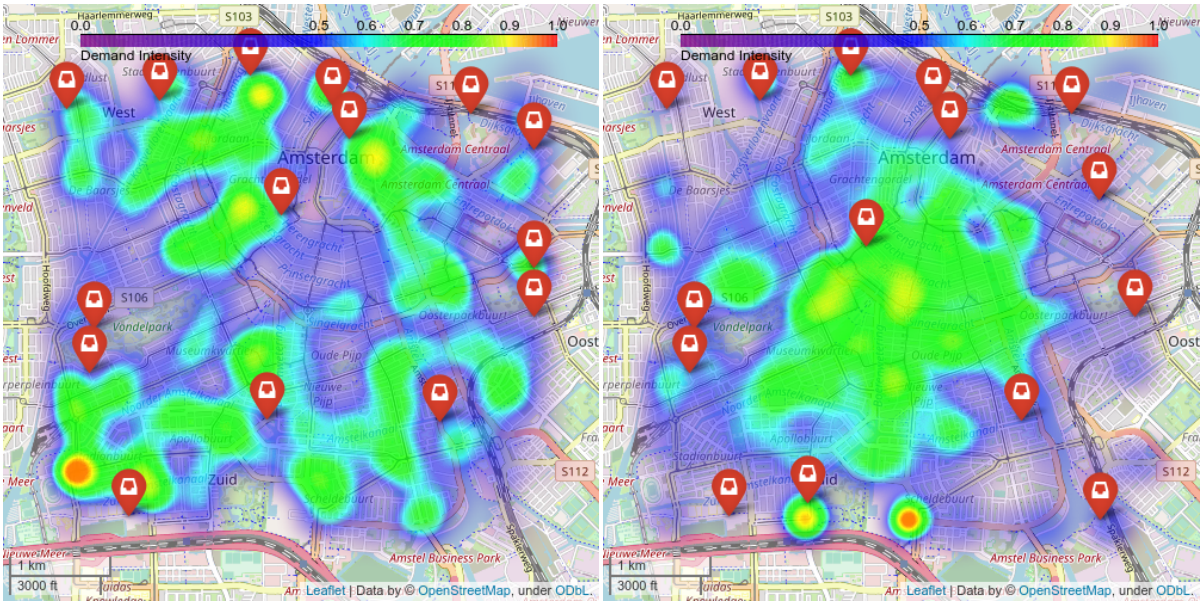
$$c = 2 * atan2(sqrt(a), sqrt(1-a)) \quad (5.5)$$

$$dist = 6373 \cdot c \quad (5.6)$$

if $dis \leq dist_{max}$ **then**
$dist_{max} = dis$
$best\_node \leftarrow node$
end
end
$demand\_nodes \leftarrow demand\_nodes, best\_node$
end
return $demand\_nodes$
end

5.3. Demand Instances

The various spatial-temporal combinations give rise to 16 unique data set types. These are highlighted in Table 5.2. Distributions from each data set type are sampled to obtain 5 independent instances. As a result, we generate a total of 80 unique instances. Further, each data set instance has the same number of orders set to 2500, and the same time period of distribution set to 3 hours. Same order quantity and time span are selected in order to simplify the comparison of solution methodologies; Any variation in performance can be largely attributed to changes in spatial-temporal configurations of the demand.



(a) Uniform Spatial Demand Distribution

(b) Gaussian Spatial Demand Distribution

Figure 5.7: Comparison between a Uniform spatial demand distribution and a Gaussian spatial demand distribution: The Gaussian distribution has orders distributed around the cluster center which in this case is the center of the graph. On the other hand, uniform distribution has no such patterns and the orders are arranged throughout the graph region. Red colors represent zones with the highest demand. The intensity of demand diminishes as color intensity reduces from red to blue.

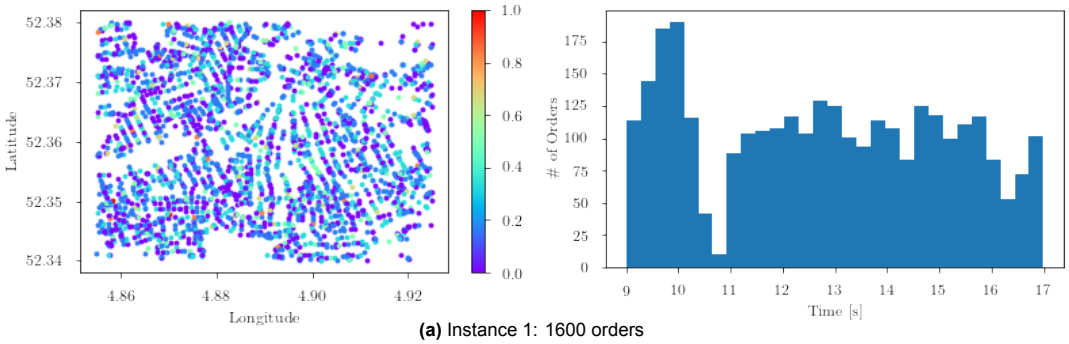
Table 5.2: Unique Data Set Types

Temporal Characteristic	Spatial Characteristic	Clusters
Uniform	Uniform	-
Uniform	Uniform Clustered	3
Uniform	Uniform Clustered	5
Uniform	Uniform Clustered	7
Uniform	Gaussian	-
Uniform	Gaussian Clustered	3
Uniform	Gaussian Clustered	5
Uniform	Gaussian Clustered	7
Gaussian	Uniform	-
Gaussian	Uniform Clustered	3
Gaussian	Uniform Clustered	5
Gaussian	Uniform Clustered	7
Gaussian	Gaussian	-
Gaussian	Gaussian Clustered	3
Gaussian	Gaussian Clustered	5
Gaussian	Gaussian Clustered	7

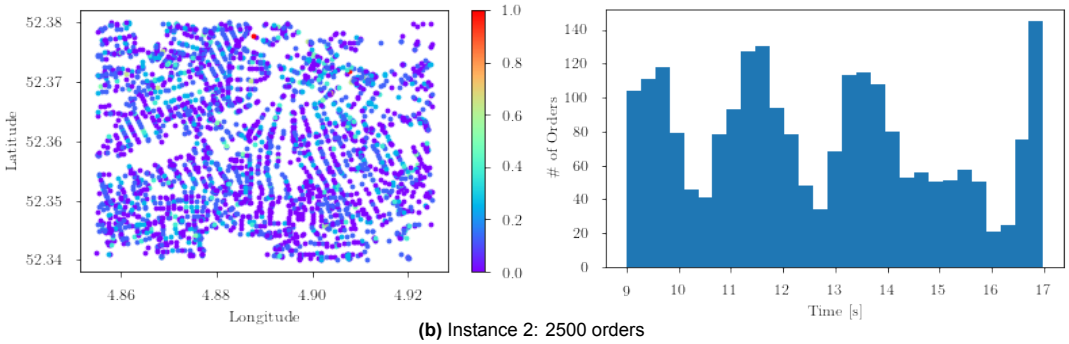
5.4. Realistic Simulation Instances

In addition to the 80 demand instances mentioned previously, 5 additional demand instances were generated by combining multiple temporal and spatial characteristics in a single instance. The goal of

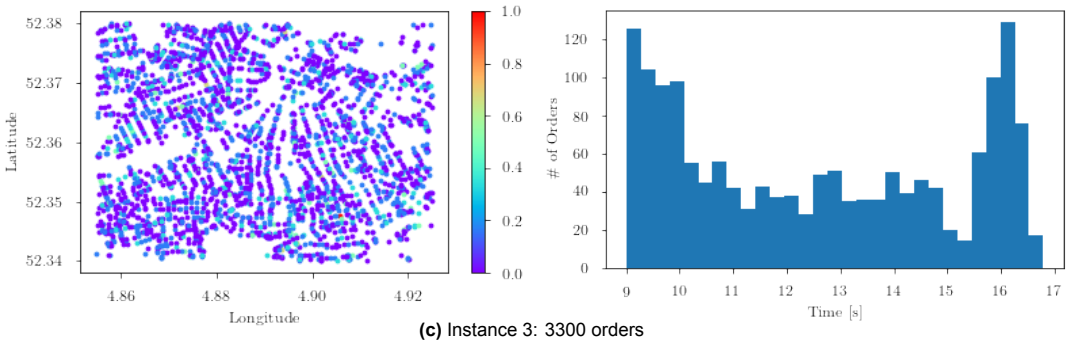
generating these instances was to obtain close-to real-world scenarios that could indicate the performance improvement of our proposed anticipatory approach in a large-scale real-world setting. These instances are illustrated in the Figure 5.8. The total demand of instances 1-5 increases from 1600 orders to 3600 orders. The main purpose of varying the demand was to test the capability of our anticipatory approaches as the overall demand per unit time increases. This was achieved by generating additional instances within the same time interval and superimposing the instances on the main demand instance. All instanced have a constant period of operation between 9 am-5 pm.



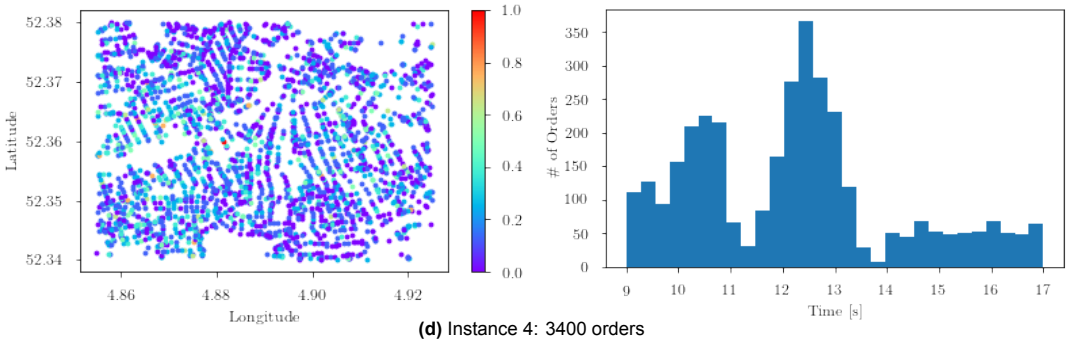
(a) Instance 1: 1600 orders



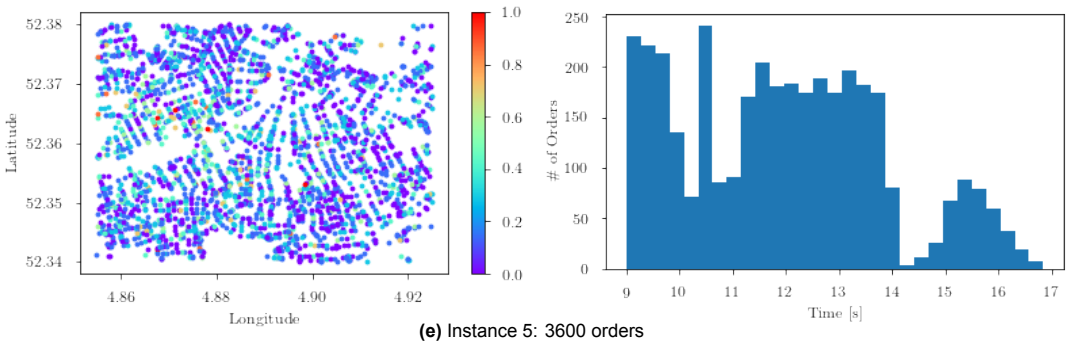
(b) Instance 2: 2500 orders



(c) Instance 3: 3300 orders



(d) Instance 4: 3400 orders



(e) Instance 5: 3600 orders

Figure 5.8: Realistic Instances: The instances are generated by superimposing multiple demand instances over each other in a random fashion. The demand increases from 1600-3600 orders from instance 1-5. This is done to test the capability of anticipation under different demand scenarios. The temporal distribution is divided into 30 bins each and the spatial distribution is normalized. In the spatial graphs, nodes with red color have the highest demand intensity. This decreases as the color of the nodes turn dark blue. This is also represented by the color bars next to the figures.

5.5. Variation due to Depot Distribution on the Graph

The decision of how depots are located on the graph can have a big impact on the performance of the algorithm. For instance, if the majority of the depots are located in one part of the graph, vehicles will have to travel long distances to serve orders in other parts of the graph. This may result in longer routes and a reduced number of orders served. In the worst-case scenario, parts of the graph may be completely under-served. On the other hand, optimally placed depots may increase the service rate but may not be representative of real-world routing.

It is safe to surmise that this impact may become even more profound with the anticipatory techniques discussed in Chapter 4. The first of the two main approaches push vehicles towards trips that end closer to depot locations. This means that our algorithm will be even more likely to avoid any high-demand regions that are farther from depots than the base algorithm. On the other hand, our second approach constantly penalizes the vehicles for leaving high-demand regions. This may result in vehicles loading more orders in the same trip limiting the benefit of rolling horizon-based optimization.

As a result, the decision of selecting depot locations is an imperative part of the instance setup. We identified three main approaches for the distribution of depots on the graph- the K-center distribution method, the Uniform distribution method, and Shorted Distance distribution method. Amongst these, we identified the K-center distribution method to be suitable for our use case. Other approaches along with our reasoning for selecting the K-center method are detailed further in Appendix J.

5.5.1. K-Center Distribution Method

The K-Center algorithm aims to minimize the maximum distance of any node with respect to their closest depot. In this method, the first depot is selected randomly. Then every subsequent depot is determined in such a way that its location is farthest from every other depot on the graph. This is repeated until k depots are obtained. The algorithm was executed five times and the depot distribution that had the lowest average distance to every other node on the graph was selected. This depot distribution is highlighted in the Figure 5.9.

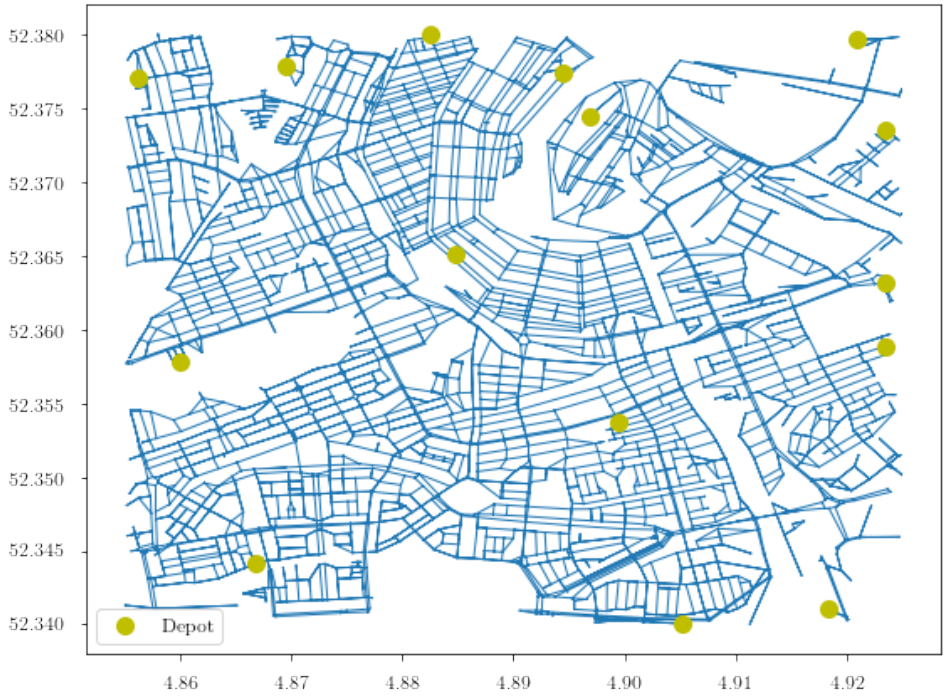


Figure 5.9: Depot distribution using the K-center Method: 15 depots are placed throughout the graph and are represented by the yellow indicators.

6

Evaluation

In this chapter, we test the proposed anticipatory approaches from Chapter 4 and analyse their performance in comparison to the base method introduced in Chapter 3.

For the simulation, a fleet of 10 vehicles is chosen, with each vehicle having a total capacity of 10. From a research standpoint, this is a fairly large fleet to route in simulation and is only possible due to the large-scale routing capabilities of VGA. From an operational standpoint, however, this is still a relatively small fleet, unable to serve a very large number of orders. This was deliberately chosen so as to obtain a high rejection rate that enables us to analyse rejections in a crisp manner. Further, the impact of anticipation and how it affects the different performance KPIs can be evaluated clearly as any change in performance will be largely attributable to the anticipatory technique itself.

Other parameters are mostly adopted from the implementation by [28]. These include the maximum size of the trip that is limited to the capacity of the vehicle and equals 10. A maximum delay δ_{delay} of 8 minutes with no possibility of re-insertion of ignored orders. Further, each order has only 5 possible pickup locations $(x = 5)$. The decision intervals of the algorithm is set to 100 seconds, i.e., δt occurs after every 100 seconds. The penalty to ignore orders is set to a high value of 10^4. As such, these parameters were determined empirically to have the best performance with limited resources in the study by Kronmueller *et al.* and remain constant throughout the simulations. A complete list of the parameters is given in Appendix K. Further, aspects of simulation such as details of the graph, and the vehicle movement are covered in Appendix L.

We evaluate the capabilities of our proposed method by studying multiple performance KPIs. The first KPI of concern is the service rate, i.e., the percentage of total orders that got served. In some cases, we switch the service rate with the rejection rate that represents the percentage of rejections of the overall orders. We then look at other customer experience metrics. These include the total time it takes to deliver the order from the time of its request, the delay in delivery time relative to its optimal delivery time, the time a parcel remains on a vehicle, and the duration an order waits at the depot before being picked up by a vehicle. In addition, we also look at the operator's cost which is determined in terms of the total travelled distance of all vehicles. Finally, we observe the capacity utilization of our vehicles. We do this by comparing the mean loaded parcels for each method. We also compute the a-posterior cost given by Equation 3.1 to give an absolute performance comparison.

In the following sections, we first analyse the performance of single simulation instances to give the reader an intuition of the comparative numbers, magnitudes and behaviour of the proposed approaches. This is highlighted in Section 6.1. We then investigate our most promising approaches on the realistic instances defined in Section 5.4. The goal is to see the large-scale long-run effects of the proposed techniques. We further evaluate the sensitivity of the reward weights to analyse the robustness of the approach. These are showcased in Section 6.2. We then collate our findings from running our simulations under high-workload scenarios present in the 80 demand instances defined in Section 5.3. We further deep-dive on the impact different demand distributions have on our anticipatory techniques. These are covered in Section 6.3.

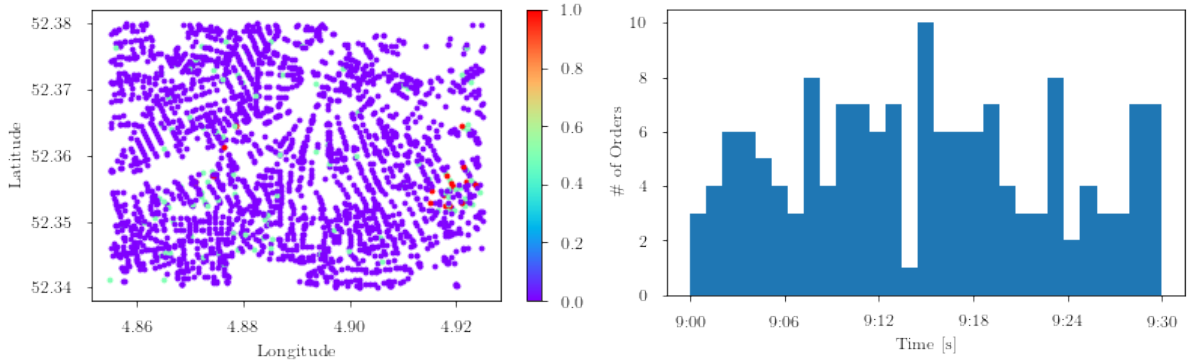
6.1. Comparative Performance on Single Data Instances

A comparison of multiple anticipatory techniques and the myopic simulation is undertaken on sample data set instances. These data set instances are only small-scale versions of the larger data sets discussed in Chapter 5. The motivation for selecting sample data set instances was twofold. First, it allowed us to run our initial simulations quickly, discarding poorly performing anticipatory techniques early on. Second, it allowed us to get an early insight that the distribution of demand impacts the results from our adjustment techniques - adjustment with rewards, and adjustment with penalty and rewards-differently.

6.1.1. Assessing Performance of Anticipation by Introducing Rewards

In this section, we analyse the performance of several rewards adjustment techniques covered in Section 4.3 by running the simulations on a sample data set instance indicated in Figure 6.1.

A smaller sample data instance other than the ones discussed in Chapter 5 was generated for comparing multiple variations of the rewards adjustment technique. This was done to quickly evaluate and identify the most promising rewards adjustment approach for further analysis. The data set instance comprises a clustered Gaussian spatial distribution of 3 clusters and a random uniform temporal distribution. The total number of orders is 150 and the period of operation is 30 minutes.



(a) Spatial Distribution of Demand

(b) Temporal Distribution of Demand

Figure 6.1: Preliminary Instance for Testing Anticipation with Adjustments using Rewards: The data set instance comprises a clustered Gaussian spatial distribution of 3 clusters and a random uniform temporal distribution. The total number of orders is 150 and the period of operation is 30 minutes. The spatial graph is normalised such that red nodes indicate the highest demand and violet nodes indicate low or no demand. This is also indicated in the colour bar. The temporal distribution is divided into 30 bins, 1 for each minute of the simulation.

Note, our definition of rewards adjustment covers only those techniques where a reward (no penalty) term exists. Figure 6.2 highlights the overall cost, service rate, time KPI's, mean loaded parcels, and total travel distance of all vehicles. The last bar (represented in red) in the graph highlights the results obtained in case of perfect anticipation. Perfect anticipation is a simulation scenario that represents the upper benchmark of performance improvement due to anticipation. The methodology is explained in detail in Appendix M. Detailed results are covered in Appendix P.

We observe that except for the adjustment technique where the value of the rewards is determined by the number of orders generated at the last node, all other techniques seem to improve the service rate by 1%-4%. Whether this is an improvement for an operator depends on the goals of the operator. For instance, in almost all simulations, the increase in the service rate comes at the cost of greater distances travelled by the vehicles. If the operator cost budget is limited, this increase in travel distance may be a great deterrent for adopting the anticipatory technique, even if it serves more customers.

One more thing to note is that the average loaded parcels on vehicles are lesser for all rewards adjustment techniques while the service rate and distance travelled is higher for a majority of the approaches. At first, it appears that for the majority of the approaches, the vehicles take shorted trips with lesser parcels and return to the depot several times in a route. This could potentially explain both the longer travel due to frequent depot returns as well as the lower occupancy of the vehicles. The service rate

improvement, however, can be the virtue of better assignments by rewards adjustment and cannot be explained by the shorter trips of the vehicles. The only exception to the above findings is the distance-based rewards adjustment- x closest depots from the last node technique that additionally reduces the total distance travelled. According to this technique, the vehicles are carrying lesser parcels on average, travelling lesser duration, but still improving the service rate over the myopic scenario. This could potentially be explained by better assignments due to rewards adjustment.

Another positive aspect from these results indicates that customer experience metrics are marginally better for 6 out of the 8 anticipatory techniques. Additionally, comparing the performance of our techniques with the perfect anticipation scenario, we observe that our approaches are able to achieve between 10% to 50% of the improvement anticipation can provide if perfect information about the future is known. Achieving such improvements with only a marginal increase in computation time indicates reasonable merit for further analysing rewards adjustment.

Amongst the explored rewards adjustment techniques, the technique that outperforms all others is the one where rewards values are computed as the average distance of x closest depots from the last node. Apart from having the highest service rate, it is the only technique that reduces the distance travelled by all the vehicles. As such, its performance comes closest to the perfect anticipation case. From now on, any rewards adjustment simulation refers to the distance-based adjustment: x closest depots from the last node.

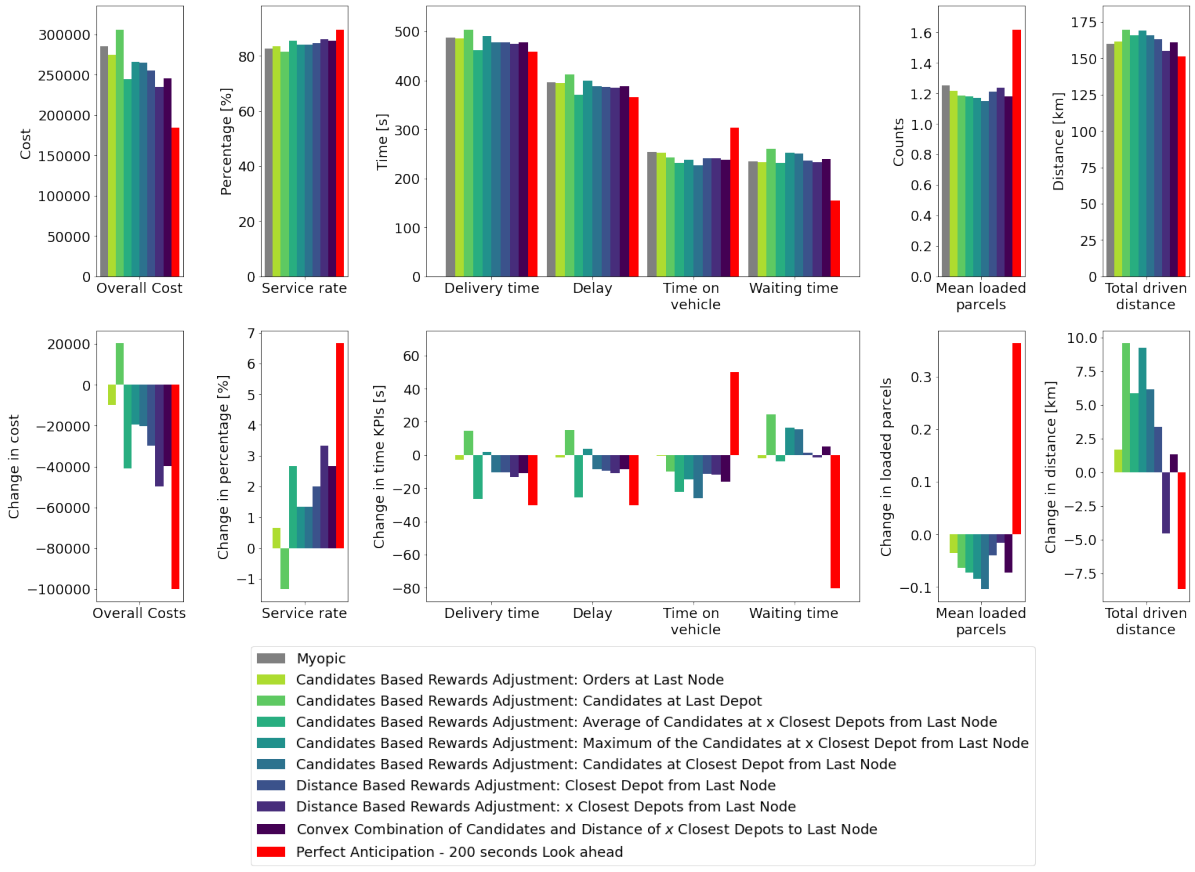
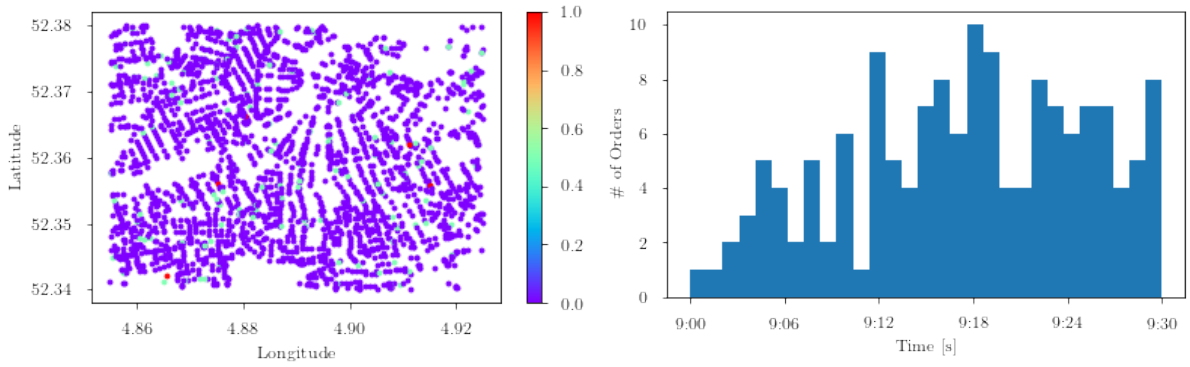


Figure 6.2: Performance Comparison of Several Adjustment Techniques using Rewards versus the Myopic Algorithm: Overall cost, service rate, time KPIs, mean loaded parcels, and total travel distance of all vehicles are displayed. The grey bar represents the performance of the Myopic simulation. Each other bar represents a unique rewards adjustment technique discussed in Chapter 4. Additionally, the Perfect Anticipation - 200s look ahead is represented by the red bar. Perfect Anticipation provides the maximum potential of anticipatory improvement over the VGA for the given data instance. The figure at the top represents absolute results whereas the bottom figure represents the difference in performance between the approach under consideration and the myopic simulation. Detailed results are covered in Appendix P.

6.1.2. Assessing Performance of Anticipation by Introducing Penalty + Rewards

In this section, we analyse the performance of several penalty/ penalty + reward adjustment techniques discussed in Section 4.3. These are compared with the results of the myopic algorithm, as well as the perfect anticipation case discussed in Appendix M. The results are obtained by simulating on a data sample instance indicated in Figure 6.3. The data set instance comprises a uniform spatial distribution and a series of temporal Gaussian distributions superimposed over each other. The total number of orders is 150 and the period of operation is 30 minutes.



(a) Spatial Distribution of Demand

(b) Temporal Distribution of Demand

Figure 6.3: Preliminary Instance for Testing Anticipation with Adjustments using Penalty/ Penalty + Rewards (P+R): The data instance comprises a uniform spatial distribution and a series of temporal Gaussian distributions superimposed over each other. The total number of orders is 150 and the period of operation is 30 minutes. The spatial graph is normalised such that red nodes indicate the highest demand and violet nodes indicate low or no demand. This is also indicated in the colour bar. The temporal distribution is divided into 30 bins, 1 for each minute of the simulation.

The key performance KPIs from the simulations are compared in Figure 6.4. These include the overall cost, service rate, time KPI's, mean loaded parcels, and total travel distance of all vehicles. We note from the figure that all anticipatory techniques with penalization outperform the myopic scenario with the overall service improvement ranging between 1%-3%. Furthermore, we note that techniques that include both a penalty and a reward term perform relatively better than ones with only a penalty term. This makes sense, as the former also incentivizes the vehicles to move towards high-demand regions while the latter only focuses on restricting vehicles in high demand regions.

Further, from the preliminary results, we observe that penalization with a constant term performs exactly the same as penalization with the travel time term. This, however, could have been a result of the nature of the distribution and not necessarily repeatable across simulations.

It is also interesting to note that unlike the rewards adjustment techniques mentioned previously, penalization based approaches load more orders on an average and reduce the total travelled distance. This seems reasonable, as by keeping the vehicles in high-demand zones we potentially limit any unnecessarily long trips that lead to low demand yield sectors. Additionally, since we are more likely to remain within small sectors of high demand, we potentially load more parcels for deliveries. The only exception to this is the zone-based adjustment technique. This can potentially be explained by the fact that the zones are defined differently under this technique. Unlike other anticipatory techniques where the zones are defined relative to the first node of the vehicle, the zone-based adjustment approach generates zones using an ILP solver. As a result, the zones generated are independent of the vehicle locations. The static generation of zones seems to not have the same impact as the dynamic zones in reducing the total travelled distance.

Comparing the performance of the proposed techniques with the perfect anticipation scenario, it seems that adjustment with penalty offers about 30% of the maximum improvement VGA can achieve with anticipation. Similarly, adjustment with penalty + rewards offers about 60%, and zone-based adjustment offers about 50% of the maximum improvement VGA can achieve with anticipation.

Amongst the explored penalization based techniques, the method where adjustment occurs due to both penalty and rewards stands out. Not only does it outperform in terms of service rate improvement, but

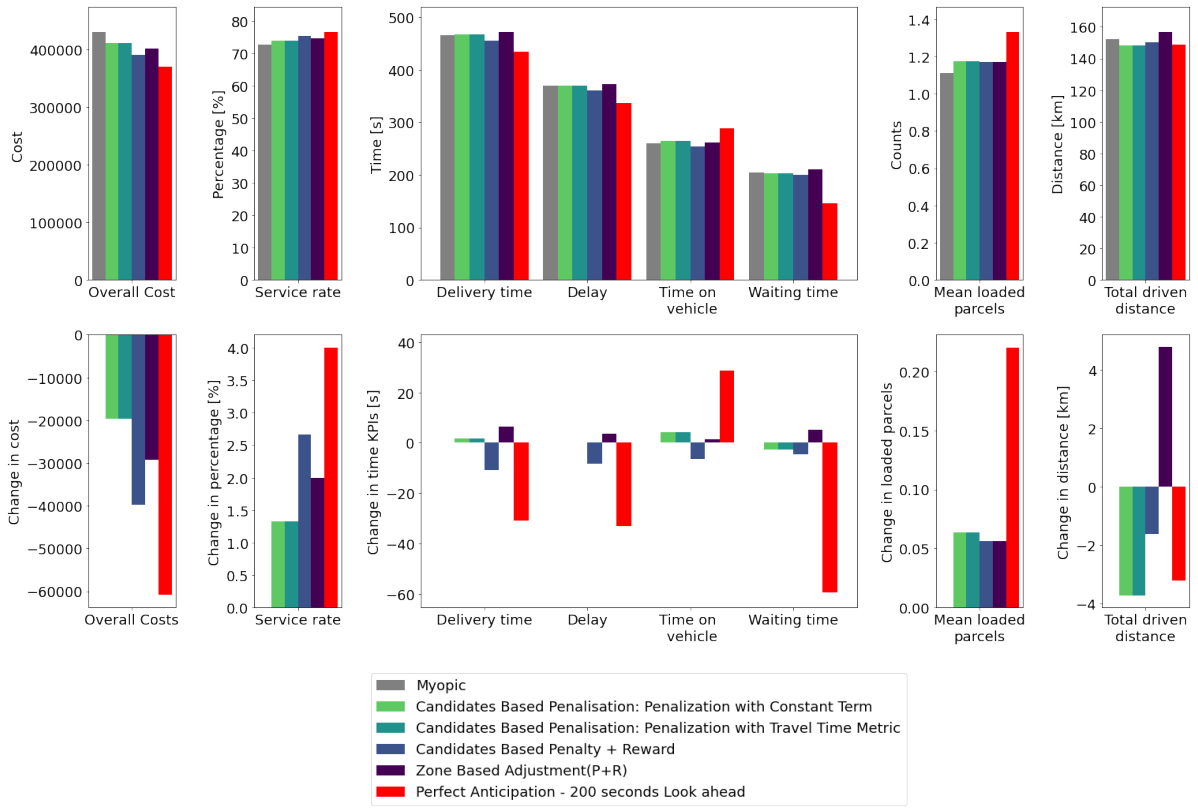


Figure 6.4: Performance Comparison of Several Adjustment Techniques using Penalty/Penalty & Reward, & Myopic Simulations: overall cost, service rate, time KPIs, mean loaded parcels, and total travel distance of all vehicles are displayed. The grey bar represents the myopic simulation results and each other bar represents a unique adjustment with P/P+R technique discussed in Chapter 4.Additionally, the Perfect Anticipation - 200s look ahead is represented by the red bar. Perfect Anticipation provides the maximum potential of anticipatory improvement over the VGA for the given data instance. The figure at the top represents absolute results whereas the bottom figure represents the difference in performance between the approach under consideration and the myopic simulation. Detailed results are covered in Appendix P.

it also marginally reduces the distance travelled as well as the delivery time, thereby offering an overall better customer experience. From now on, all further results for adjustment with penalty + reward refer to the technique of candidates based penalty + reward adjustment.

6.1.3. Discussion on Preliminary Results

While the preliminary results are promising and offer about 1-4% service rate improvement with both adjustment techniques, it is imperative to note that the simulations are only executed on independent sample data instances. Running rewards adjustment on the data instance used for penalty adjustment show marginal to no improvement over the myopic scenario. Similar findings are also observed in the case of penalty adjustment where performance improvement on the rewards adjustment data instance is mild. This stresses the need for a methodology that identifies a suitable anticipatory technique given the underlying demand distribution. Furthermore, the possibility of switching anticipatory techniques during operations could be explored.

Another important aspect to note here is that the underlying data instances used are fairly small. As a result, minor improvement in performance can have a relatively large impact on performance KPIs. The data instances however are deliberately chosen to be small so as to identify the most promising adjustment techniques without long computation cycles. Whether the selected anticipatory approaches offer performance enhancement in large data instances still remains to be seen.

6.2. Analysis over Realistic Demand Instances

The objective of this section is to identify whether the proposed anticipatory techniques can replicate their performance from Section 6.1 on near-realistic large-scale data instances. Further, the long period of operation helps us analyse the long term effects of our anticipation methodologies.

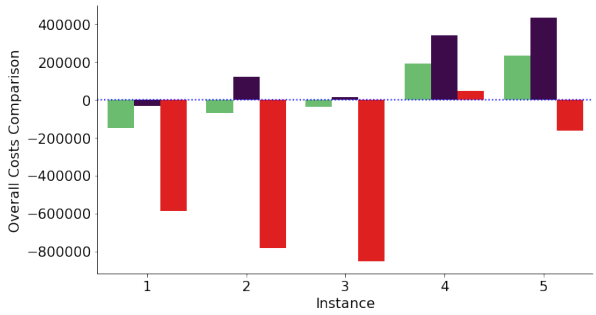
6.2.1. Comparative Performance

Figure 6.5 highlights the relative performance of anticipation by rewards adjustment as well as the adjustment with penalty + rewards with respect to the myopic algorithm for all 5 realistic data instances.

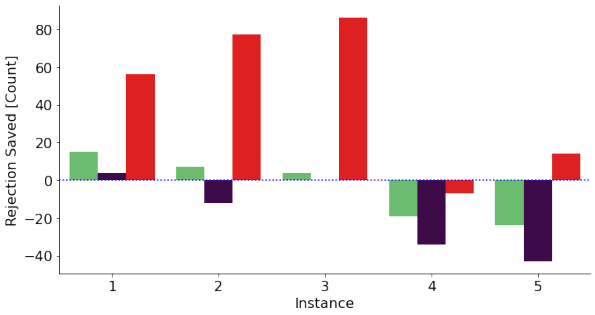
Looking at Figure 6.5b, we observe that rewards adjustment serves marginally more orders in 3 out of the 5 instances. This translates to a service rate improvement of 0.1-1% over the myopic simulation. It is important to note that resources such as total fleet size, number of depots, and other operational constraints are kept constant for all 5 demand instances and across each methodology- myopic, anticipatory approach, etc. As a result, the change in relative performance between the anticipatory technique and myopic instance is solely due to anticipation. As the demand in instances 1-5 increases from 1600-3600 orders, we observe that the potential of rewards adjustment diminishes with an increase in total demand and even degrades in comparison to the myopic simulation. In particular, for instance, 4 and 5, the orders served by rewards adjustment are lower than the myopic simulations.

In addition, Figure 6.5c highlights the change in average delivery time for the different anticipatory techniques with respect to the myopic simulation. It can be seen that the delivery time for both anticipatory techniques typically increases in comparison to the myopic simulation. A similar phenomenon can also be observed with distance travelled. This is illustrated in Figure 6.5d. These results indicate that the improvement observed amongst the preliminary results (Section 6.1) in terms of reduced delivery times and distance travelled seem to have diminished entirely for large scale instances.

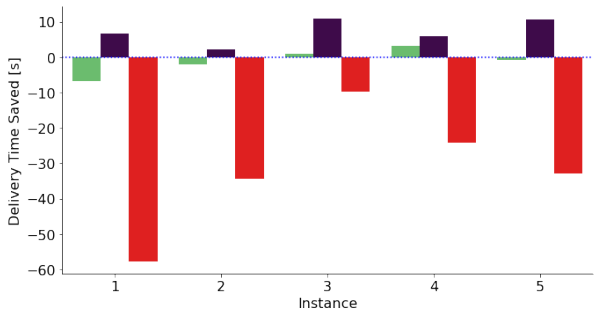
It is also important to note that the adjustment with penalty and rewards outperforms the overall service rate in only 1 out of the 5 instances. The instance for which this technique offers improvement is the one with the least demand. Additionally, resources such as total fleet size, number of depots, and other operational constraints are kept constant for all 5 demand instances and across solution methodology. The results indicate a sharp degradation in performance of adjustment with penalty + rewards as the overall demand increases while the other features are kept constant.



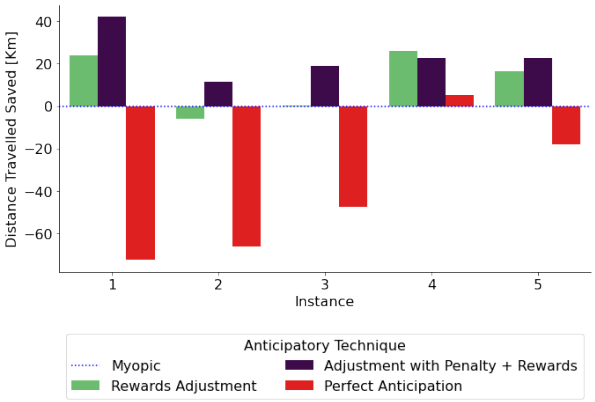
(a) Comparison of Overall Cost for individual realistic instances: A positive value of the bars indicates higher cost than the myopic simulation, and hence, poorer performance.



(b) Comparison of the number of Orders Served for individual realistic instances: A positive value indicates that the anticipatory methodology serves more orders than the myopic simulation.



(c) Comparison of the total delivery time for individual realistic instances: A positive delivery time for the anticipatory methodology indicates that customers on average receive their orders later than in the myopic simulation.



(d) Comparison of overall distance travelled for individual realistic instance: A higher distance means a greater amount of travel for all vehicles.

Figure 6.5: Performance Comparison of Rewards Adjustment, Adjustment with Penalty + Reward, & Perfect Anticipation with respect to the Myopic Algorithm for the realistic demand instances: Overall Cost Comparison, Rejections Saved, Delivery Time, & Distance travelled are the key performance KPIs that are compared between the considered anticipatory methodology (colour in legend) and the myopic simulation represented by the blue dotted line. Other KPIs are covered in a detailed table in Appendix P.

An interesting observation is that the upper benchmark of performance improvement i.e, perfect anticipation also has a threshold limit of total orders after which its performance diminishes. This can be observed in Figure 6.5a where the overall cost difference between perfect anticipation and myopic simulation diminishes as demand increases from instances 1-5, with perfect anticipation performing worse than the myopic simulation in instance 4. This, however, can not be attributable to the performance limits of the approach but is rather the result of limited computation time for trips of the CTV graph. To illustrate this, we simulate two perfect anticipation scenarios with trip computation cut off $\rho_{CTV,max}$ of 30s and 90s for instance 4. This is depicted in Figure 6.6. The 30s approach performs worse than the myopic simulation, but the 90s approach does better. This seems reasonable, as with more number of orders in the perfect anticipation horizon, a greater number of trips are possible. Hence, a greater trip computation time is required for exhaustively generating all possible trips. We, therefore, conclude that in the case of instance 4, the cut-off time of 30 seconds can not exhaustively search all vehicle-trip combinations. This leads to relatively poor performance. Increasing the cut-off time solves this issue but at the cost of exponentially longer computations.

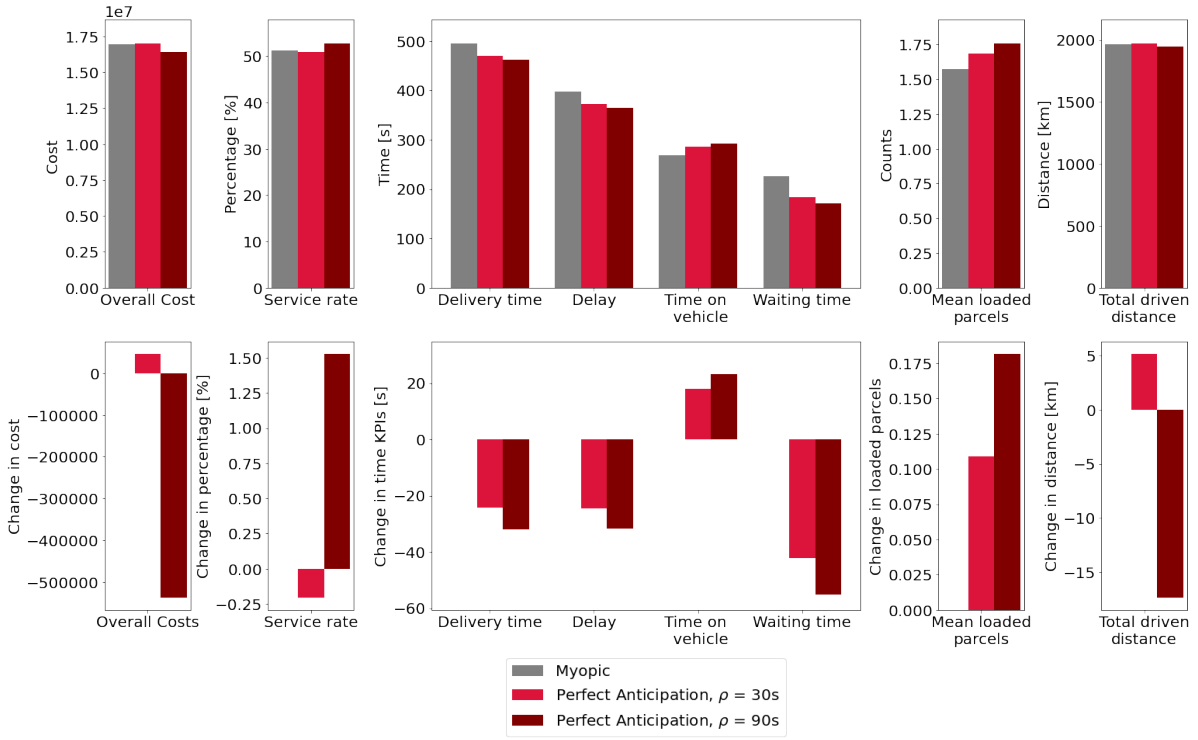


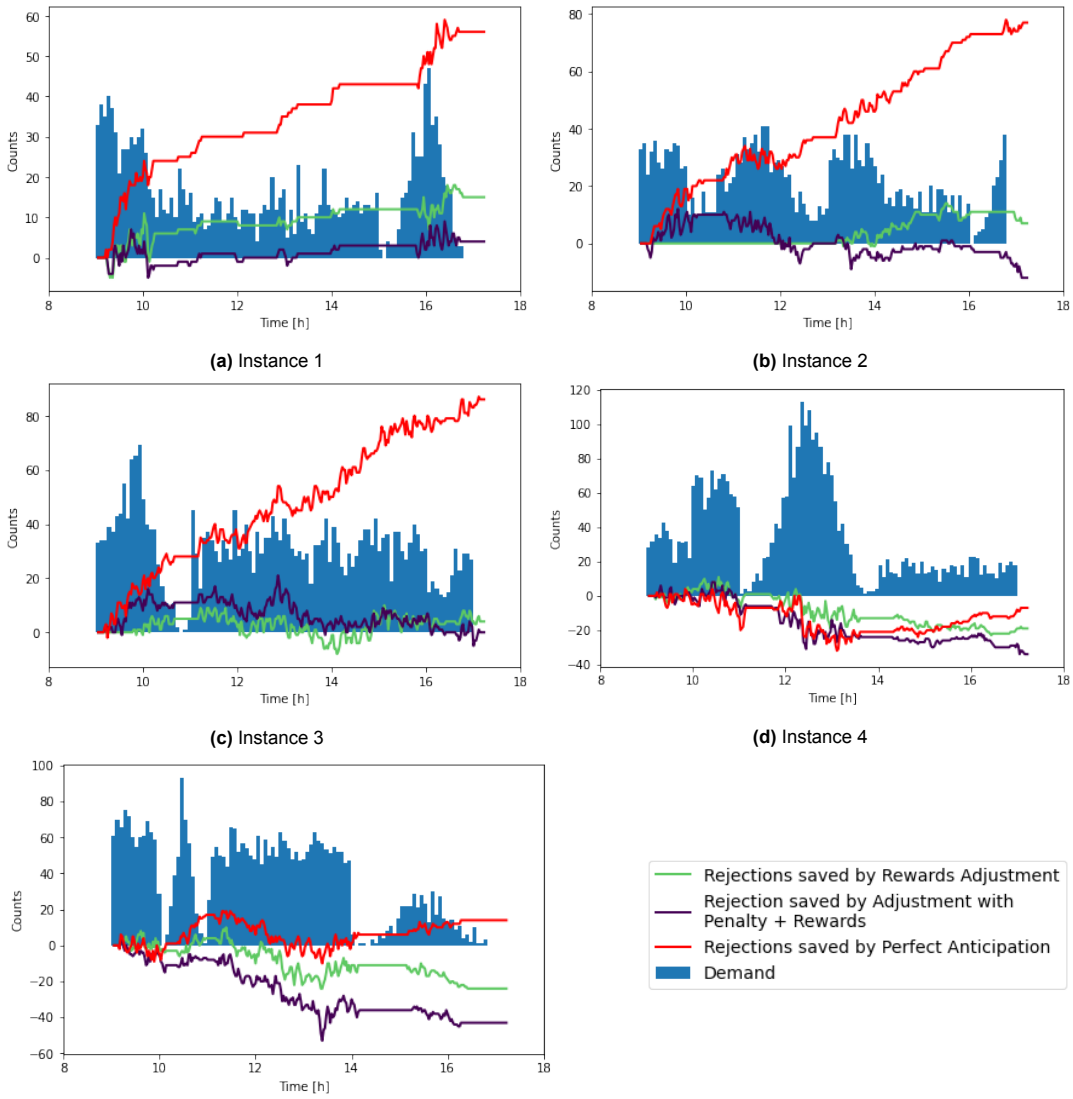
Figure 6.6: Performance comparison to illustrate perfect anticipation performance at different $\rho_{CTV,max}$ cut-offs. The figure displays Overall Cost, Service rate, time KPIs, mean loaded parcels and total driven distance for all vehicles. The bottom figure represents a change in KPIs with respect to the myopic simulation run.

6.2.2. Impact over Temporal Evolution of the System

In this section, we study how the number of rejections evolves in time, compared to the case of no anticipation. The main objective is to reason about the diminished performance of our approaches during a full day of operation. Figure 6.8b shows how many rejections are saved by introducing anticipation over VGA's algorithm for each of the 5 realistic data instances. This is done by depicting the difference between the accumulated rejections in the myopic scenario, and the accumulated rejections in all our anticipatory techniques. The figure also illustrates the demand over time to assess any variation in anticipatory potential with respect to the temporal distribution of orders.

In the case of rewards adjustment, results are hard to analyse. The inter-dependence of demand per time step and anticipatory technique is not apparent. This is possible because the rewards adjustment technique itself is not directly dependent on the demand but rather a function of the depot locations in relation to the demand. One indisputable finding remains that rewards adjustment shows limited benefit in a high rate of demand (approximately 50+ orders in 5 minutes) situations and continues to perform worse even when this rate of demand diminishes. This can clearly be observed in instance four of the Figure 6.8b. In the figure, the demand/time decreases to 20 orders/5 minutes around 14:00. At such demand rates, our anticipatory technique has already shown performance improvements(refer instances 1 and 2). Despite the reduction in the demand rate, the rewards adjustment technique completely fails at recovering from the initial loss of orders and continues to have a higher number of rejections than the myopic simulation.

Observing results from individual instances, we find that while adjustment using penalty + rewards shows performance improvement in only 1 of the 5 instances, it initially saves rejections in 3 out of the 5 cases with performance diminishing after a few hours of operations. The 2 instances where it does not seem to do well from the beginning of the simulation are cases where demand per time instance is substantially higher (instances 4 and 5) than other instances. We conclude that restricting vehicles in certain regions may result in poorer states of the vehicles in the long term. Perhaps, this explains why adjustment with penalty + rewards outperformed the myopic simulation in small data instances (Section 6.1) but failed to replicate those results in large-scale long term scenarios (Section 6.2). We



(a) Instance 1

(b) Instance 2

(c) Instance 3

(d) Instance 4

(e) Instance 5

Figure 6.7: The difference in accumulated rejections over time with and without anticipatory methods: The y-axis shows the cumulative difference between the number of accumulated rejections if using the no-anticipatory method at all, and introducing rewards (green curve), penalty + rewards (purple curve) or perfect anticipation (red curve). Demand is represented in blue to evaluate variation in anticipatory performance with respect to variation in demand. Instances 1-5 increase in demand from 1600-3600 orders. Each vertical bar of demand represents the number of orders in approximately 5 minutes of operation.

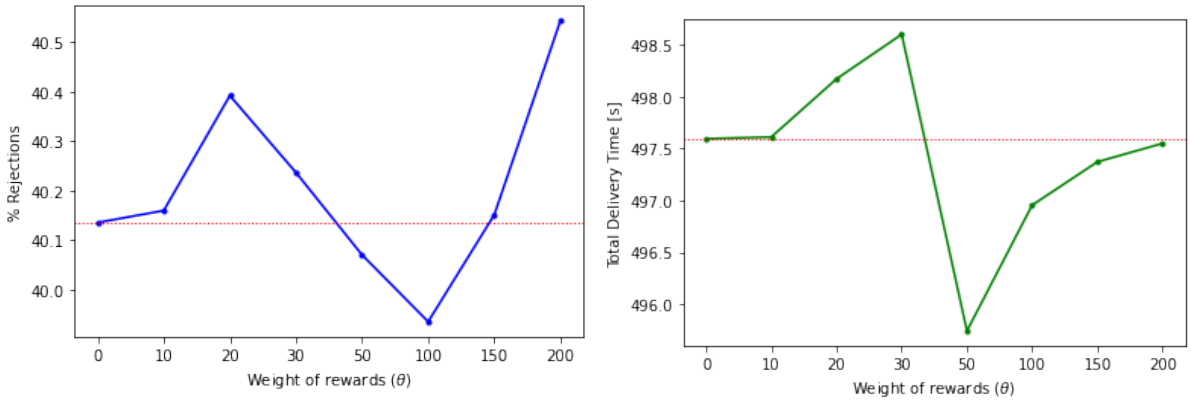
also find that anticipation shows limited benefit in cases where the demand per time step is extremely high - approximately 40+ orders every 5 minutes- and continues to perform worse even when this rate of demand reduces. Therefore, previous vehicle states may have a strong compounding effect on the future performance of anticipatory methodologies.

Further, we also note that in all simulations, first, the anticipatory techniques perform slightly worse than the myopic simulation before improving the number of rejections. This is in line with the principle of anticipation, where decisions are modified to be better prepared for the future.

6.2.3. Assignment Including Rewards: Sensitivity to Change in Tuning Parameter Θ

We study the effect rewards weight Θ from the rewards adjustment anticipatory technique has on our simulations. We do this by systematically varying the tuning parameter Θ while keeping all other parameters unchanged. Figure 6.8 shows the resulting effect of the %rejections and total delivery time as Θ is varied. The results in the graph represent an average performance value for all 5 realistic instances.

We note from the figure that both the number of rejections and the total delivery time first increases with an increase in Θ. This indicates that a mild rewards adjustment that only impacts the system in a limited manner ends up diminishing performance. As Θ is further increased, the anticipatory impact becomes more apparent, peaking at a Θ value between 50-100. As this range of Θ where performance improvement is wide, one can expect a performance improvement even if an optimal Θ is not achieved. Additionally, any further increase in Θ after 100 drastically worsens performance. This is because a higher Θ leads to an aggressive adjustment where vehicles compromise on serving current orders and select trips closer to depots. Building from the findings of Fielbaum et al.[15], we conclude that identifying an optimal tuning parameter is a crucial issue for our anticipatory techniques.



(a) Average rejected requests percentage as a function of the weight of the rewards Θ

(b) Average delivery time as a function of the weight of the rewards Θ

Figure 6.8: Sensitivity to tuning parameter Θ: % rejections and the total delivery time as a function of Θ are displayed. The baseline results, i.e, myopic simulation results are shown by a horizontal red line. The results represent averages of 5 simulations where each simulation is executed on a unique realistic data instance. Other KPIs are illustrated in Appendix N.

A similar analysis was conducted for the technique involving adjustment with penalty + rewards. It was found that the best performance across all realistic instances was achieved when both κ and χ were equal to 1. Any deviation from this value resulted in gross degradation in performance.

6.2.4. Discussion on Analysis over Realistic Demand Instances:

Analysing our anticipatory techniques over large scale instances, it is clear that the impact of anticipation reduces as the demand increases. However, it is important to note that while the demand increased from instances 1-5, the number of vehicles, depots, and time constraints remained the same for all instances. As a result, each vehicle had a higher workload that increased from instances 1-5. It is likely that the anticipatory potential reduces under extremely high workload scenarios and eventually becomes counter-intuitive to the routing algorithm (as observed in instances 4 and 5). Increasing the

number of vehicles proportionally to the demand may resolve this diminished anticipatory potential and indicate a similar improvement in performance as observed in Section 6.1 over a myopic algorithm with the same proportional increase in vehicles.

6.3. Analysis across 80 Demand Instances

The results obtained in Section 6.1 and Section 6.2 indicate that the performance of an anticipatory technique is largely dependent on the underlying demand distribution. It, therefore, becomes imperative to understand how the impact of anticipation varies with different demand scenarios. In this section, we will explore the performance of our techniques over 80 standardized high demand scenarios introduced in Section 5.3, and evaluate spatial-temporal characteristics that favour anticipation.

6.3.1. Overall Performance

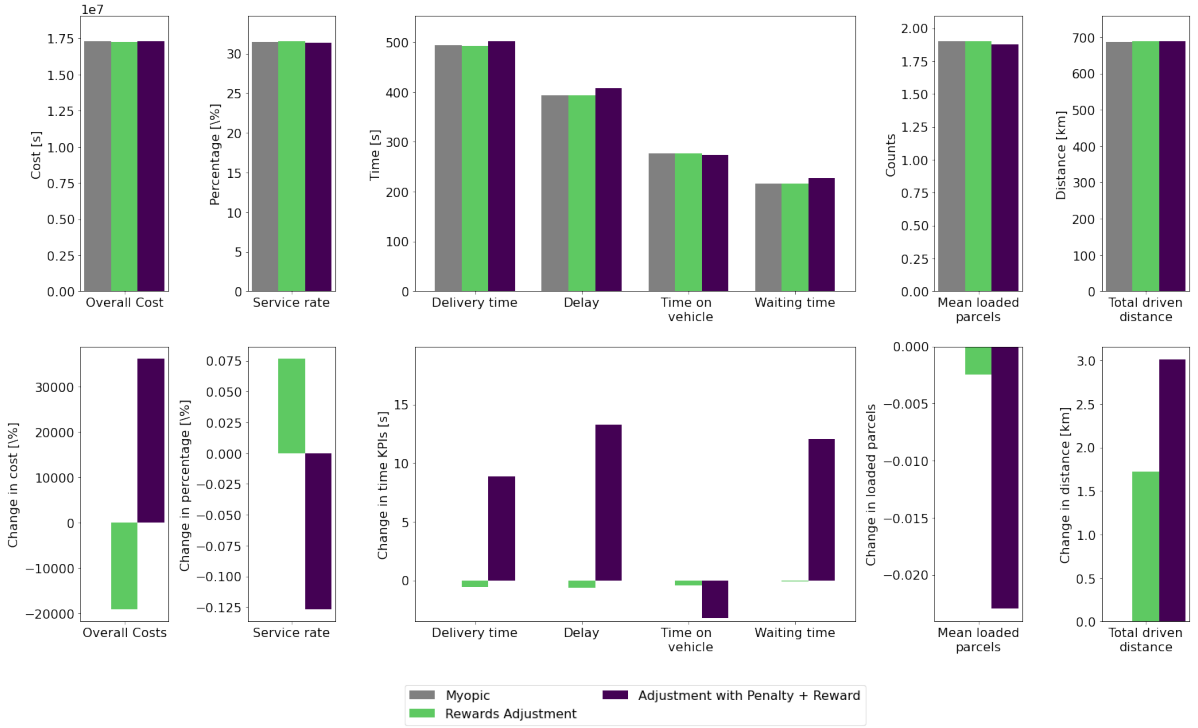


Figure 6.9: Overall Performance Comparison of Rewards Adjustment and Adjustment with Penalty + Reward with respect to Myopic Algorithm for all standardized demand instances: Overall cost, service rate, time KPIs, mean loaded parcels, & total travel distance of all vehicles are displayed. The result is aggregated over 80 demand instances introduced in Section 5.3.

Figure 6.9 highlights the performance of the myopic, rewards adjustment, and adjustment with penalty + rewards approaches for all 80 data sets combined. To eliminate any influence of outliers in individual results, a median of each selected KPI is computed for all instances. From the figure, it appears that rewards adjustment marginally improves performance in a majority of the data set instances. This is true for all performance KPIs except the distance travelled. However, the impact of an increase in distance is minor in comparison to the increase in service rate. This can also be observed by the lower overall costs compared to the myopic simulation. These results suggest that the rewards adjustment technique proves to be robust at improving performance without any additional computation budget requirement. Although, it is important to note that the degree of improvement is marginal and only up to a maximum demand per unit time value.

On the other hand, adjustment with penalty and rewards diminishes performance in a majority of the data set instances. This relatively poor performance of anticipation with penalisation and rewards is, however, not unexpected. One possible suggestion could be that since the duration of operations is 3 hours, the long-term impact of restricting vehicles in certain zones have taken effect and diminished

overall results. Other reasons could be associated with the demand distribution itself. These are discussed in more detail in subsequent sections.

6.3.2. Performance Comparison across 80 Demand Instances

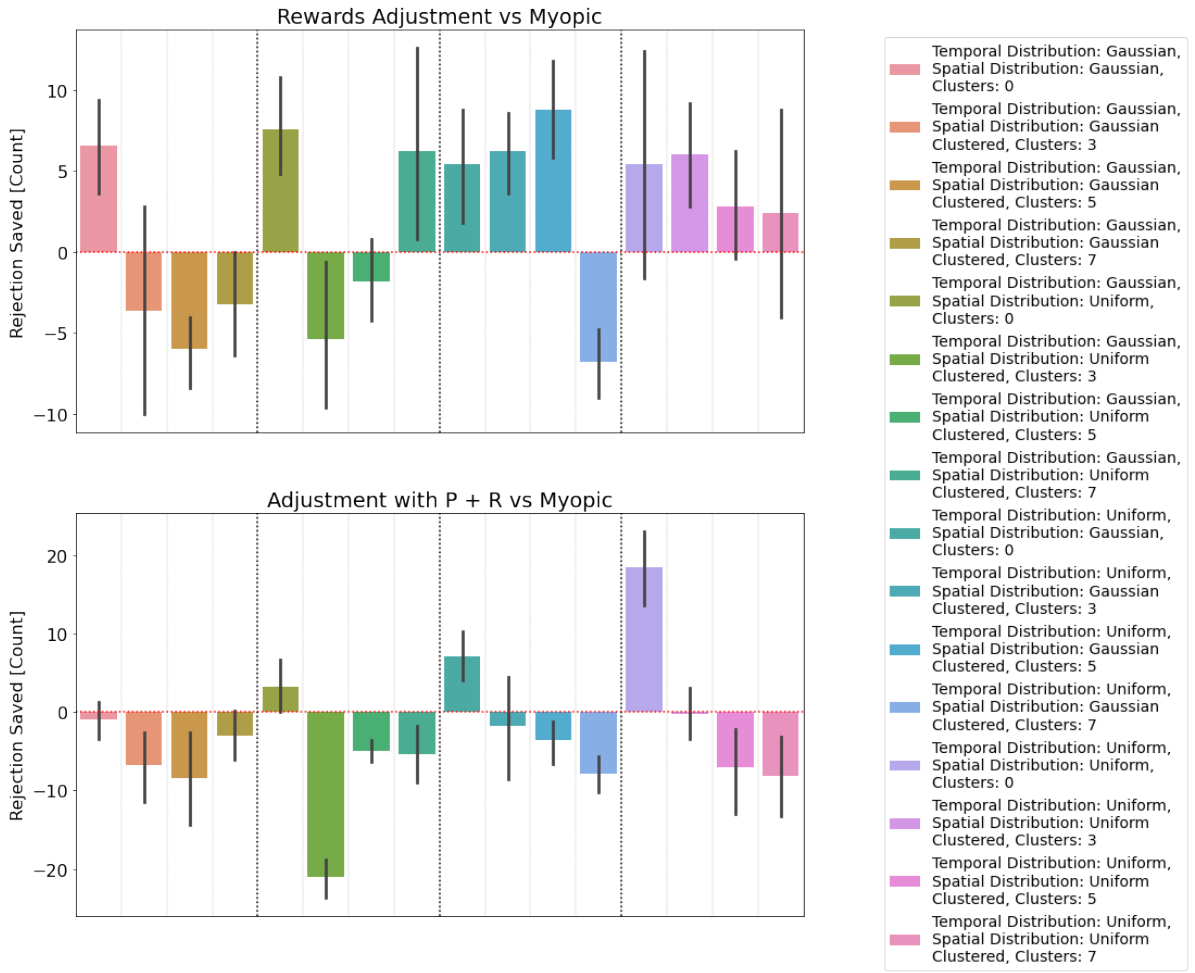


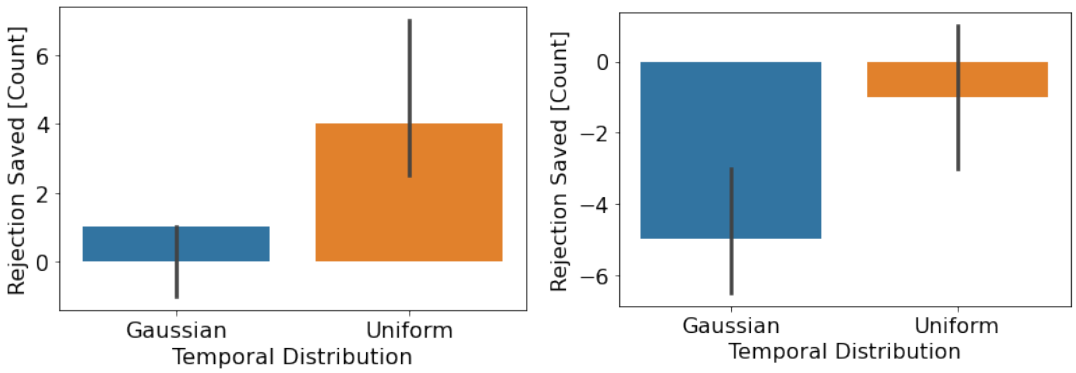
Figure 6.10: Median difference in accumulated rejections with and without anticipatory methods: The y-axis represents the median difference in rejection with and without anticipatory techniques, and the x-axis represents independent demand distributions (highlighted in the legend). Results are aggregated over 5 instances for each distribution. A positive value of a bar indicates that the median number of orders served by the anticipatory methodology is greater than the myopic method for particular data distribution. The black lines over the bars represent the standard error over the median rejections saved.

In this section, we study the difference in accumulated rejections between our proposed anticipatory techniques and the myopic simulation over each data set type described in Section 5.3.

Figure 6.10 highlights the median difference in rejections between each of the proposed anticipatory techniques with respect to the myopic simulations. The results are aggregated over 5 instances for each unique distribution. Further, a confidence interval of 65% was chosen to evaluate the standard error around the median value. From the figure, we observe that the rewards adjustment technique serves more orders in 10 out of 16 data set types. On the other hand, adjustment with penalty and rewards outperforms the myopic simulations only in 3 out of the 16 data set types. Further looking into the results, we observe that rewards adjustment tend to do better under certain distribution types over others. For instance, rewards adjustments outperform the myopic simulation in all scenarios when both the spatial and temporal distributions are Uniform. On the other hand, rewards adjustment outperforms the myopic simulation in only 1 out of the 4 instances when both the spatial and temporal distributions are Gaussian. As such, the results from the figure strengthen our hypothesis that there exists an inter-dependence between the impact of anticipation and the underlying demand distribution. Apart from

rejections saved, differences in other KPIs are also studied and can be found in Appendix O.

Impact of Temporal Distribution



(a) Rewards adjustment vs Myopic

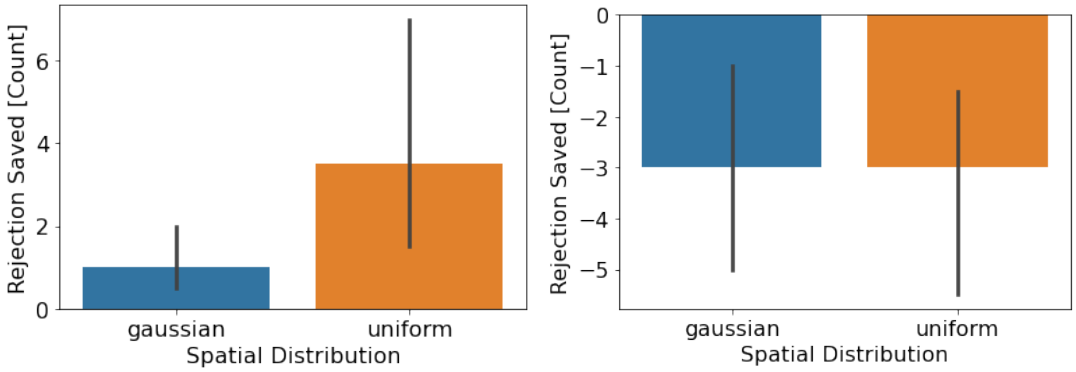
(b) Adjustment with penalty + rewards vs Myopic

Figure 6.11: Median difference in accumulated rejections with and without anticipation for independent temporal distribution: Both figures represent a comparison of rejections between a unique anticipatory technique and its myopic counterpart. The two bars in each figure correspond to results obtained on different temporal demand patterns discussed in Section 5.1. As such, a positive value of the bar represents additional orders served when deploying an anticipatory methodology in place of the myopic method. The results are aggregated over 40 instances for each of the temporal distributions.

Figure 6.11 illustrates the median rejections accumulated with and without anticipatory techniques for each temporal data distribution. As such, a positive value of the bar represents additional orders served when deploying an anticipatory methodology in place of the myopic method. The results are aggregated over 40 instances for each of the temporal distributions.

The figure highlights that both anticipatory techniques are capable of serving more orders in comparison to their myopic counterpart in the case of Uniform temporal distributions. As Gaussian distribution aggregate the majority of the demand within a small operation period, the solution algorithm has more trips and routes to work on within that period. This large rate of demand limits the anticipatory benefit provided by the proposed techniques. This also aligns with our findings in Section 6.2 that anticipatory techniques tend to diminish in performance as the number of orders per time instance increases.

Impact of Spatial Distribution



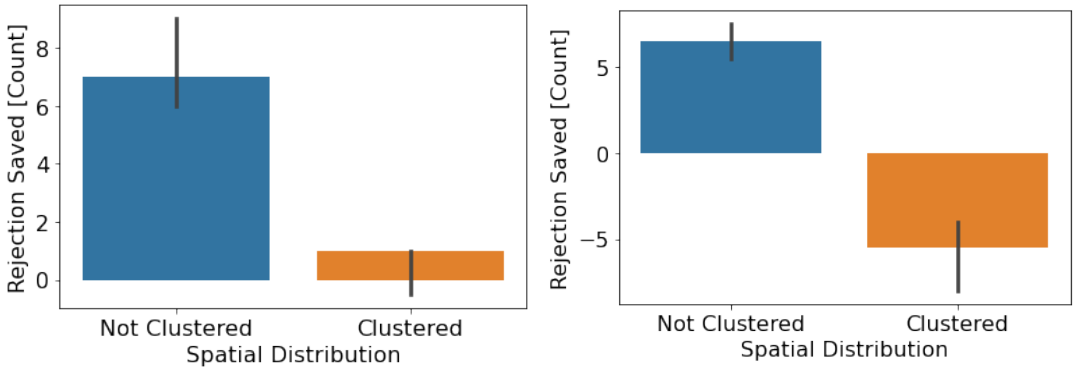
(a) Rewards adjustment vs Myopic

(b) Adjustment with penalty + rewards vs Myopic

Figure 6.12: Median difference in accumulated rejections for Spatial Distribution Types- Nature of Distribution: Each figure represents a comparison in rejections accumulated between an anticipatory technique and its myopic counterpart. Independent bars in each figure correspond to one of the spatial distributions discussed in Section 5.2.2. As such, a positive value of the bar represents additional orders served when deploying an anticipatory methodology in place of the myopic method. The results are aggregated over 40 instances for each of the temporal distributions.

Figure 6.12 highlights the impact that the nature of a spatial distribution has on the proposed anticipatory techniques. It does so by illustrating the median rejections accumulated with and without anticipatory techniques for each spatial data distribution. As such, a positive value of the bar represents additional orders served when deploying an anticipatory methodology in place of the myopic method. The results are aggregated over 40 instances for each of the spatial distributions - Uniform & Gaussian.

Looking at the figure, the rewards adjustment approach tends to perform relatively better when the distribution is Uniform rather than Gaussian. On the other hand, it appears from the figure that adjustment with penalty + rewards has no dependence on the spatial distribution pattern of demand. Surprisingly, one would expect adjustments with penalties + rewards to serve more orders in the case of Gaussian distributions. This is because the approach restricts vehicles to certain regions where demand is high. As a result, such an approach should keep vehicles at the centre of the Gaussian demand clusters thereby serving more orders. This, however, does not seem to be the case.



(a) Rewards adjustment vs Myopic

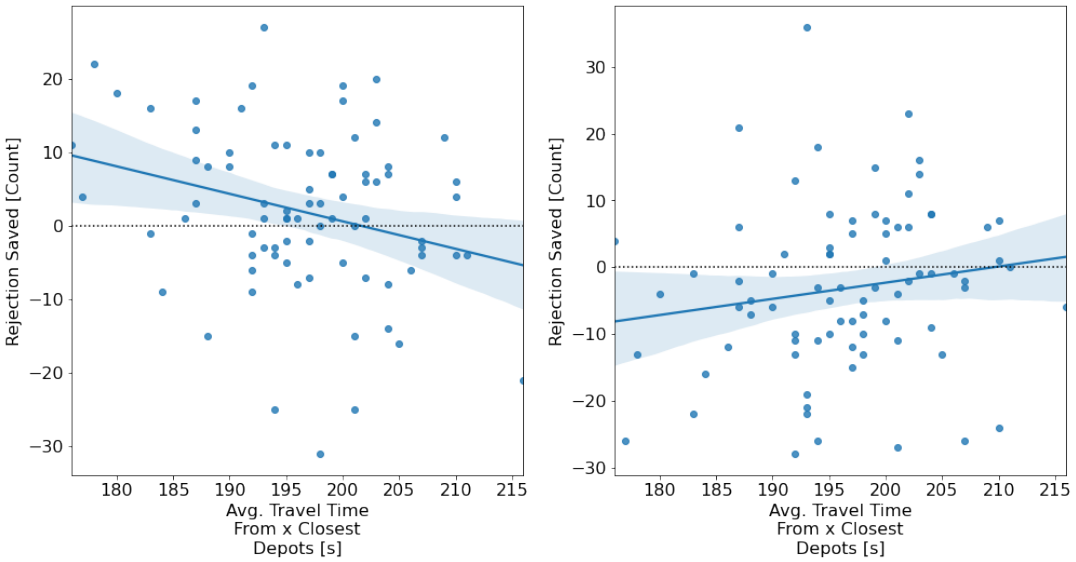
(b) Adjustment with penalty + rewards vs Myopic

Figure 6.13: Median difference in accumulated rejections for Spatial Distribution- Clustered vs Non-clustered: Each figure represents a comparison in rejections accumulated between an anticipatory technique and its myopic counterpart. Independent bars in each figure correspond to results obtained on different demand types as per Section 5.2.1. As such, a positive value of the bar represents additional orders served when deploying an anticipatory methodology in place of the myopic method.

Apart from the different spatial distributions, demand locations are also divided into clustered or non clustered locations as per Section 5.2.1. Figure 6.13 highlights the difference in accumulated rejections with and without anticipatory techniques for clustered and not clustered demand. The figure highlights that both anticipatory techniques perform substantially better in scenarios where the demand is not clustered.

Perhaps, the decrease in performance of rewards adjustment when demand is clustered could be explained by the relative distribution between the depots and sub-graphs where demand is generated. To clarify, we know that the rewards adjustment approach tends to select trips that end closer to depots. Now, in the case where the sub-regions of demand are substantially farther away from depot locations, the rewards adjustments technique will influence vehicles to locations of relatively lower demand that are closer to the depots. As a result, in the case of clustered demand, where clusters themselves are located farther from depots, the anticipatory technique may prove to be counter-intuitive and limit the benefit of anticipation. A potential solution in this direction could be incorporating a rewards value term that uses the endogenous demand of the graph. Such a rewards value term may limit the counter-intuitive trip selection of the proposed approach by keeping in view the location of the demand.

The results obtained for adjustment with penalty + rewards are even more surprising. Since this anticipatory approach restricts vehicles in certain locations, one would expect it to outperform the myopic simulations when the demand is already clustered in certain regions. However, this does not seem to be the case. From these results, it can be argued that a combination of multiple spatial & temporal characteristics influences the anticipatory potential of the discussed approach. We will further explore this relationship between depots and demand in further discussions to validate our hypothesis.



(a) Rewards adjustment versus Myopic

(b) Adjustment with penalty + rewards versus Myopic

Figure 6.14: Variation in rejections accumulated (anticipatory potential) with and without anticipatory techniques for an increasing average travel time between demand and depots: Each point on the scatter plot corresponds to a pair of simulations executed on a unique data instance from the 80 data instances generated in Section 5.3. The y-axis represents the rejections saved by deploying an anticipatory methodology in place of the myopic simulation. The x-axis represents the overall average travel time between all order destinations and their respective x closest depots. The blue line represents the regression line for the scatter points and the shaded area represents 95% confidence interval.

Impact of Proximity to Depots

In this section, we studied the impact of relative distribution between depots and demand on the anticipatory potential of our approaches. Figure 6.14 illustrates the difference in rejections accumulated with and without anticipatory techniques versus the travel time between the depots and demand. Each point on the scatter plot corresponds to a pair of simulations executed on a unique data instance from the 80 data instances generated in Section 5.3. The y-axis represents the rejections saved by deploying an anticipatory methodology in place of the myopic simulation. The x-axis represents the overall average travel time between all order destinations and their respective x closest depots. The blue line represents the regression line for the scatter points and the shaded area represents 95% confidence interval.

Following the regression line from the figure, we observe that rejections saved by using rewards adjustment technique marginally reduces as the average travel time between the depots and demand increases. After around 200 seconds of average travel time between depots and demand, the anticipatory benefit becomes completely void and serves even lesser orders in comparison to the myopic simulations.

On the other hand, adjustment with penalty + rewards serves lesser orders than its myopic counterpart when the average travel time between depots and demand is low. As the average travel time increases, the gap in the orders served between the adjustment with penalty + rewards and its myopic counterpart reduces, becoming greater than zero after 210 seconds. This indicates that adjustment with penalty + rewards has a higher anticipatory potential when the demand is located farther away from depots. Since the goal of the adjustment with penalty + rewards anticipatory technique is to keep vehicles in high demand zones, an increase in distance between demand and depots will cause the anticipatory method to focus on longer trips with more parcels per vehicle that limit the number of returns to depots. Thus, these results are aligned with our understanding of the approach. The interesting question here is that why does this anticipatory potential not translate to a similar performance when demand is clustered in specific regions as covered in the previous section? Additional analysis on the relationship between adjustment with penalty + rewards needs to be explored.

It is important to note that an increase in anticipatory potential does not indicate more orders served

over the myopic simulation. Rather, it focuses on the relative difference in orders served between the anticipatory technique and its myopic counterpart. An increase in anticipatory potential therefore, highlights a positive impact on the anticipatory approach. This could be either reducing the gap of orders served between the anticipatory technique and its myopic counterpart when the myopic simulation outperformed the anticipatory technique, or increase in rejections saved by the anticipatory technique. As a result, when we refer to increase in anticipatory potential, it is irrespective of whether the anticipatory techniques outperforms the myopic simulation.

The results from Figure 6.14 provide directional evidence about the impact of average travel time on anticipation. At the same time, it can be observed that the independent simulation results are widely scattered to be able to draw any strong conclusive insights. Hypothesizing that the spatial-temporal nature of demand also impacts this relationship, we further split the Figure 6.14 for each unique demand distribution in Section 5.3. This is indicated in Figure 6.15 for rewards adjustment vs myopic simulations and Figure 6.16 for adjustment with penalty + rewards versus myopic simulations. Just like Figure 6.14, each chart in Figures 6.15 and 6.16 illustrates the difference in rejections accumulated with and without anticipatory techniques versus the travel time between the depots and demand. Each chart further corresponds to a unique demand distribution from Section 5.3.

Independent charts for both anticipatory techniques provide a relatively clearer relationship between the average travel times from depots to demand and their impact on the anticipatory potential. As such, for rewards adjustment, 13 out of the 16 distribution setups indicate that increasing the average distance between depots and demand limits the impact of anticipation for rewards adjustment, i.e, the rejections saved by rewards adjustment reduces with increase in average travel time. Furthermore, adjustment with penalty and rewards highlights a much higher anticipatory impact in 11 out of the 16 distribution setups with an increase in the travel time.

6.3.3. Discussion on Analysis across 80 Demand Instances

Analysing the performance of both anticipatory techniques over the different demand instances provides several interesting insights.

First, we identify that rewards adjustment seems more robust than adjustment with penalty+rewards in improving performance over the myopic solution. Despite this, adjustment with penalty+rewards notes the largest number of median rejections saved over the myopic scenario for all data instances. This is observed when the demand is Uniformly distributed in space and time and has no clustered orientation. This indicates that adjustment with penalty + rewards offers a unique opportunity to be deployed whenever the underlying demand is Uniformly distributed.

Furthermore, both forms of anticipation show limited improvement in spatially clustered demand patterns. As a result, one can maximize performance enhancement by applying anticipation only when the demand is not clustered and turning anticipation off when clustering of demand is observed. This would allow the system to benefit from the advantages of anticipation whenever possible.

In addition, we note that the performance of rewards adjustment decreases as the distance to the depot increases. In contrast, the performance of adjustment with penalty + rewards increases with distance. An anticipatory technique that deploys rewards adjustment when demand is close to depots and adjustment with penalty + rewards when demand moves further away from depots could allow an improvement over both independent anticipatory techniques.

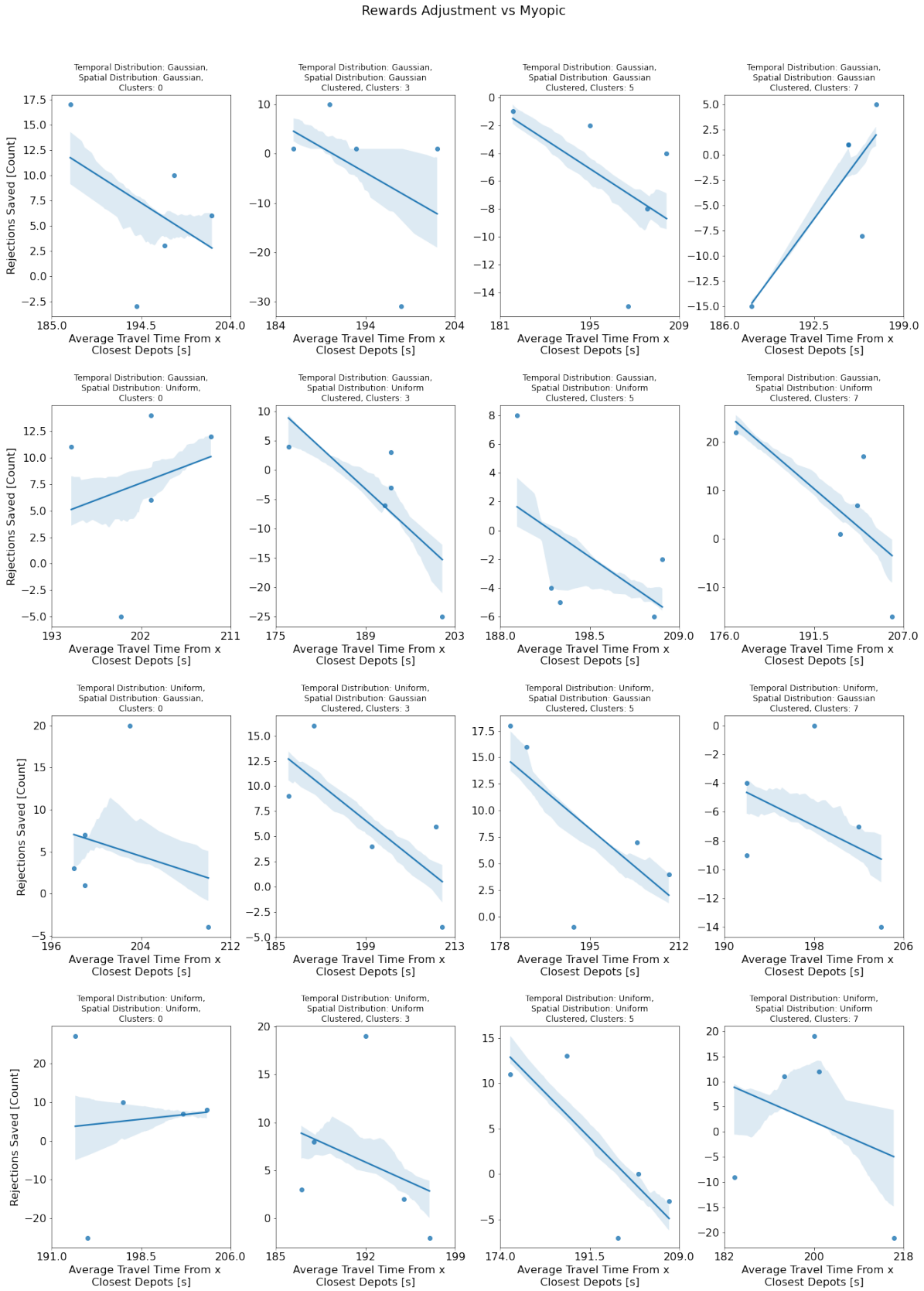


Figure 6.15: Variation in anticipatory potential (difference in rejections accumulated) of rewards adjustment when increasing average distance between travel time between depots and x closest depots for each demand distribution: Each point on the scatter plot corresponds to a pair of simulations executed on a unique data instance from the 5 data instances of a particular demand distribution. The y-axis represents the rejections saved by deploying the rewards adjustment methodology in place of the myopic simulation. The x-axis represents the overall average travel time between all order destinations and their respective x closest depots. The blue line represents the regression line for the scatter points and the shaded area represents a 65% confidence interval.

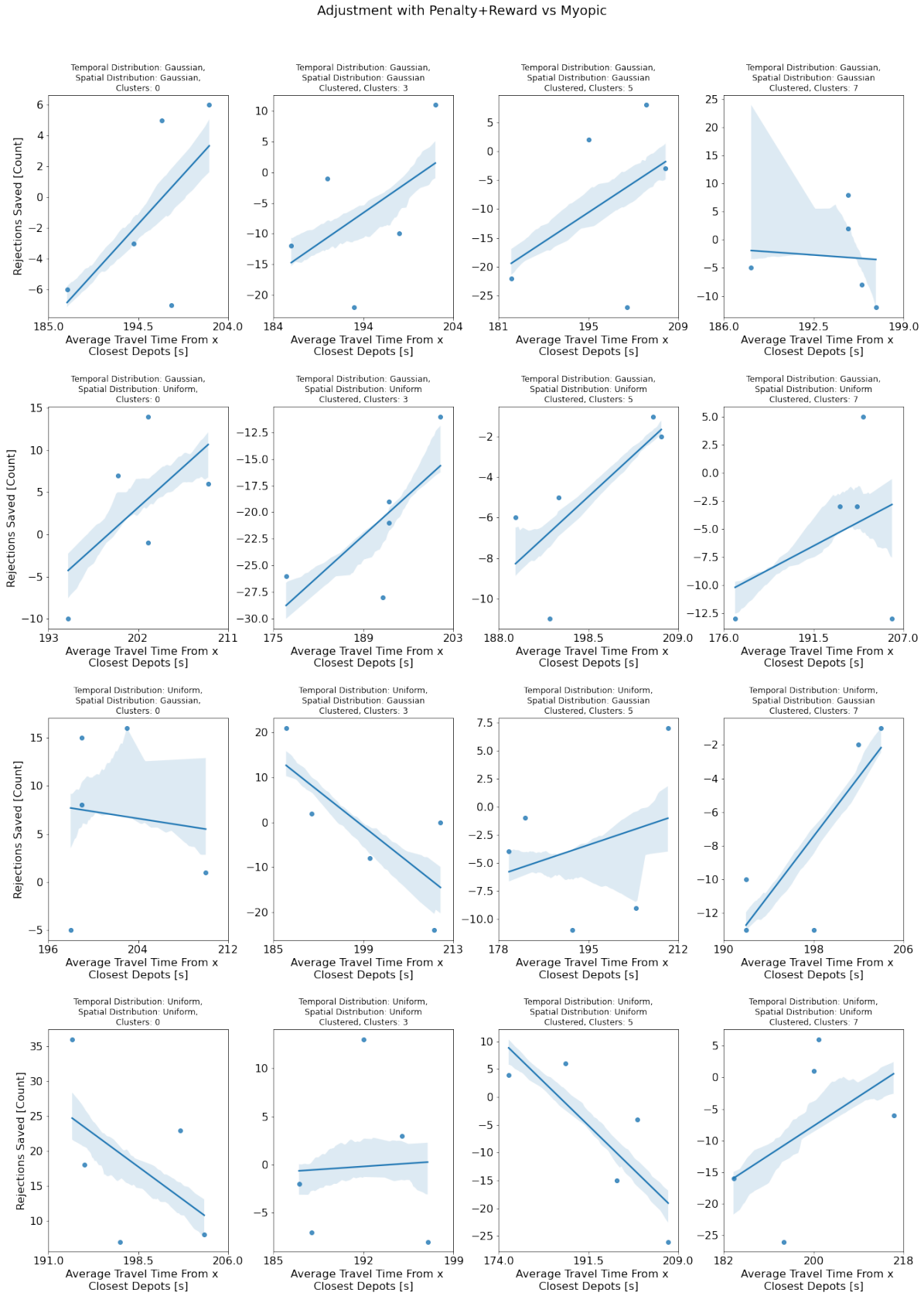


Figure 6.16: Variation in anticipatory potential (difference in rejections accumulated) of adjustment with penalty and rewards when increasing average travel time between depots and x closest depots for each demand distribution: Each point on the scatter plot corresponds to a pair of simulations executed on a unique data instance from the 5 data instances of particular demand distribution. The y-axis represents the rejections saved by deploying the adjustment with penalty + rewards methodology in place of the myopic simulation. The x-axis represents the overall average travel time between all order destinations and their respective x closest depots. The blue line represents the regression line for the scatter points and the shaded area represents a 65% confidence interval.

7

Conclusion

The objective of this study was to enhance the state-of-the-art solution for large-scale on-demand same-day grocery deliveries. We focus on devising several anticipatory techniques that can be implemented over the state-of-the-art VGA algorithm and improve the overall performance.

We propose a total of 12 modifications to existing anticipatory methods to suit the context of same-day delivery. For the first 8 methods, the objective function is modified by introducing a reward to all possible combinations of routes a vehicle can take. The other 4 methods focus on introducing an additional penalty term to prevent the vehicles from taking certain routes.

We observe that for small problem instances of 150 orders and 30 minutes of operation, eleven out of the twelve anticipatory methods serve up to 5% more orders and marginally reduce delivery times in comparison to the myopic method. This comes at a cost of greater distance traveled for a majority of the methods.

Additionally, the quality of anticipatory improvement decreases as the demand increases to real-world settings while keeping other resources constant. In particular, for large-scale long-duration problem instances, the rewards adjustment anticipatory technique improves service rate over the myopic methodology by only 0.1-1% of total demand. In certain instances of very high demand (3400+ orders in 3 hours), the rewards adjustment anticipatory techniques is even outperformed by the myopic methodology. On the other hand, adjustment with penalty + rewards is outperformed by the myopic algorithm in 4 out of the 5 real-world instances. For both anticipatory techniques, a threshold demand rate (approximately 40+ orders/ 5 minutes for adjustment with penalty + rewards and 50+ orders/ 5 minutes for rewards adjustment) is found. Any demand scenario with a higher demand rate generally leads to poorer performance than myopic methodology when deploying anticipatory techniques. While discussing this limitation, it is important to consider that operational resources such as fleet size, number of depots, etc, remain constant and do not scale with the rising demand. As a result, the diminished anticipatory potential could be attributed to the over-saturation of the existing fleet. An interesting question here would be whether increasing the number of vehicles or the number of depots helps increase the impact of anticipation for large problem sizes?

On further exploration, we identify that the performance of our methods is highly dependent on the underlying demand. To explore this phenomenon further, we devise strategies to generate several different types of demand distributions varying in their spatial-temporal characteristics and relationship with the graph environment. Our analysis indicates that our methods perform much better when demand per time instance is moderate and is distributed uniformly in time and space. Further, depending on the anticipatory technique, the average distance between demand and their feasible depots also plays an important role in influencing the overall performance of our methods. Despite their limitations, we identify that the proposed technique for rewards adjustment: adjustment based on distance from x closest depots to the last node of the trip, outperforms the myopic methodology on a majority of the instances. The rate of improvement is, however, marginal and ranges from 0.05-1.2% of the total demand.

We believe that our study opens up avenues for multiple directions of future research:

First, more robust anticipatory techniques could be explored. For instance, in the rewards adjustment case, more advanced methods depending on endogenous or exogenous demand could be utilized to compute the value of the reward. Further, for adjustment with penalty and rewards, a time metric of demand could also be added to their adjustment term to limit their diminishing impact over a long duration of operation. In addition, entirely new anticipatory techniques such as the artificial requests method with VGA could also be explored.

Second, anticipation has shown to be sensitive to underlying demand and graph characteristics. Therefore, further analysis on how these characteristics impact performance will be of considerable value. This insight will allow us to understand where our methods succeed and where they don't. This could enable an alternate direction of research on ensemble methods that switch one or more "less robust" anticipatory techniques based on changes in the underlying demand patterns. Additionally, the impact of operational parameters such as fleet size, number of depots, re-insertions of ignored orders, etc. on anticipation can also be explored.

Third, in our work, we only performed a limited study on how the tuning parameters vary the performance of anticipation. A thorough analysis on Θ, χ, and κ and how they impact the performance also needs to be conducted. Further, adapting the parameter values during operations could also be explored. How one tunes the parameters would depend on a multitude of external and internal factors and can be a considerable challenge. One approach to addressing this challenge could be through learning procedures.

Finally, assumptions that limit the practical implementation of our methods such as negligible trip computation time, fixed vehicle speeds, the lack of real data, etc. could be navigated to reduce the gap between these methods and their industrial applications.

A

Attributes of Vehicle Routing Problems

A.1. Constraints

Three main constraints affect the retail-based same-day delivery problem. These include the total working hours, the capacity of the vehicle, and the delivery time constraints [59].

- **Total Working Hours:** All SDD research has fixed total working hours within which it needs to satisfy as many requests as possible. Inability to satisfy a request or rejection (if allowed) is often associated with a penalty. With the total working hours fixed, the focus of research is to find optimum routes for serving the maximum number of requests.
- **Capacity:** A CVRP is associated with the limited capability of a vehicle to deliver orders due to lack of space or weight carrying capacity within the vehicle. The logistical context often influences the vehicle capacity. For instance, postal deliveries may consider infinite capacity due to the limited volume of an individual post. In contrast, transportation of customer grocery may occupy considerable vehicle space, and hence setting capacity constraints seems logical. We focus on problem formulations where capacity constraint incorporation is imperative to the routing criteria.
- **Time Constraints:** Same-day delivery can have two distinct types of constraints associated with delivery times. These are known as TW and DD. In TW formulations, a customer places an order to be delivered within a pre-specified time period during the day. On the other hand, DD formulations focus on delivering the order as early as possible after the request is made or within a maximum allowable delay time. The choice of time constraint selected within literature can largely affect the routing policy performance[14]. Further within the two constraint scenarios, a delay in the specified delivery times may or may not be allowed or only allowed up to a specific margin.

A.2. Type of fleet

A homogeneous fleet is one where all the deliveries are carried out by the same type of vehicle. In contrast, a heterogeneous fleet uses different types of vehicles or even a multi-modal transportation system. An example of such a fleet is given in Ulmer and Thomas (2018)[62] where the delivery can either be carried by a drone or a vehicle. A heterogeneous fleet brings with it the added complexity of selecting an appropriate vehicle for the delivery of services in addition to assignment and routing. In our work, we focus only on research based on a homogeneous fleet.

A.3. Fleet Size

Mainly research across SDD can be divided into two separate scenarios- a single-vehicle scenario and a multi-vehicle scenario. Single vehicle scenarios are generally simplifications of the fleet routing problems to highlight the effectiveness of a robust state-of-the-art approach such as the one deploying approximate dynamic programming for routing decisions[61][63]. The proposed work focuses on the multi-vehicle fleet scenario.

A.4. Objective Function

An objective function for a minimization or maximization problem defines the business goals in mathematical terms. In-vehicle routing, existing literature largely varies in the formulation of the objectives desired from a model. Some of the commonly observed objectives in same-day delivery include the number of customers serviced or rejected[66], cost of operations, total slack time available[61], travel distance, and total service delay[20]. An objective function can also be a combination of two or more objectives. For instance, the meal delivery problem mentioned in [48] considers a weighted combination of customer service delay and total travel time as its objective. In some works such as the one by Mitrovic-Minic *et al.*[36], anticipation is also heuristically incorporated by making suitable additions to the objective function. Comparison of different objectives remains out of the scope of this literature.

A.5. Uncertainties

Existing literature covers uncertainty in four main aspects - Travel time, Service time, Demand, and Requests[42].

- **Travel Time:** In a real-world setting, travel time between two locations is neither constant nor known in advance. This is largely attributable to varying traffic conditions, unforeseen accidents, or traffic management. In addition, the existence of multiple routes between the origin and the destination as well as the interdependence of travel between routes can also affect the uncertainty of travel times. In the context of modeling travel, a considerable amount of research focuses on assuming a static travel time. Albeit simple, a static travel time can not accurately represent a real-world scenario. There are three possibilities of modeling uncertainties in travel times - time-dependent, stochastic, or both. Ritzinger, in his literature review, highlights that models that consider uncertainty in travel times show the performance improvement of up to 10% over models that consider static travel time[49].
- **Service Time:** The time required to provide service to a customer could vary due to several factors such as varying duration to find parking or differing services required by the customer. At least for the case of retail-related vehicle routing, the service represents the drop off of the order at the required location. There is limited interest in the research community to model uncertainty in-service time[42] despite the fact that it may be of substantial benefit to the logistics industry. This could probably be attributed to the complexity of modeling the service time in the solution approaches. Currently, most research assumes a constant service time for simplicity.
- **Demand:** Applications where the actual quantity of items required are not known prior to reach the customer destination represent uncertainty in demand. Typical examples that showcase uncertainty in demand include garbage collection, gas supply, etc. In the case of on-demand same-day delivery problems, the quantity demanded is always known prior to reach customer destinations and hence, does not require any special modeling.
- **Request:** Uncertainty in requests is a widely researched problem class where the information about a customer request for pickup, delivery or service remains unknown prior to the order itself. All same-day delivery problems deal with stochastic requests and utilize a range of approaches to accurately anticipate orders.

B

Types Of Anticipatory Techniques

The main idea of anticipation is to incorporate future information in the decision-making model so as to obtain a better overall solution. This in turn will make the overall solution more optimal and robust to incomplete information. Several techniques of anticipation have been developed and deployed over the last two decades in the context of SDVRP. In the following sections, we will first present a classification of the different anticipation techniques. This will be followed by studying some commonly recurring anticipatory techniques. We will further analyze their possible advantages and drawbacks and briefly touch upon any heuristics required for their implementation.

Largely different anticipatory techniques differ in terms of the degree to which they provide anticipation to a decision point. As such Schneeweiss *et al.*[53] categorized anticipatory techniques by the following criteria-

- **Reactive vs Non-Reactive Anticipation:** Reactive anticipation classifies to those set of anticipatory techniques that incorporate future developments within the present decision. In particular, MDP and look-ahead-based solution methodologies that simulate future decision points are inherently reactive. Generally, such methods make use of the Bellman equation[5] for estimating the value of a decision. On the other hand, non-reactive approaches are techniques that do not simulate the future using the Bellman equation[59].
- **Implicit vs Explicit Anticipation:** Anticipatory techniques that do not utilize stochastic information, but rather depend on heuristics or rule of thumb are considered implicit anticipation. On the contrary, techniques that require information of future requests/demand or other factors of uncertainty and explicitly utilize them in the solution methodologies are known as explicit anticipatory techniques.

	Non-reactive	Reactive
Implicit	-	$\hat{\mathscr{B}}$
Explicit	$\hat{\mathbb{P}}$	$\hat{\mathscr{B}}, \hat{\mathbb{P}}$
Perfect	n/a	$\mathscr{B}, \mathbb{P}$

Table B.1: Classification of Anticipatory Techniques[59]: *Reactive techniques make use of derivations of bellman equation, where as sampled stochasticities are incorporated when the anticipatory techniques are explicit. A perfect anticipation scenario is also covered where all stochastic information and the original bellman equation can be used.*

These two broad criteria are further combined to make four distinct anticipatory classes- Non-reactive Implicit, Non-reactive Explicit, Reactive Explicit, and Reactive Implicit. These are highlighted in Table B.1, where $\mathbb{B}$ stands for the use of Bellman equation and $\mathbb{P}$ stands for utilization of stochasticity. It highlights that reactive anticipation is a technique that makes use of either the Bellman equation or one

of its derivations. Further, it covers the difference between explicit and implicit anticipation, highlighting the relevance of sampling stochastic information for the former. Another category in Table B.1 is perfect anticipation, it represents all known future states and therefore these can be solved without the approximation of Bellman equation and sampling. Perfect anticipation is tractable only for problems with q small number of decision states where the recursive calculation is possible. As such, most SD-VRP literature problems are too vast for perfect anticipation and therefore, not included in this literature survey. The interested reader can read the work of Schneeweiss for more details on perfect anticipation[53]. Further, we did not identify any reactive implicit approaches during the course of the literature study.

C

Approximate Dynamic Programming

Within the MDP framework, a number of decision points occur subsequently. For each decision point, a set of states is given. For each state and decision point, a subset of decisions is available. The combination of a state and a decision leads to a PDS. A transition would then lead to a new state from the state space. In the context of SDVRP, a state could indicate vehicle locations, the routes, the current timestep, and the set of ordered but unassigned requests. Decisions can be taken regarding the assignment of the unassigned orders to vehicles and their subsequent routing. A PDS could then represent the updated routes, new-vehicle locations, and assigned requests. A new request or subset of new requests at a later time would represent the transition to the next state[59]. For more details, the interested reader can review the modeling study by Ulmer *et al.*[64].

In the ADP framework, expected future values are accumulated at the PDS and depend on a base policy applied. The value of a post-decision state is computed by recursively applying the policy rules from the post-decision state to the terminal state. This is also given in Equation C.1.

$$V_S^{\pi}(S_k^x) = [\sum_{j=k+1}^{K} R(S_j, X^{\pi}(S_j))|S_k^x] \tag{C.1}$$

where $V_S^{\pi}(S_k^x)$ is the value of the PDS S_k^x when applying policy π.

Further, the decision rule of the particular state is computed by the Bellman equation[5], which is the sum of immediate reward and the expected reward given by the PDS value. This is shown in Equation C.2

$$X_k^{\pi^*}(S_k) = \max_{x \in X(S_k)} R(S_k, x) + V_S^{\pi}(S_k^x) \tag{C.2}$$

where $X_k^{\pi^*}(S_k^x)$ is a decision at the particular post-decision state.

As the PDS space can be vast, depending on several attributes, the computation of their values can suffer due to dimensionality constraints on the algorithm. In particular, aggregation and partitioning methods are used to simplify and reduce the state space for the approximation of the values. These methods are further explored in [59]. Further, two main ADP methods exist for mapping the state space to expected rewards- VFA and PFA. The interested reader is referred to the thesis study by Engelen for a complete working of a PFA[14]. Here, the author uses PFA to estimate weights for the convex objective function of the used policy. Additionally, VFA can further be classified as MVF and AVI. While MVF are parametric methods such as regression analysis, AVI computes independent and individual values of the PDS. The interested reader is referred to the study by Powell[41] for more details on the working of AVI's.

D

Illustrating Multiple Routes for a Vehicle-Trip Combination

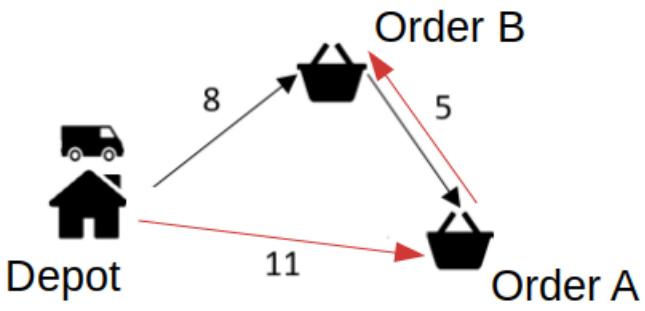


Figure D.1: Visualization of two possible sequences of deliveries for the same vehicle-trip combination

The scenario displayed in Figure D.1 showcases two ways in which a vehicle can serve a trip. Option 1: Indicated by the black arrow represents a route that first delivers order B and then orders A. Option 2: Indicated by the red arrow represents a route that first delivers order A and then orders B.

In terms of cost, let us assume that orders A and B are requested at time t = 0 seconds and have an optimum delivery of 11 seconds and 8seconds respectively. In this case, we assume service and pickup times are negligible. In the selection of option 1, the total operating cost is 8+5 = 13 seconds, whereas the cost due to delay to order A is 8+5-11=2seconds. On the other hand, option 2 results in an operator cost of 11+5=16 seconds, and a delay to order B of 11+5-8 = 8 seconds. If we assume a weightage factor β of 0.5, the total cost of option 1 becomes 7.5 seconds and that of option 2 becomes 12 seconds, making option 1 the more optimal route.

E

Alteration to the Implementation of VGA from Original Works

In the work by Kronmueller *et al.*[28] when an assignment of trips to vehicles at any decision interval of ψ is decided, requests that do not get picked up before the maximum waiting time $delta_{max}$ are re-inserted into the pool of orders. As such, they describe two different delay terms $\delta_{delay,real}$ which is defined by the service level and $\delta_{delay,heuristic}$ which is equivalent to the maximum delay at which the method performs well. As long as $\delta_{delay,heuristic} < \delta_{delay,real}$ orders can be re-inserted after being ignored. The re-inserted candidate gets a new release time t, which is equivalent to the current time. This is done for satisfying the feasibility constraints of the re-inserted order. However, the original release time is used for the computation of the overall costs. The total number of re-insertions ζ can be defined as-

$$\zeta = \frac{\delta_{delay,real}}{\delta_{delay,heuristic}} \tag{E.1}$$

This re-insertion affects performance improvements greatly.

We introduce a slight modification to their approach. In our work, we keep $\delta_{delay,heuristic} = \delta_{delay,real}$. As a result, any request that gets ignored by the system is not re-assigned in the next iteration but removed from the pool of orders. While this grossly limits the capability of the VGA approach, results with re-insertions can be hard to interpret and analyze. As the main objective of our research is to compare the benefits of anticipation over a myopic setting, absolute method performance is not of prime importance to us. This simplification grossly eases the comparison between the proposed anticipatory techniques and their myopic counterpart.

F

Explored Anticipatory Techniques

In this section, we discuss each of the anticipatory techniques explored during the course of our research. These include several waiting approaches, the double edge horizon technique by Minic *et al.*[37], modifications to the assignments by introducing rewards first proposed by Fielbaum *et al.*[15], and a trip trimming method inspired by the slackness-savings approach first proposed by Kalina *et al.*[26].

F.1. Waiting Strategies

A common approach found in the literature for anticipation is the concept of waiting to accumulate more requests before a trip is executed. The fundamental idea of this approach is based on the premise that if a vehicle is capable of delivering requests before their delivery deadline then there may be a benefit in waiting at the pick-up location for any future request possibilities as long as the delivery deadline of all the requests along the trip is not violated. This way the vehicle can deliver more orders by clubbing them and limiting the extent of the detour the vehicle may need to take otherwise.

In our study, the waiting strategy was applied to the VGA algorithm post computation of trips. At first, the waiting strategy computed how much estimated time was available for delivering each order before the delivery deadline (constraint of maximum delay). The free time available is considered slack time. Next, the strategy computed the minimum amount of slack available across all the pickups and deliveries of the trip. Once the minimum slack was computed, it was assigned as the waiting time of the corresponding vehicle. As a result, a vehicle with a non-zero waiting time was required to wait at specified locations for a total duration equivalent to the waiting time. The cost of the trip was also adjusted to compensate for the additional waiting time that results in an added delay to the customer. This cost adjustment is highlighted in Equation F.1. In addition to the waiting added to each trip, the ILP solver also receives an adjusted trip-vehicle cost parameter and hence, may assign an entirely new trip for the delivery of goods. Each of the waiting strategy approaches is discussed in the following sub-sections and their simulation results are illustrated in Section F.1.4.

$$\gamma_A(v_j, T_i, \pi_{v_j}) = \gamma(v_j, T_i, \pi_{v_j}) + \lambda \cdot \sum_{r \in T_i} \delta_r \tag{F.1}$$

where δ_r is the additional delay in delivery for each order r in trip T_i due to waiting of the vehicle at the pick-up location, and λ is a weight associated with delay computation of a trip.

F.1.1. Wait-First

The wait-first approach presented by Minic *et al.*[37] and discussed briefly in Chapter 2 utilizes the entire slack. It does so by waiting at the depot as long as it does not violate the constraints of any order of the trip. The vehicle waits at the first pickup location of the planned route.

F.1.2. Demand-Based Waiting

Demand-based waiting aims to improve the limit the slack consumption of wait first by rationing the available waiting time. It attempts to do so as a function of the change in input demand to the system. As such, this approach is similar in principle to the advanced dynamic waiting proposed by Minic*et al.*[37]. Under this approach, we look at demand at six consecutive simulation runs $i+1, i, i-1, i-2, i-3, i-4$. It is imperative to note that each simulation run represents a fixed time horizon simulation. We compute a metric called the demand value factor at i as the moving average of the demands from $i-4$ to i. Similarly, the overall demand value factor at $i+1$ is the moving average value from $i-3$ to $i+1$. Whenever the demand factor at $i+1$ is greater than the demand factor at i, a waiting factor is computed according to equation F.2 and is multiplied with the total available slack at iteration $i+1$. The resulting value is the total time the vehicle waits at the depot. On the other end, if the demand factor at i is greater than $i+1$, no waiting is applied. The intuition behind this approach is that in situations when input demand has already peaked, the algorithm should focus on optimizing the route instead of waiting. This was devised in order to further limit the negative impact of information loss due to the fixed time horizon simulations instead of re-execution at every new request as is common in other waiting strategy studies.

$$WF = max(0, (1 - DVF_i/DVF_{i+1})) \tag{F.2}$$

Here WF stands for waiting factor, and DVF_i stands for Demand value factor for simulation at time $i-1$ to i.

F.1.3. Distributed Waiting

All of the waiting strategies discussed to utilize the waiting time at the first pickup location of the route followed in the vehicle-trip plan. However, as shown by [9] and [57], distributing the waiting time proportionally throughout the trip performs substantially better than the wait first approach. In particular, they found that distributing the waiting times gives similar service rates as waiting first without increasing the number of vehicles in the simulation. In our approach, we distribute the total waiting time at each pick-up location encountered by the vehicle during the course of the route of the vehicle-trip combination.

F.1.4. Results of Waiting strategies

The results of the different waiting strategies were compared on 30-minute simulations of orders with uniform spatial and temporal distributions in Figure F.1. The figure highlights the overall cost, service rate, time KPIs, mean loaded parcels, total driven distance, and distance per order for each of the waiting strategies.

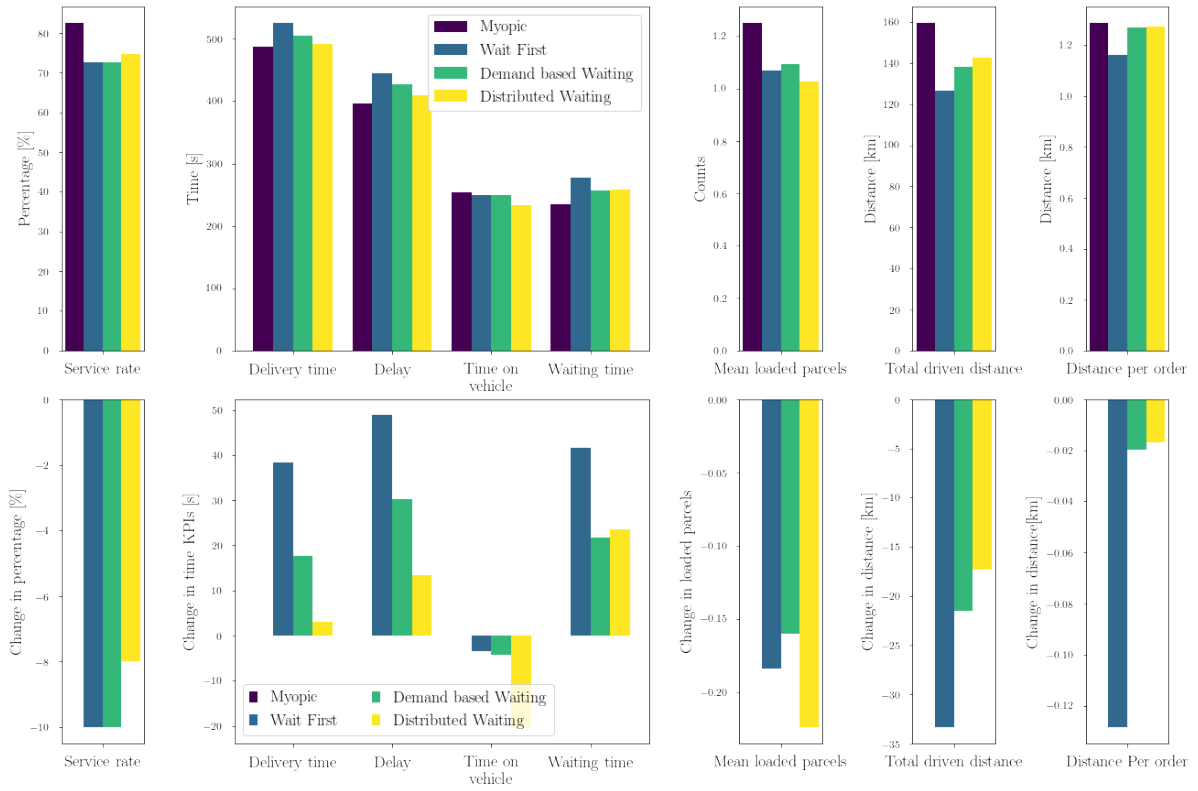


Figure F.1: Comparative Performance of Waiting strategies: overall cost, service rate, time KPI's, mean loaded parcels, total driven distance, and distance per order are displayed for each of the waiting strategies. The top figure represents absolute performance, whereas the bottom figure represents differences in performance with respect to the myopic simulation.

Analyzing the performance of the preliminary results, we can see that all waiting strategies perform worse than the myopic approach. This can be attributed to the fact that in our approach we use a fixed time cadence for simulations, whereas in Minic's and other waiting strategy studies, the time cadence is dependent on new information. Neglecting this new information in time could grossly affect the performance benefit offered by waiting strategies. As a result incorporation of waiting strategies with VGA or other rolling horizon-based techniques may not be advisable.

F.2. Trip Trimming

Motivated from the slackness savings heuristic first covered in [26], the goal of the trip trimming approach is to provide temporal flexibility to the vehicle route. Intuitively what this means is that the more free the vehicle would be at any given time, the greater would be its capability to serve orders in the future. While the fundamental motivation of both slackness saving and trip trimming is the same, they largely differ in their implementation. We cover this briefly below-

Slackness savings attempts to add flexibility to the route by modifying the cost computation of the DVRP. Instead of optimizing for distance traveled as is common in routing problems, it looks at a weighted combination of vehicle distance and reduction in slack. This is shown in Equation F.3.

$$c_i = \mu \cdot (SLR_i) + (1 - \mu) \cdot (f_i) \tag{F.3}$$

$$SLR_i = \sum_{i \in \Gamma} (sl_i - sl_{i-1}) \tag{F.4}$$

Here, c_i represents the cost obtained at a single simulation run i, f_i represents the distance traveled by the vehicle at simulation run i, μ represents the weight parameter of the convex combination of distance and reduction in slack, SLR_i represents a reduction in slack at i, sl_i and sl_{i-1} represent the slack before and after insertion of a request at simulation run i. Finally, Γ represents all requests that

are currently requested and known. It is important to note that slackness savings according to Kalina *et al.*[26], re-planned trips at every new request. Hence, the simulation runs i corresponded with the request i for the study. This is in contrast to our study where i stands for a fixed time step in the simulation and only executes once the previous time step has completed its execution.
On the other hand, the trip trimming approach proposed by us makes no adjustment to the objective cost function discussed in Equation 3.1. Instead, it eliminates the last pick-up and delivery of the trip to provide free slack to the vehicle for future orders. This in turn frees the vehicle's bandwidth to serve orders better in the future. This is better explained through an illustration shown in Figure F.2. Further, trip trimming just like the VGA executes on a rolling horizon basis and not at every new request. Trip trimming is further divided into two sub-approaches as covered below:

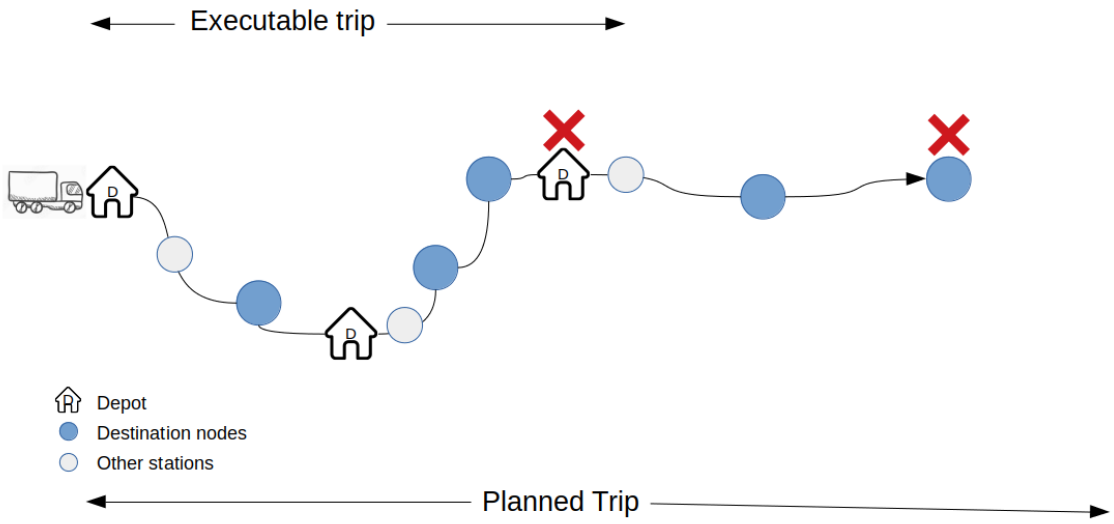


Figure F.2: Illustration of Trip Trimming: The D huts represent the depots, the blue nodes represent delivery locations, and the white nodes represent nodes traversed along the route. The cross above each node represents trip trimming.

- **Trim Trip:** The first approach of trip trimming involves the elimination of all last order pickup and deliveries irrespective of the vehicle and trip size even if it would eliminate a trip that could be completed before the next simulation run.
- **Trim Trip Criterion Constraint:** The second approach within trip trimming is identical to the first, however adds a constraint before it eliminates the last pick-up of the trip. This constraint allows a request to be eliminated only if the vehicle already has two requests scheduled for pick-up before the to-be-eliminated request. This was added as we hypothesized that the original approach may over-compensate by eliminating orders that could have been successfully completed without vastly affecting future performance.

F.2.1. Results of Trip Trimming

The preliminary results of trip trimming are highlighted in Figure F.3. All experiments were executed in an environment of 150 orders distributed uniformly in space and time along with the graph and having an operation period of 30 minutes.

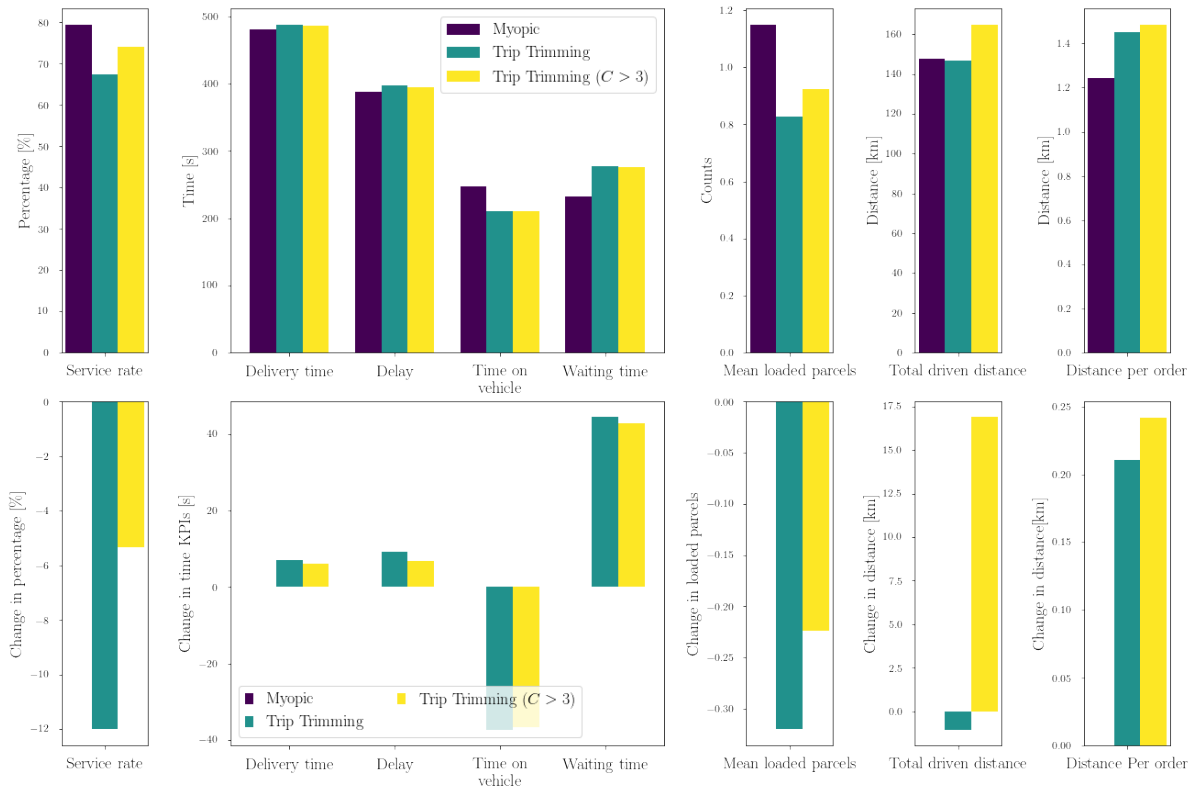


Figure F.3: Comparative Performance of Trip Trimming Based approaches: overall cost, service rate, time KPI's, mean loaded parcels, total driven distance, and distance per order are displayed for each of the waiting strategies. The top figure represents absolute performance, whereas the bottom figure represents differences in performance with respect to the myopic simulation.

Both of the trip-trimming approaches performed worse than the myopic approach. This performance worsened as the problem was scaled to large-scale instances and hence, further exploration into trip trimming-based anticipation was halted for exploring more robust anticipatory techniques.

F.3. Double Edge Horizon

Introduced in Chapter 2, Double Edge Horizon by Minic *et al.*[36] was adapted to work in tandem with the assignment procedure of VGA. This adaptation is highlighted by means of two Algorithms 7 and 8. The first of the two algorithm introduces how the reduction in slack is computed in the original DEH method to provide the reader with a fundamental understanding of the method. This reduction in slack is added to the objective function as per the slackness savings Equation F.3. The second Algorithm 8 highlights the modifications made to our problem formulation and the VGA framework defined in Chapter 3 for incorporating the double edge horizon framework in its objective function.

The Algorithm 8 successfully highlights that a change in delay in the given problem context translates to a reduction in slack from the DEH problem covered in Algorithm 7. However, we identify several caveats that make the implementation of DEH with VGA non-trivial. A simple illustration depicting the three main requirements for implementing VGA and DEH together is depicted in Figure F.4. The first of the three requirements is resolved by Algorithm 8. The second and third requirements are explained further:

It is important to note that there is a major difference in the cadence with which the double edge horizon executes in Minic's study. While Minic's algorithm re-plans its trips after every new request, VGA executes via a rolling horizon period. This makes the translation of double edge to VGA non-trivial. This difference is indicated in the algorithms by the variable i. In the original Double Edge Horizon framework, i represents the arrival of a new request that re-executes the entire plan. This brings us to challenges with adapting the VGA to suit the double edge horizon framework.

Algorithm 7 Double Edge Horizon - Reduction in slack

$e_i \leftarrow$ earliest start time of request i
$l_i \leftarrow$ latest start time of request i
$E_i \leftarrow$ earliest possible start time of request i
$L_i \leftarrow$ latest possible start time of request i
$S_i \leftarrow$ service time at request i

$E_i \leftarrow max(e_i, E_{i-1} + S_{i-1} + t_{i-1,t})$

$L_i \leftarrow min(l_i, L_{i+1} - S_i - t_{i,i+1})$

$Slack(SL_i) \leftarrow L_i - E_i$

Reduction in slack $\leftarrow \sum_{i \in 0, t_{end}} (SL_i - SL_i^{'})$

Algorithm 8 Adjustment to VGA for reduction in slack computation

$e_i \leftarrow$ Request time + optimum travel time at simulation run i
$l_i \leftarrow$ Request time + optimum travel time + max delay at simulation run i
$E_i \leftarrow$ Actual arrival time = Request time + Optimum travel time + delay at simulation run i
$L_i \leftarrow l_i$

$Slack(SL_i) \leftarrow L_i - E_i =$ Request time + optimum travel time + max delay - request time - optimum travel time - delay

$Slack(SL_i) \leftarrow$max delay - $delay_i$

Slack before insertion (SL_{i-1}) $\leftarrow$max delay - $delay_{i-1}$

Slack after insertion (SL_i) $\leftarrow$max delay - $delay_i$

Reduction in slack $\leftarrow delay_i - delay_{i-1}$

Furthermore, the planning horizon for double edge horizon methodology is at least 4 hours in the future with the short-term horizon itself spanning an hour. On the contrary, our method plans trips for only 8 minutes of operation. As a result, the division of a short-term and long-term horizon for the VGA framework is not straightforward. In particular, the short-term horizon computation is based on the assumption that the majority of the requests in that horizon are already known. This is because DEH follows a time window formulation where orders requested at a particular time have to be delivered at a specific future duration. As a result, several orders occur much before their delivery durations. In contrast, orders generated in our problem context have to be delivered as quickly as possible. Hence, exact information about orders to be delivered even a few minutes into the future is hard to predict. Thus, simulations with DEH were not executed due to their inability to work in synergy with the VGA framework.

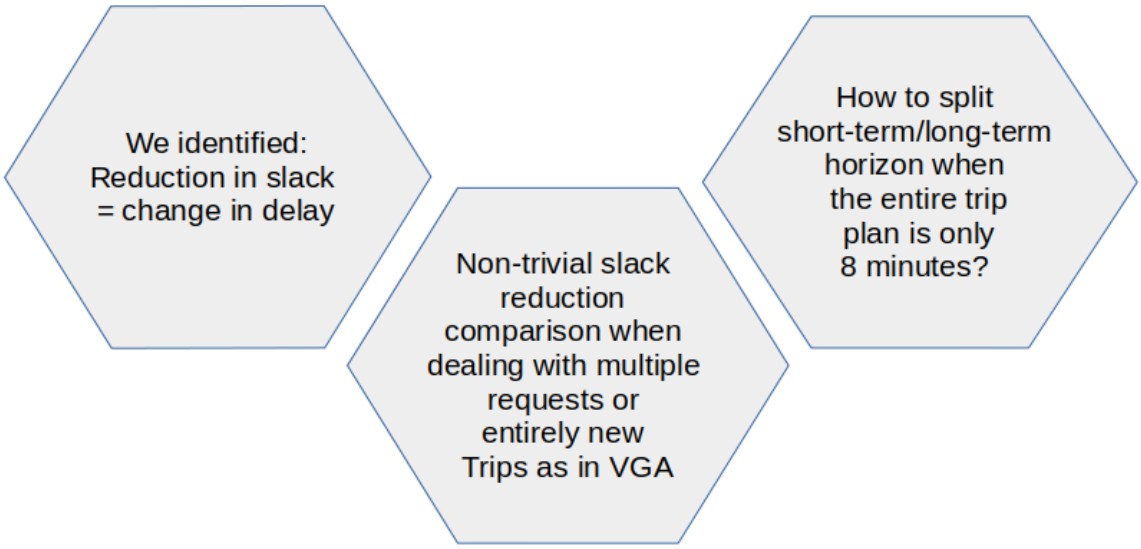


Figure F.4: The double edge horizon dilemma Illustrates the three pillars for integrating VGA with DEH.

G

Preliminary Results of Anticipatory Approaches

As a part of our initial exploration, we implemented several anticipatory approaches introduced in Chapter 4 and discussed in Appendix F over the VGA algorithm. The goal was to identify a suitable anticipatory technique that complimented the VGA framework. We evaluated the performance of each approach - wait-first, trip trimming, rewards adjustment, and over the defined performance KPIs - overall cost, service rate, time KPIs, mean loaded parcels, and total travel distance of all vehicles.

For the preliminary results, all simulations were executed on a data set of 150 orders and 30 minutes of operation periods. The demand was distributed as a clustered Gaussian distribution in space and a combination of Uniform and Gaussian distribution in time. The data distribution is highlighted in Figure G.1. The spatial map in Figure **??** represents the demand intensity where intensity on any node is computed as the demand on the corresponding node divided by the maximum demand over all nodes of the graph.

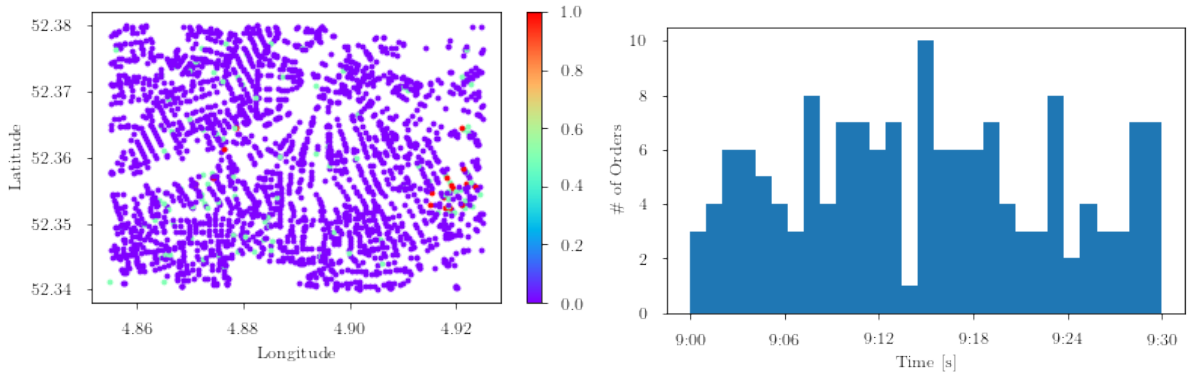


Figure G.1: Spatial & Temporal Distribution of demand for Preliminary Simulations: The data set instance comprises a clustered Gaussian spatial distribution of 3 clusters and a random uniform temporal distribution. The total number of orders is 150 and the period of operation is 30 minutes. The spatial graph is normalized such that red nodes indicate the highest demand and violet nodes indicate low or no demand. This is also indicated in the color bar. The temporal distribution is divided into 30 bins, 1 for each minute of the simulation.

The results of the simulation are further covered in Figure G.2 and Table G.1. Figure G.2 highlights the overall cost, service rate, time KPIs, mean loaded parcels, and total travel distance of all vehicles. The last bar (represented in red) in the graph depicts the results obtained in case of perfect anticipation. Perfect anticipation is a simulation scenario that represents the upper benchmark of performance improvement due to anticipation. We compare the performance across each of the explored anticipa-

tory techniques that are used in conjunction with the same base assignment procedure P of the VGA algorithm. It is substantially clear from the figure that amongst all the selected approaches, adjustment using rewards is the only technique where all performance parameters are improved considerably over the myopic simulation. This suggests that the improvements noted by Fielbaum *et al.* [15] over myopic simulations in DARP can also be translated to same-day grocery deliveries.

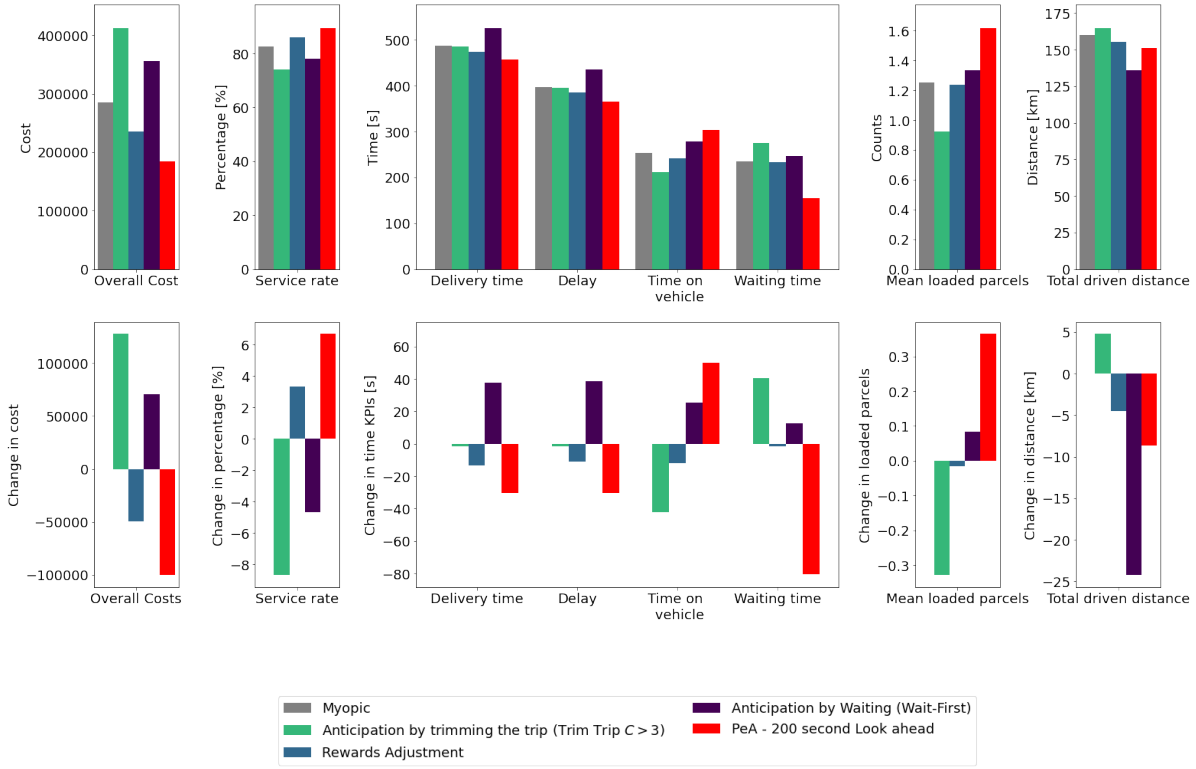


Figure G.2: Performance Comparison of Several Adjustment Techniques versus the Myopic Algorithm: Overall cost, service rate, time KPI's, mean loaded parcels, and total travel distance of all vehicles are displayed. The grey bar represents the performance of the Myopic simulation. Each other bar represents a unique anticipatory technique. Additionally, the Perfect Anticipation - 200s look ahead is represented by the red bar. Perfect Anticipation provides the maximum potential of anticipatory improvement over the VGA for the given data instance. The figure at the top represents absolute results whereas the bottom figure represents the difference in performance between the approach under consideration and the myopic simulation.

Table G.1: Preliminary simulation results for anticipatory techniques and the myopic approach

S.no	Simulation Type	Overall Cost	Service Rate	Orders Delivered	Delivery Time	Delay Time	Time On Vehicle	Waiting Time	Distance Travelled
1	Myopic	284664.47	82.67	124	488.07	396.52	253.19	234.88	159.93
2	Anticipation by Trimming Trip (C > 3)	412005.84	74.0	111	486.38	395.02	210.95	275.42	164.68
3	Rewards Adjustment	234935.19	86.0	129	474.78	385.39	241.37	233.4	155.38
4	Wait-First	355515.86	78.0	117	525.76	435.01	278.45	247.31	135.71
5	PA-200 second Look Ahead	184607.13	89.33	134	457.79	366.14	303.22	154.57	151.26

Computation of Dynamism

In this section, we mathematically define the computation of dynamism by Van Lon *et al.* [65]. Their definition of dynamism as mentioned previously computes the degree of continuity of change. This means any deviation from a perfectly continuous occurrence of orders will end up being less dynamic. Note that we only briefly highlight how dynamism is computed mathematically. The reader is urged to peruse their work for an in-depth study.

At first, we define the inter-arrival times between any two orders as follows-

$$\Delta = \{\delta_1, \delta_2, ..., \delta_n\} \qquad \forall o_i \in O \tag{H.1}$$

where Δ is the list of inter-arrival times and δ_i is the time gap between order i and $i-1$.

The authors further define a case of 100% dynamism, wherein every event has the same inter-arrival time. This is illustrated in Figure H.1.

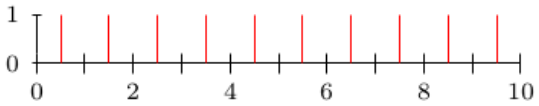


Figure H.1: A scenario of the perfect inter-arrival time

The perfect inter-arrival time θ can be computed as per Equation H.2.

$$\theta = \frac{\tau}{\epsilon} \tag{H.2}$$

where τ is the total time period of operation and ϵ is the number of orders in that period.

Once the perfect inter-arrival time is obtained, deviations σ_i for each order i from the 100% dynamism case can be computed as follows-

$$\sigma_i = \begin{cases} \theta - \delta_i & \text{if } i = 0 \text{ and } \delta_i < \theta \\ \theta - \delta_i + \frac{\theta - \delta_i}{\theta} \cdot \sigma_{i-1} & \text{if } i > 0 \text{ and } \delta_i < \theta \\ 0 & otherwise \end{cases} \tag{H.3}$$

Consequently, the deviation for an entire scene is given as-

$$\sum_{i=0}^{|\Delta|} \sigma_i \tag{H.4}$$

Next, a theoretical maximum of the deviation $\tilde{\sigma_i}$ is computed as follows-

$$\tilde{\sigma_i} = \begin{cases} \theta + \frac{\theta - \delta_i}{\theta} \cdot \sigma_{i-1} & \text{if } i > 0 \text{ and } \delta_i < \theta \\ \theta & otherwise \end{cases} \tag{H.5}$$

The value of $\tilde{\sigma_i}$ is computed to normalize the deviations of inter-arrival times.

Finally, the value of dynamism is computed as-

$$dynamism = 1 - \frac{deviation}{maximumdeviation} = 1 - \frac{\sum_{i=0}^{|\Delta|} \sigma_i}{\sum_{i=0}^{|\Delta|} \tilde{\sigma_i}} \tag{H.6}$$

Spatial Gaussian Distribution Methods

Three methods were explored during the course of this study to generate a perfect 2D Spatial Gaussian distribution. These were the displacement to the center method, index method, and independent axis method. The independent-axis method has already been discussed in previous chapters and hence, will not be covered in this section. The other two methods are algorithmically described in the following sections.

I.1. Displacement to Center Method

In this method, first, the center node of the graph is obtained for computing the travel time distance between all nodes of the graph and the central node. The distances are stored in memory and using a truncated normal distribution function, a new Gaussian distribution of distances is obtained with the minimum distance from the center node as the mean and a function of maximum distance from the central node as the standard deviation. The Gaussian distribution contains as many samples as the number of orders to be generated. Next, each element of the Gaussian distribution is subtracted from all the elements of the list containing the distance of each node from the center. At every iteration, the index where the difference of distances is minimum is selected and stored in memory. The final collection of indexes represents the indexes of the nodes of the graph where demand is to be generated.

I.2. Index Method

In the index method, first, the center node of the graph is identified and the index of the corresponding node on the graph is stored as the central index. Next, a standard deviation of the indexes is computed by passing all indexes of the graph. Finally, a truncated normal distribution of indexes is generated as per Equation I.4. The indexes are used to identify the appropriate nodes of the graph where demand is to be generated.

Algorithm 9 Generation of Spatial Demand using Displacement to Center Method

input: num_orders, $G = (N, A)$
output: $demand\_nodes$
begin
Get Graph Center,
$x, y, z \leftarrow 0$
for $n \in N$ **do**
$latitude \leftarrow rad(n \rightarrow latitude)$
$longitude \leftarrow rad(n \rightarrow longitude)$
$x+ = cos(latitude)cos(longitude)$
$y+ = cos(latitude) * sin(longitude)$
$z+ = sin(latitude)$
end
$total \leftarrow length(N)$
$x, y, z \leftarrow x/total, y/total, z/total$
$centre \leftarrow atan2(y, x), atan2(z, (x^2 + y^2)^0.5)$

for $node \in N$ **do**
$dlat \leftarrow centre.longitude - node.longitude$
$dlon \leftarrow centre.latitude - node.latitude$

$$a = sin(dlat/2)^2 + cos(node \cdot latitude) * cos(centre \cdot latitude) * sin(dlon/2)^2 \quad \text{(I.1)}$$

$$c = 2 * atan2(sqrt(a), sqrt(1 - a)) \quad \text{(I.2)}$$

$$dist = 6373 \cdot c \quad \text{(I.3)}$$

if $dis \leq dist_{max}$ **then**
$dist_{max} = dis$
$best\_node \leftarrow node$
end
end
$centre\_node \leftarrow best\_node$

Find distance of every node from center,
$dist\_center \leftarrow \emptyset$
for $n \in N$ **do**
$dist\_center \leftarrow dist\_center, travel(center\_node, n)$
end

Generate truncated normal distribution of distances,
$dis\_gaus = []$
while $count(dis\_gaus) \leq num\_orders$ **do**
$dis\_gaus \leftarrow truncNorm(min(dist\_center), max(dist\_center/3))$
end

Mapping gaussian distribution of distances to closest node,
for $i \in dis\_gaus$ **do**
$rel\_dis \leftarrow abs(dist\_center - i)$
$indexes \leftarrow indexes, rel\_dis.index(min(rel\_dis))$
end

$demand\_nodes \leftarrow N[indexes]$
return $demand\_nodes$
end

Algorithm 10 Generation of Spatial Demand using Index Method

input: num_orders, $G = (N, A)$
output: $demand\_nodes$
begin
  Get Graph Center,
  $x, y, z \leftarrow 0$
  for $n \in N$ **do**
    $latitude \leftarrow rad(n \rightarrow latitude)$
    $longitude \leftarrow rad(n \rightarrow longitude)$
    $x+ = cos(latitude)cos(longitude)$
    $y+ = cos(latitude) * sin(longitude)$
    $z+ = sin(latitude)$
  end
  $total \leftarrow count(N)$

  $x, y, z \leftarrow x/total, y/total, z/total$

  $centre \leftarrow atan2(y, x), atan2(z, (x^2 + y^2)^0.5)$

  Find index of central node in the graph G,

  $central\_index \leftarrow N.index(centre)$

  Find the standard deviation of indexes of graph G,

  $sigma \leftarrow std(0, count(N))$

  Find the truncated Gaussian distribution of indexes,

$$x \leftarrow truncNorm((0 - central\_index)/sigma, (count(N) - central\_index)/sigma) \quad \text{(I.4)}$$

  $demand\_nodes \leftarrow N[x]$
  return $demand\_nodes$
end

J

Depot Distribution:

In this section, we highlight other approaches for distributing the depots across the graph. These are discussed below-

J.1. Uniform Distribution Method:

In the Uniform Distribution Method, k depots are randomly distributed along with all the nodes of the graph. This can be observed in Figure J.1.

Figure J.1: Uniformly distributed Depot Locations

J.2. Shortest Distance Distribution Method:

In the Shortest Distance Distribution Method, an integer linear program is used to minimize the distance of the depot to its neighboring nodes. At first, a data set highlighting the travel time between all nodes is prepared. Next, the reach-ability of a node to every other node is established basis a threshold travel

time value. This threshold is determined empirically. Finally, an ILP with a cost matrix minimizing the distance between the nodes and depots is prepared under the constraint that every node should be reachable by at least one depot. A final constraint setting the minimum number of depots to at least k is established and the depot distribution is obtained. This is illustrated in Figure J.2

Figure J.2: Depot distribution using the Shortest Distance Distribution Method

J.3. Why K-center distribution?

After a detailed analysis of preliminary results from each method, we observe that the performance of the base method remained best with the Shortest Distance Distribution of depots and worst with the Uniformly Distributed depot locations. The K-center approach results in solutions in between the two approaches. This can be attributed to the fact the uniform distribution method leads to spatially imbalanced depot locations whereas the integer linear program approach leads to highly optimized depot locations. As we hope to replicate the performance in as realistic a scenario as possible, we select the K-center method that only moderately optimizes the distribution of depots on the graph.

K

Numerical Values of Simulation Parameters

Table K.1: Unique Data Set Types

Parameter	Value
Maximum Delay δ_{delay}	8 minutes
Number of vehicles n	10
Capacity per vehicle c	10
Weight of cost function β	0.5
Penalty for rejection of request α	10000
Maximum considered vehicles per candidate	7
Maximum size of each Trip	10
Allowed Re-insertions of orders ζ	0
Service time per order $\delta_{service}$	30 seconds
Loading time per order $\delta_{loading}$	15 seconds
Time-interval between each decision Δt	100 seconds
Extra time at the end of the day	15 minutes
Number of depots condisered per order x	5
CTV graph calculation timeout per vehicle $\rho_{max,CTV}$	30 seconds
Rewards weight for assignment including rewards Θ	100
Rewards weight for assignment including penalty κ	1
Penalty weight in assignments including penaltyχ	1
Number of zones/sub-regions obtained from the ILP solver M	12
Upper bound of reachability t_M	100 seconds

Simulation

To assess the performance of the fleet using the anticipatory routing algorithm, on-demand SDD operations need to be simulated. In this section, we first introduce the graph network. In the subsequent section, we highlight the movement of the vehicle over the graph network in simulation.

L.1. Graph

The city of Amsterdam is identified for simulating the performance of the proposed solution. For this purpose, the map of Amsterdam is first extracted from the open-source Open Street Maps repository[4]. The graph obtained consists of a large graph of nodes and edges comprising of waterways, road-ways, pedestrian-pathways, etc. The map is filtered for only motorways and highways that are accessible by vehicles as well as for a specific size. The final graph contains 2717 nodes and 5656 edges. Amongst the 2717 nodes, fifteen are chosen as depot nodes basis the depot distribution algorithm. The depot distribution algorithm adds stochasticity to the location of the depots. One such graph is illustrated with Figure L.1.

L.2. Vehicle Movement

As discussed in Section3.1, vehicles travel along with any two known locations $x1$ and $x2$ where the travel time is given by the function $travel(x1, x2)$. However, no prior information about the travel time between the graph nodes is available. To compute the travel times, we first extract distance information between any two nodes that are connected by an arc from the OSM graph [4]. We then assume that each vehicle travels with a constant velocity of 10m/s and is not affected by congestion or one-off incidents. Using the distance and constant velocity, we compute the travel times for traversing on any of the 5656 edges of the graph[28]. While this gives the travel time between any two directly connected nodes, the question arises on how shall the vehicle travel across two non-consecutive nodes. We solve this by first computing the shortest path between any combination of nodes using the Dijkstra shortest path algorithm [13]. Once the shortest paths are known, we calculate the total travel time as the sum of individual travel times of all the arcs included in the shortest path between any two nodes. We store the resulting travel times between the node-node combinations offline. This can be then queried by the solution framework for online route optimization during simulation.

Figure L.1: Map of Amsterdam: Representation of the graph obtained from Open Street Map indicating road network, stations, and depot locations. The waterways are highlighted by thick light blue lines, while the transportable network is highlighted by dark blue thin lines.

M

Perfect Anticipation

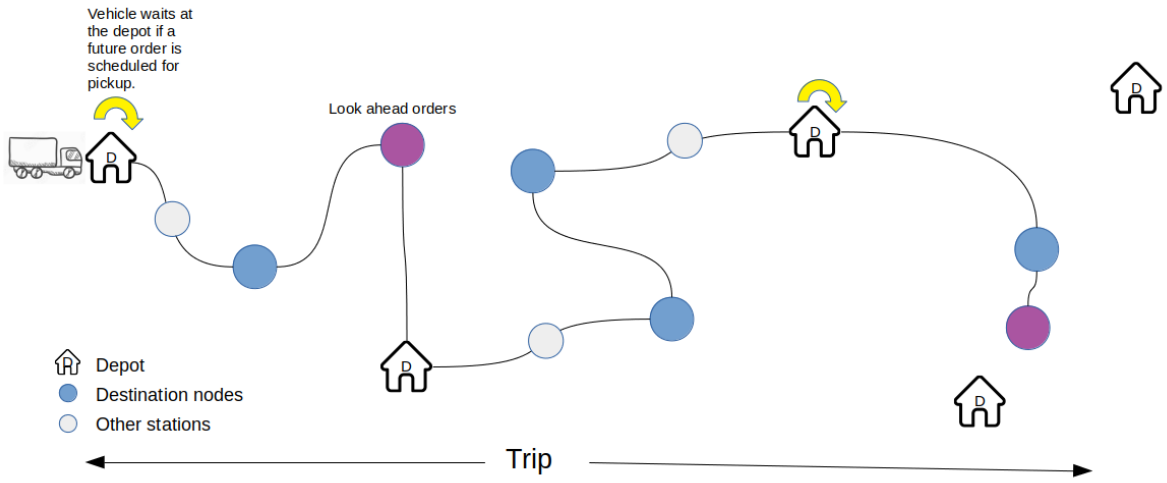


Figure M.1: Illustration of Perfect Anticipation

The results of any anticipatory policy are hard to evaluate without a common benchmark of an optimal policy. In our study, we attempted to define an upper bound to anticipation using the concept of perfect anticipation. Perfect anticipation works by allowing the VGA framework to look at exact future demand up to a certain horizon. This information of future and current orders are then used to plan for a trip. Additionally, suitable adjustments are made within the VGA framework to wait at the depot if the vehicle reaches its location before an order is requested. Perfect anticipation is further explained by means of an illustration shown in Figure M.1.

The optimal amount of look-ahead was evaluated by running VGA and perfect anticipation with different times of look ahead from 100 seconds to 700 seconds. The overall cost, service rate, time KPI's, mean loaded parcels, and total travel distance of all vehicles are compared with the Myopic simulation and illustrated in Figure M.2. Further, the time estimate for executing the complete look ahead for each simulation is highlighted in Figure M.3.

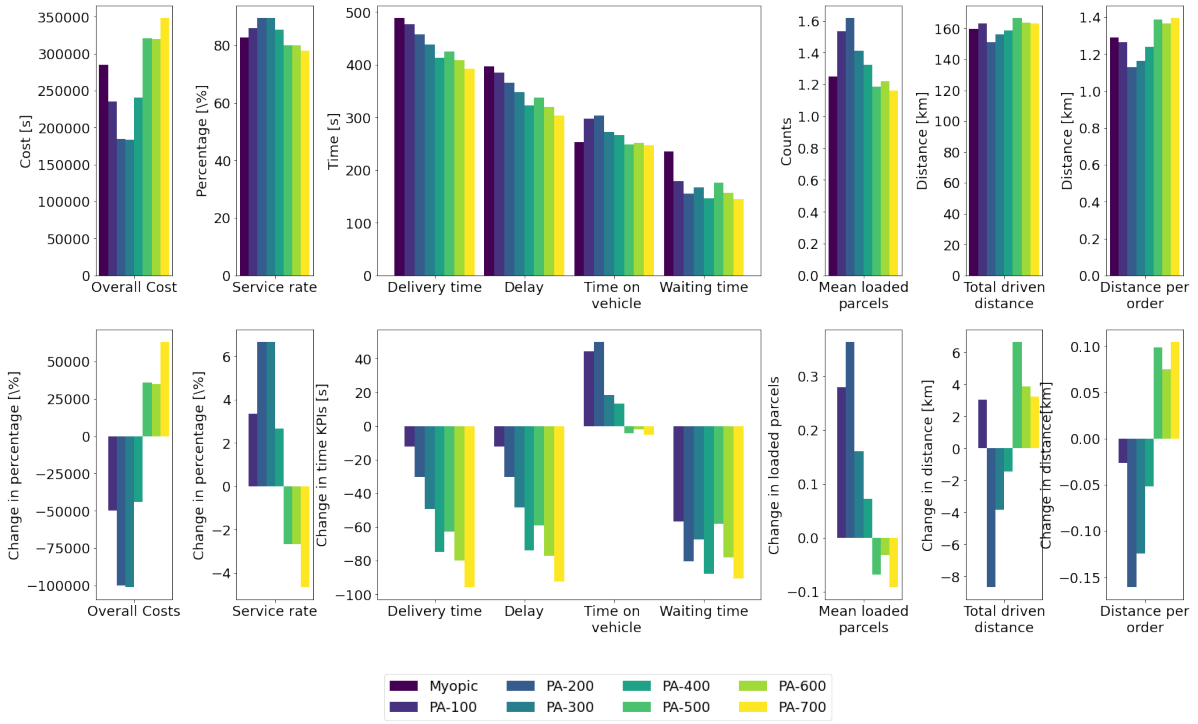


Figure M.2: Results of Perfect Anticipation: Overall cost, service rate, time KPI's, mean loaded parcels, and total travel distance of all vehicles are displayed. Each bar represents a perfect anticipation scenario with a different look-ahead horizon.

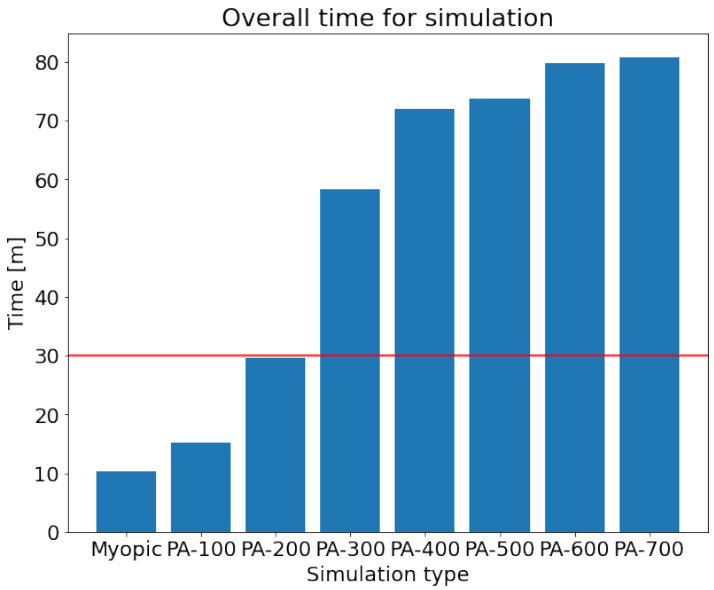


Figure M.3: Simulation time vs Look ahead of Perfect anticipation for 30 minutes of simulation: Each bar represents the time taken to complete the simulation when using the underlying technique given on the x-axis.

It is clear from the figures, that the improvement in performance saturates after a certain level of look ahead and even worsens. This is primarily due to the vehicle-trip computation limit $\rho_{CTV,max}$ of 30 seconds that cuts off the exhaustive trip generation function. As such with more number requests, more combinations of trips can be formed and the simulation time increases exponentially. This can be observed in Figure M.3. The cut-off keeps the simulation time in check at the cost of solution quality. For our demand and graph inputs, it was observed that 200 seconds of look ahead made for a reasonable parameter for both the solution quality as well as the simulation time.

N

Performance Sensitivity to Change in Θ

Variation in performance KPIs due to change in tuning parameter Θ is represented in the Figure N.1. The underlying data used for simulations is the realistic-data instances covered in Chapter 5. From the figure, the range 50-100 for Θ seems to be reasonable for all KPIs considered.

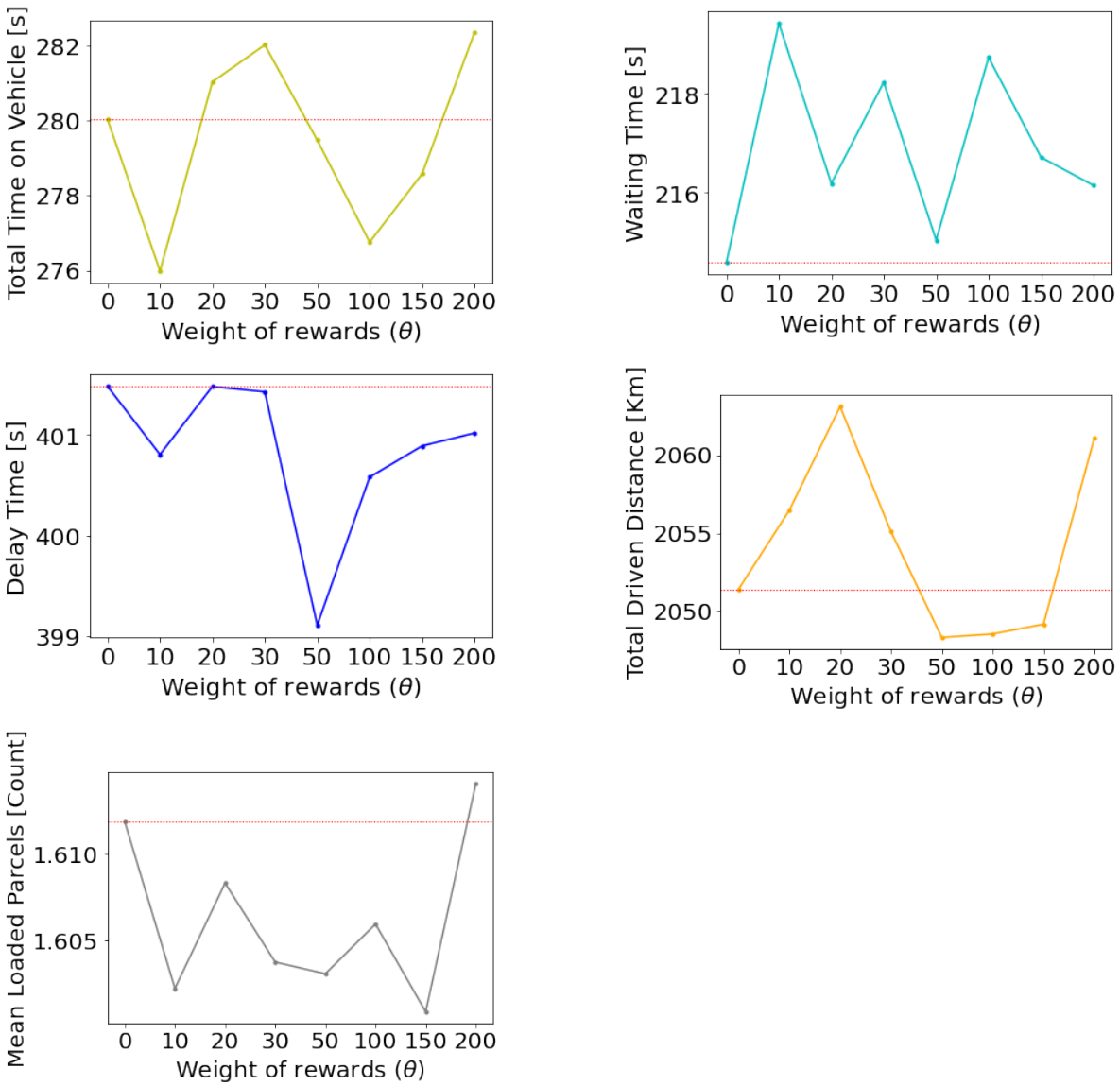


Figure N.1: Time On the vehicle, waiting time, delay time, total distance traveled, and mean loaded parcels are displayed. The baseline (red dotted line) represents the results of the myopic ($\Theta = 0$) simulation. Results represent average values obtained from simulations on 5 realistic data instances.

O

The Difference in Performance KPI's between Anticipatory & Non-Anticipatory Simulations across each Spatial-Temporal Demand Distribution

Figure O.1 highlights the service metrics comparison between anticipatory and myopic simulations for all instances of a distribution type. Key insights can be drawn on how different demand distribution types impact the performance of the anticipatory techniques.

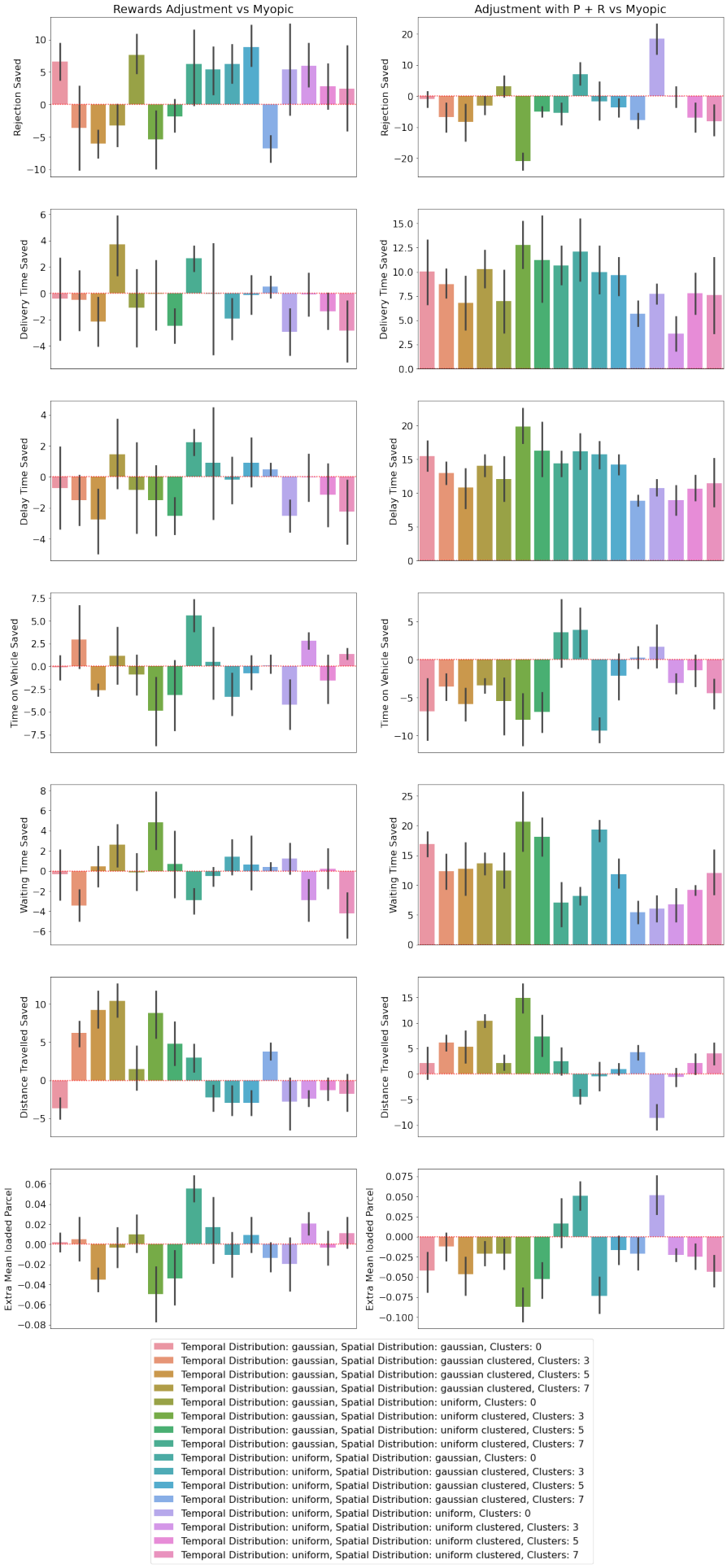


Figure O.1: Comparison between all service KPIs across all 5 instances of each data set with and without anticipatory methods

P

Independent Simulation Results

P.1. Preliminary Data Instances

Table P.1: Preliminary simulation results for various rewards adjustment techniques and the myopic simulation

S.no	Simulation Type	Overall Cost	Service Rate [%]	Orders Delivered	Delivery Time [s]	Delay Time [s]	Time On Vehicle [s]	Waiting Time [s]	Distance Travelled [Km]
0	Myopic	284664.47	82.67	124	488.07	396.52	253.19	234.88	159.93
1	RA - Orders Generated at Last Node	274784.8	83.33	125	485.34	395.26	252.66	232.68	161.6
2	RA - Candidates at Last Depot	305183.26	81.33	122	502.86	411.45	243.31	259.55	169.52
3	RA - Average of the Candidates at x Closest Depots from Last Node	243834.4	85.33	128	461.66	371.12	230.78	230.88	165.8
4	RA - Maximum of the Candidates at x Closest Depots from Last Node	265284.09	84.0	126	490.06	399.99	238.59	251.47	169.17
5	RA - Candidates at Closest Depot from Last Node	264522.54	84.0	126	477.74	387.93	227.35	250.39	166.09
6	RA - Distance of Closest Depot to Last Node	254654.14	84.67	127	477.78	386.97	241.77	236.01	163.28
7	RA - Average Distance of x Closest Depots to Last Node	234935.19	86.0	129	474.78	385.39	241.37	233.4	155.38
8	RA - Convex Combination of Candidates and Distance of x Closest Depots to Last Node	244897.14	85.33	128	477.16	387.76	237.09	240.06	161.27
9	PA - 200 seconds Look ahead	184607.13	89.33	134	457.79	366.14	303.22	154.57	151.26

Table P.2: Preliminary simulation results for various adjustments with penalization & rewards and the myopic simulation

S.no	Simulation Type	Overall Cost	Service Rate [%]	Orders Delivered	Delivery Time [s]	Delay Time [s]	Time On Vehicle [s]	Waiting Time [s]	Distance Travelled [Km]
0	Myopic	430183.52	72.67	109	465.24	368.94	260.38	204.86	152.05
1	Candidate Based Penalization: Penalization with Constant Term	410549.66	74.0	111	466.91	368.93	264.72	202.19	148.33
2	Candidate Based Penalization: Penalization with Travel Time Metric	410549.66	74.0	111	466.91	368.93	264.72	202.19	148.33
3	Adjustment with Candidate Based Penalty + Reward	390453.72	75.33	113	454.27	360.68	253.9	200.37	150.44
4	Zone Based Adjustment (P+R)	400936.92	74.67	112	471.79	372.47	261.79	210.0	156.84
5	PA - 200 seconds Look ahead	369389.92	76.67	115	434.35	335.92	289.0	145.35	148.83

P.2. Realistic Data Instances

Table P.3: Simulation results for anticipatory techniques and the myopic simulation for independent pseudo-realistic instances

Instance	Simulation Type	Overall Cost	Service Rate [%]	Orders Delivered	Delivery Time Saved [s]	Delay Time Saved [s]	Time on Vehicle Saved [s]	Waiting Time Saved [s]	Distance Travelled Saved [Km]
1	Myopic	12833789.38	59.26	1814	494.62	399.98	280.02	214.6	2010.76
	Rewards Adjustment	12795192.09	59.39	1818	495.48	400.65	276.75	218.72	2011.18
	Adjustment with Penalty + Reward	12844721.32	59.26	1814	505.49	412.02	278.9	226.58	2029.65
	Perfect Anticipation - 200s Look ahead	11980525.7	62.07	1900	484.93	388.99	308.43	176.51	1963.39
2	Myopic	6617262.5	72.02	1619	500.57	402.99	275.27	225.3	2092.01
	Rewards Adjustment	6546661.44	72.33	1626	498.43	400.51	272.81	225.62	2085.88
	Adjustment with Penalty + Reward	6737050.18	71.49	1607	502.83	405.72	274.07	228.75	2103.37
	Perfect Anticipation - 200s Look ahead	5833765.87	75.44	1696	466.22	368.81	298.82	167.39	2025.74
3	Myopic	3415343.66	80.12	1282	468.79	365.6	255.22	213.56	1992.31
	Rewards Adjustment	3263981.53	81.06	1297	461.98	359.25	248.4	213.58	2016.06
	Adjustment with Penalty + Reward	3380260.66	80.38	1286	475.44	372.07	257.6	217.84	2034.32
	Perfect Anticipation - 200s Look ahead	2827138.92	83.62	1338	410.99	308.19	270.69	140.31	1919.85
4	Myopic	16946169.06	51.18	1740	494.62	396.77	268.46	226.15	1965.12
	Rewards Adjustment	17134500.16	50.62	1721	497.77	399.19	266.93	230.83	1991.33
	Adjustment with Penalty + Reward	17286438.41	50.18	1706	500.42	404.98	267.78	232.64	1987.82
	Perfect Anticipation - 200s Look ahead	16993453.15	50.97	1733	470.44	372.15	286.38	184.05	1970.3
5	Myopic	19388284.48	47.08	1695	494.91	398.13	273.16	221.75	1744.96
	Rewards Adjustment	19621705.27	46.42	1671	494.07	395.96	273.87	220.2	1761.55
	Adjustment with Penalty + Reward	19819392.75	45.89	1652	505.54	409.82	273.42	232.12	1767.5
	Perfect Anticipation - 200s Look ahead	19223389.03	47.47	1709	462	365.74	293.67	168.33	1727.06

P.3. 80 Data Instances

Table P.4: Results for Independent Simulations

Instance	Simulation Type	Overall Cost	Service Rate [%]	Orders Delivered	Delivery Time [s]	Delay Time [s]	Time On Vehicle [s]	Waiting Time [s]	Distance Travelled [Km]
1	Myopic	18633664.235	25.96	649	473.073959938367	380.090909090909	264.305084745763	208.768875192604	649.47
	Rewards Adjustment	18534251.08	26.36	659	468.911987860395	376.115326251897	266.488619119879	202.423368740516	642.16
	Adjustment with Penalty & Rewards	18708471.91	25.68	642	488.688473520249	399.208722741433	262.714953271028	225.973520249221	651.82
2	Myopic	18594802.08	26.12	653	487.407350689127	381.211332312404	276.08269525268	211.324655436447	673.16
	Rewards Adjustment	18535983.225	26.36	659	487.1593323217	381.332321699545	278.200303490137	208.959028831563	668.45
	Adjustment with Penalty & Rewards	18539591.645	26.36	659	491.468892261002	392.286798179059	268.748103186646	222.720789074355	666.29
3	Myopic	18505604.195	26.48	662	473.601208459214	378.486404833837	271.980362537764	201.62084592145	650.39
	Rewards Adjustment	18479221.185	26.6	665	484.968421052632	387.666165413534	273.593984962406	211.374436090226	644.37
	Adjustment with Penalty & Rewards	18462285.035	26.68	667	486.064467766117	395.685157421289	258.697151424288	227.367316341829	648.07
4	Myopic	18300549.495	27.32	683	472.94729136164	381.341142020498	257.836017569546	215.111273792094	642.99
	Rewards Adjustment	18134223.005	28	700	473.695714285714	382.578571428571	256.247142857143	217.448571428571	641.01
	Adjustment with Penalty & Rewards	18366841.12	27.08	677	492.283604135894	403.288035450517	266.035450516987	226.248153618907	656.24
5	Myopic	18176798.685	27.84	696	485.367816091954	392.155172413793	266.294540229885	219.073275862069	657.37
	Rewards Adjustment	18202703.05	27.72	693	475.532467532468	382.030303030303	261.357864357864	214.174603174603	659.1
	Adjustment with Penalty & Rewards	18209012.89	27.72	693	484.092352092352	400.236652236652	246.099567099567	237.992784992785	661.78
6	Myopic	17879135.475	29.04	726	470.643250688705	382.429752066116	241.409090909091	229.234159779614	626.95
	Rewards Adjustment	17869596.72	29.08	727	473.317744154058	383.163686382393	249.403026134801	223.914718019257	633.44
	Adjustment with Penalty & Rewards	18100451.88	28.16	704	483.551136363636	398.109375	240.997159090909	242.553977272727	634.76
7	Myopic	17812664.12	29.32	733	476.113233287858	388.416098226467	253.778990450205	222.334242837653	619.24
	Rewards Adjustment	17800191.145	29.36	734	467.453678474115	381.141689373297	242.797002724796	224.656675749319	624.29
	Adjustment with Penalty & Rewards	17936471.995	28.84	721	489.73092926491	405.44105409154	243.231622746186	246.499306518724	620.99
8	Myopic	18592202.045	26.12	653	478.37212863706	373.275650842266	258.408882082695	219.963246554364	655.09
	Rewards Adjustment	18582358.875	26.16	654	478.620795107034	373.178899082569	260.793577981651	217.827217125382	658.75
	Adjustment with Penalty & Rewards	18486378.36	26.56	664	481.564759036145	379.665662650602	256.103915662651	225.460843373494	658.72
9	Myopic	17595747.54	30.2	755	462.419867549669	385.331125827815	220.531125827815	241.888741721854	570.08
	Rewards Adjustment	17900812.36	28.96	724	469.385359116022	388.17955801105	229.660220994475	239.725138121547	582.72
	Adjustment with Penalty & Rewards	17698092.21	29.8	745	469.465771812081	396.781208053691	218.445637583893	251.020134228188	582.42
10	Myopic	18557642.705	26.28	657	487.613394216134	387.566210045662	255.803652968037	231.809741248097	654.41
	Rewards Adjustment	18458354.805	26.68	667	483.850074962519	383.886056971514	261.866566716642	221.983508245877	657.61
	Adjustment with Penalty & Rewards	18572153.865	26.24	656	494.545731707317	401.90243902439	253.253048780488	241.292682926829	659.73
11	Myopic	18430478.72	26.8	670	504.341791044776	388.444776119403	295.153731343284	209.188059701493	699.44
	Rewards Adjustment	18469008.65	26.64	666	501.953453453453	386.36036036036	294.186186186186	207.767267267267	701.3
	Adjustment with Penalty & Rewards	18461443.9	26.68	667	503.155922038981	393.07796101949	288.52323838081	214.632683658171	704.8
12	Myopic	18499250.085	26.52	663	497.579185520362	388.855203619909	275.303167420814	222.276018099548	689.17
	Rewards Adjustment	18516734.885	26.44	661	491.465960665658	382.402420574887	271.904689863843	219.561270801815	701.77
	Adjustment with Penalty & Rewards	18482383.815	26.6	665	501.317293233083	397.114285714286	275	226.317293233083	686.63
13	Myopic	18505403.08	26.48	662	479.756797583082	377.844410876133	261.555891238671	218.200906344411	673.16
	Rewards Adjustment	18649316.27	25.88	647	473.383307573416	367.782071097372	256.503863987635	216.879443585781	677.54
	Adjustment with Penalty & Rewards	18775882.675	25.4	635	492.233070866142	395.4	248.236220472441	243.996850393701	686.35
14	Myopic	17988698.125	28.6	715	469.632167832168	387.093706293706	240.304895104895	229.327272727273	624.25
	Rewards Adjustment	17998403.505	28.56	714	467.182072829132	386.795518207283	238.606442577031	228.575630252101	635.01
	Adjustment with Penalty & Rewards	18211290.04	27.72	693	485.454545454545	406.844155844156	230.487734487734	254.966810966811	637.08
15	Myopic	18812177.095	25.24	631	497.892234548336	386.188589540412	287.16323296355	210.729001584786	669.19
	Rewards Adjustment	18892248.405	24.92	623	504.46709470305	391.349919743178	285.274478330658	219.192616372392	685.81
	Adjustment with Penalty & Rewards	18735017.44	25.56	639	501.075117370892	390.247261345853	287.727699530516	213.347417840376	666.88
16	Myopic	18350699.395	27.12	678	480.053097345133	384.600294985251	265.72418879056	214.328908554572	639.79
	Rewards Adjustment	18426939.62	26.8	670	474.383582089552	377.964179104478	256.858208955224	217.525373134328	643.24
	Adjustment with Penalty & Rewards	18435856.24	26.8	670	495.888059701493	404.573134328358	261.553731343284	234.334328358209	648.48
17	Myopic	18596311.15	26.12	653	486.595712098009	385.83460949464	271.828483920368	214.767228177642	672.3
	Rewards Adjustment	18588877.52	26.16	654	497.816513761468	393.081039755352	282.249235474006	215.567278287462	680.04
	Adjustment with Penalty & Rewards	18521317.3	26.44	661	495.553706505295	396.295007564296	269.590015128593	225.963691376702	683.6
18	Myopic	18381584.63	27	675	491.425185185185	388.928888888889	275.518518518519	215.906666666667	642.26
	Rewards Adjustment	18371428.14	27.04	676	493.835798816568	387.866863905325	276.741124260355	217.094674556213	658.28
	Adjustment with Penalty & Rewards	18367760.435	27.08	677	504.540620384047	406.002954209749	275.546528803545	228.994091580502	656.87
19	Myopic	18566366.23	26.24	656	488.40243902439	384.237804878049	276.792682926829	211.609756097561	672.46
	Rewards Adjustment	18714075.155	25.64	641	494.572542901716	386.05616224649	271.88767550702	222.684867394696	688.31
	Adjustment with Penalty & Rewards	18619878.285	26.04	651	499.142857142857	397.967741935484	271.121351766513	228.021505376344	679.57
20	Myopic	18694644.53	25.72	643	496.804043545879	386.662519440124	274.940902021773	221.863141524106	665.06
	Rewards Adjustment	18647545.43	25.92	648	501.319444444444	392.618827160494	282.79012345679	218.529320987654	673.86
	Adjustment with Penalty & Rewards	18815224.795	25.24	631	499.610142630745	395.838351822504	270.112519809826	229.497622820919	675.59
21	Myopic	18409291.825	26.88	672	478.717261904762	383.815476190476	264.916666666667	213.800595238095	659.65
	Rewards Adjustment	18301487.105	27.32	683	479.910688140556	384.055636896047	268.002928257687	211.90775988287	664.21
	Adjustment with Penalty & Rewards	18512519.725	26.48	662	492.561933534743	399.358006042296	274.35498489426	218.206948640483	664.45
22	Myopic	18535781.585	26.36	659	484.039453717754	380.734446130501	278.742033383915	205.297420333839	659.17
	Rewards Adjustment	18477975.545	26.6	665	486.44962406015	383.885714285714	281.795488721805	204.654135338346	667.09
	Adjustment with Penalty & Rewards	18405270.215	26.92	673	498.702823179792	401.001485884101	277.832095096582	220.87072808321	666.43
23	Myopic	18468464.82	26.64	666	491.492492492493	384.756756756757	290.558558558559	200.933933933934	681.64
	Rewards Adjustment	18352352.25	27.12	678	494.620943952802	389.430678466077	286.072271386431	208.548672566372	670.5
	Adjustment with Penalty & Rewards	18415254.025	26.88	672	500.831845238095	401.528273809524	275.534226190476	225.297619047619	681.05
24	Myopic	18489531.715	26.56	664	490.98343373494	389.138554216867	279.579819277108	211.403614457831	675.43
	Rewards Adjustment	18352504.955	27.12	678	491.542772861357	389.873156342183	281.740412979351	209.802359882006	675.91
	Adjustment with Penalty & Rewards	18502204.25	26.52	663	494.458521870287	397.785822021116	274.805429864253	219.653092006033	676.5
25	Myopic	18432830.72	26.8	670	495.516417910448	395.498507462687	287.292537313433	208.223880597015	677.44
	Rewards Adjustment	18477500.03	26.6	665	482.741353383459	382.431578947368	278.828571428571	203.912781954887	683.06
	Adjustment with Penalty & Rewards	18363982.45	27.08	677	489.234859675037	394.81388478582	271.155096011817	218.07976366322	675.9

Table P.4: Results for Independent Simulations

Instance	Simulation Type	Overall Cost	Service Rate	Orders Delivered	Delivery Time	Delay Time	Time On Vehicle	Waiting Time	Distance Travelled
26	Myopic	18185061.555	27.8	695	486.873381294964	387.748201438849	269.451798561151	217.421582733813	638.11
	Rewards Adjustment	18155941.105	27.92	698	491.070200573066	388.593123209169	271.0888252149	219.981375358166	644.21
	Adjustment with Penalty & Rewards	18375944.165	27.04	676	494.334319526627	401.230769230769	274.133136094675	220.201183431953	656.33
27	Myopic	18507641.005	26.48	662	482.235649546828	384.613293051359	258.270392749245	223.965256797583	668.01
	Rewards Adjustment	18533522.075	26.36	659	471.775417298938	373.860394537178	250.241274658574	221.534142640364	670.15
	Adjustment with Penalty & Rewards	18717500.125	25.64	641	487.294851794072	396.767550702028	246.563182527301	240.731669266771	672.25
28	Myopic	18094690.285	28.16	704	478.302556818182	381.772727272727	265.997159090909	212.305397727273	612.57
	Rewards Adjustment	18153095.61	27.92	698	475.994269340974	380.462750716332	248.097421203438	227.896848137536	628.22
	Adjustment with Penalty & Rewards	18377107.6	27.04	676	492.936390532544	404.705621301775	247.430473372781	245.505917159763	634.2
29	Myopic	18005996.435	28.52	713	469.697054698457	380.589060308555	254.201963534362	215.495091164095	632.87
	Rewards Adjustment	18251402.95	27.52	688	472.861918604651	381.039244186047	248.375	224.486918604651	650.9
	Adjustment with Penalty & Rewards	18122180.77	28.08	702	485.74358974359	404.155270655271	241.940170940171	243.803418803419	644.54
30	Myopic	17709058.07	29.72	743	445.635262449529	373.519515477793	222.331090174966	223.304172274563	591.14
	Rewards Adjustment	17671004.105	29.88	747	450.899598393574	376.726907630522	228.041499330656	222.858099062918	593.21
	Adjustment with Penalty & Rewards	17973919.875	28.68	717	466.311018131102	400.599721059972	220.536959553696	245.774058577406	609.75
31	Myopic	18369381.645	27.04	676	484.686390532544	381.829881656805	266.973372781065	217.713017751479	646.29
	Rewards Adjustment	18290219.39	27.36	684	479.997076023392	379.81432748538	265.223684210526	214.773391812865	645.78
	Adjustment with Penalty & Rewards	18436051.9	26.8	670	503.114925373134	405.153731343284	258.835820895522	244.279104477612	650.8
32	Myopic	18122756.07	28.04	701	467.743223965763	377.851640513552	245.807417974322	221.935805991441	638.14
	Rewards Adjustment	18182432.16	27.8	695	469.490647482014	380.168345323741	249.653237410072	219.837410071942	647.32
	Adjustment with Penalty & Rewards	18142956.73	28	700	493.382857142857	407.525714285714	249.34	244.042857142857	645.46
33	Myopic	18324638.99	27.24	681	492.945668135095	394.477239353891	267.499265785609	225.446402349486	638.98
	Rewards Adjustment	18343500.44	27.16	679	491.692194403535	392.293078055965	275.59499263623	216.097201767305	633.88
	Adjustment with Penalty & Rewards	18348014.695	27.16	679	495.930780559647	405.589101620029	263.017673048601	232.913107511046	634.39
34	Myopic	18154951.395	27.92	698	480.606017191977	385.763610315186	268.736389684814	211.869627507163	639.79
	Rewards Adjustment	18193166.125	27.76	694	479.14409221902	382.829971181556	253.893371757925	225.250720461095	648.25
	Adjustment with Penalty & Rewards	18267880.265	27.48	687	492.459970887918	400.43231441048	257.183406113537	235.276564774381	663.53
35	Myopic	18019697.175	28.48	712	483.181179775281	391.511235955056	263.575842696629	219.605337078652	638.35
	Rewards Adjustment	18065952.09	28.28	707	476.56152758133	383.669024045262	252.513437057992	224.048090523338	650.18
	Adjustment with Penalty & Rewards	18069632.1	28.28	707	480.312588401697	394.087694483734	249.623762376238	230.68882602546	644.2
36	Myopic	18436377.255	26.76	669	471.715994020927	376.828101644245	263.823617339312	207.892376681614	656.51
	Rewards Adjustment	18221505.105	27.64	691	471.84515195369	379.678726483357	261.635311143271	210.20984081042	652.21
	Adjustment with Penalty & Rewards	18569749.855	26.24	656	485.91006097561	394.560975609756	262.618902439024	223.291158536585	667.71
37	Myopic	18460367.125	26.68	667	495.670164917541	389.932533733133	282.269865067466	213.400299850075	649.25
	Rewards Adjustment	18617376.96	26.04	651	499.646697388633	390.322580645161	288.637480798771	211.009216589862	653.92
	Adjustment with Penalty & Rewards	18591234.695	26.16	654	500.683486238532	400.325688073395	268.640672782875	232.042813455657	656.39
38	Myopic	18750228.705	25.48	637	480.568288854003	376.431711145997	269.824175824176	210.744113029827	670.41
	Rewards Adjustment	18584936.02	26.16	654	486.403669724771	381.038226299694	281.14373088685	205.25993883792	673.04
	Adjustment with Penalty & Rewards	18707536.135	25.68	642	496.154205607477	396.274143302181	281.766355140187	214.38785046729	664.27
39	Myopic	18361491.905	27.08	677	483.69423929099	387.534711964549	263.648449039882	220.045790251108	622.81
	Rewards Adjustment	18351563.755	27.12	678	483.88790560472	387.163716814159	269.520648967552	214.367256637168	630.51
	Adjustment with Penalty & Rewards	18393814.66	26.96	674	488.856083086053	396.154302670623	269.786350148368	219.069732937685	621.32
40	Myopic	18468958.27	26.64	666	482.507507507508	386.31981981982	256.677177177177	225.83033033033	627.54
	Rewards Adjustment	18401552.865	26.92	673	485.560178306092	390.005943536404	263.144130757801	222.416047548291	631.73
	Adjustment with Penalty & Rewards	18503476.65	26.52	663	495.844645550528	401.69532428356	271.466063348416	224.378582202112	629.3
41	Myopic	16753268.31	33.64	841	489.67063020214	387.38049940547	281.843043995244	207.827586206897	749.62
	Rewards Adjustment	16564161.325	34.44	861	504.507549361208	403.693379790941	296.47619047619	208.031358885017	742.65
	Adjustment with Penalty & Rewards	16608094.175	34.28	857	512.471411901984	414.757292882147	293.742123687281	218.729288214702	741.35
42	Myopic	16466144.6	34.84	871	502.708381171068	403.623421354765	288.107921928817	214.60045924225	733.2
	Rewards Adjustment	16506773.1	34.68	867	507.565167243368	406.930795847751	292.559400230681	215.005767012687	737.2
	Adjustment with Penalty & Rewards	16463588.74	34.88	872	515.05504587156	420.23623853211	292.307339449541	222.747706422018	731.48
43	Myopic	16593823.785	34.32	858	501.782051282051	404.320512820513	289.412587412587	212.369463869464	740.57
	Rewards Adjustment	16522176.76	34.6	865	491.891329479769	397.243930635838	281.743352601156	210.147976878613	737.52
	Adjustment with Penalty & Rewards	16450951.985	34.92	873	505.084765177549	413.711340206186	287.437571592211	217.647193585338	733.97
44	Myopic	16331178.82	35.3741496598639	884	503.846153846154	409.080316742081	286.772624434389	217.073529411765	730.64
	Rewards Adjustment	16318481.48	35.4141656662665	885	494.81129435028	402.523163841808	275.28813559322	219.523163841808	729.96
	Adjustment with Penalty & Rewards	16257714.715	35.6942777110844	892	509.440582959641	420.072869955157	279.743273542601	229.69730941704	724.43
45	Myopic	16444836.3	34.92	873	492.537227949599	399.70332187858	282.995418098511	209.541809851088	731.6
	Rewards Adjustment	16414823.01	35.04	876	491.62899543379	398.309360730594	285.570776255708	206.058219178082	727.02
	Adjustment with Penalty & Rewards	16501058.11	34.72	868	508.94470046083	416.341013824885	295.278801843318	213.665898617511	732.22
46	Myopic	15643640.75	38.2	955	497.815706806283	404.797905759162	282.794764397906	215.020942408377	699.5
	Rewards Adjustment	15485944.27	38.84	971	492.125643666323	402.880535530381	274.332646755922	217.792996910402	691.54
	Adjustment with Penalty & Rewards	15630473.76	38.28	957	506.345872518286	418.237199582027	276.038662486938	230.307210031348	694.52
47	Myopic	15680903.185	38.04	951	500.166140904311	400.745531019979	273.40483701367	226.761303890641	697.37
	Rewards Adjustment	15624836.585	38.28	957	503.76802507837	406.455590386625	272.942528735632	230.825496342738	695.17
	Adjustment with Penalty & Rewards	15925741.745	37.08	927	515.077669902913	421.549083063646	262.142394822006	252.935275080906	707.49
48	Myopic	16246212.315	35.76	894	523.475391498881	415.771812080537	271.808724832215	251.666666666667	724.63
	Rewards Adjustment	16283948.54	35.6	890	519.733707865169	412.548314606742	269.165168539326	250.568539325843	729.08
	Adjustment with Penalty & Rewards	16250926.93	35.76	894	525.064876957494	426.321029082774	258.623042505593	266.441834451902	722.86
49	Myopic	15772222.115	37.68	942	510.815286624204	407.366242038217	297.831210191083	212.984076433121	705.23
	Rewards Adjustment	15731883.895	37.84	946	506.815010570825	404.931289640592	287.698731501057	219.116279069767	702.79
	Adjustment with Penalty & Rewards	15856288.8	37.36	934	517.798715203426	419.561027837259	285.845824411135	231.952890792291	707.6
50	Myopic	16218148.035	35.84	896	493.487723214286	396.845982142857	283.818080357143	209.669642857143	722.07
	Rewards Adjustment	16130398.735	36.2	905	493.64861878453	397.880662983425	288.850828729282	204.797790055249	715.47
	Adjustment with Penalty & Rewards	16022200.515	36.68	917	511.332606324973	418.415485278081	280.123227917121	231.209378407852	714.03
51	Myopic	16220171.905	35.8143257302921	895	509.148603351955	401.806703910615	304.867039106145	204.28156424581	726.81
	Rewards Adjustment	16183895.79	35.9743897559024	899	513.563959955506	408.304783092325	298.735261401557	214.828698553949	725.58
	Adjustment with Penalty & Rewards	16157895.355	36.09443777511	902	520.347006651885	415.815964523282	303.09866962306	217.248337028825	724.71
52	Myopic	16126518.835	36.24	906	507.019867549669	410.955849889625	279.318984547461	227.700883002207	711.67
	Rewards Adjustment	15969117.625	36.88	922	503.460954446855	409.467462039046	276.280911062907	227.180043383948	706.25
	Adjustment with Penalty & Rewards	16144031.325	36.2	905	516.764640883978	428.009944751381	270.114917127072	246.649723756906	713.65
53	Myopic	16484761.745	34.76	869	514.200230149597	401.364787111623	306.318757192175	207.881472957422	737.49
	Rewards Adjustment	16417194.07	35.04	876	514.88698630137	403.711187214612	304.083333333333	210.803652968037	737.14
	Adjustment with Penalty & Rewards	16577161.65	34.4	860	519.201162790698	411.147674418605	307.904651162791	211.296511627907	736.3
54	Myopic	15958422.045	36.92	923	506.199349945829	407.51570964247	286.191765980498	220.00758396533	707.09
	Rewards Adjustment	15779891.435	37.64	941	501.510095642933	402.853347502657	289.638682252922	211.871413390011	697.87
	Adjustment with Penalty & Rewards	16002822.565	36.76	919	511.54189336235	418.861806311208	276.21436343852	235.32752992383	711.13

Table P.4: Results for Independent Simulations

Instance	Simulation Type	Overall Cost	Service Rate	Orders Delivered	Delivery Time	Delay Time	Time On Vehicle	Waiting Time	Distance Travelled
55	Myopic	16316725.08	35.4141656662665	885	500.811299435028	398.559322033898	290.835028248588	209.976271186441	725.16
	Rewards Adjustment	16327377.795	35.3741496598639	884	503.343891402715	400.485294117647	295.02149321267	208.322398190045	726.59
	Adjustment with Penalty & Rewards	16432847.05	34.9739895958383	874	517.747139588101	417.5823798627	299.400457665904	218.346681922197	727.1
56	Myopic	16388592.995	35.1340536214486	878	510.9715261959	405.980637813212	303.800683371298	207.170842824601	734.99
	Rewards Adjustment	16525756.835	34.5738295318127	864	513.552083333333	405.986111111111	305.949074074074	207.603009259259	741.67
	Adjustment with Penalty & Rewards	16402521.48	35.094037615046	877	518.638540478905	415.403648802737	307.574686431015	211.063854047891	733.96
57	Myopic	16376494.635	35.2	880	502.45	400.295454545455	295.418181818182	207.031818181818	729.27
	Rewards Adjustment	16465058.32	34.84	871	501.916188289323	401.126291618829	295.559127439724	206.357060849598	735.64
	Adjustment with Penalty & Rewards	16508257.145	34.68	867	509.41291810842	410.356401384083	299.22952710496	210.18339100346	735.29
58	Myopic	16457111.435	34.88	872	514.792431192661	405.376146788991	303.525229357798	211.267201834862	734.87
	Rewards Adjustment	16526324.565	34.6	865	516.679768786127	406.834682080925	305.721387283237	210.95838150289	737.13
	Adjustment with Penalty & Rewards	16479130.47	34.8	870	515.252873563218	410.945977011494	301.742528735632	213.510344827586	737.94
59	Myopic	16494243.2	34.6938775510204	867	503.525951557093	401.099192618224	296.123414071511	207.402537485582	733.4
	Rewards Adjustment	16532993.415	34.5338135254102	863	501.008111239861	400.061413673233	292.276940903824	208.731170336037	733.83
	Adjustment with Penalty & Rewards	16596124.295	34.2937174869948	857	508.690781796966	410.164527421237	296.30805134189	212.382730455076	737.59
60	Myopic	16271621.32	35.64	891	511.244668911336	406.86531986532	301.019079685746	210.225589225589	725.64
	Rewards Adjustment	16272155.86	35.64	891	512.294051627385	408.061728395062	300.837261503928	211.456790123457	728.72
	Adjustment with Penalty & Rewards	16403510.74	35.12	878	519.211845102506	417.183371298405	296.126423690205	223.085421412301	734.48
61	Myopic	16407311.285	35.08	877	496.744583808438	403.524515393387	289.689851767389	207.054732041049	731.57
	Rewards Adjustment	16140122.3	36.16	904	488.375	397.70685840708	283.438053097345	204.936946902655	717.6
	Adjustment with Penalty & Rewards	16059640.955	36.52	913	503.167579408543	414.641840087623	289.46878422782	213.698795180723	713.91
62	Myopic	16231921.75	35.8	895	504.891620111732	405.71843575419	303.663687150838	201.227932960894	725.5
	Rewards Adjustment	16474859.31	34.8	870	500.405747126437	401.131034482759	297.001149425287	203.404597701149	734.62
	Adjustment with Penalty & Rewards	16061033.115	36.52	913	512.780941949617	417.688937568456	296.392113910186	216.38882803943	716.23
63	Myopic	16429221.46	35	875	513.384	408.812571428571	303.329142857143	210.054857142857	731.92
	Rewards Adjustment	16331516.75	35.4	885	516.650847457627	409.381920903955	305.315254237288	211.335593220339	730.5
	Adjustment with Penalty & Rewards	16363966.795	35.28	882	521.72335600907	416.327664399093	314.380952380952	207.342403628118	732.59
64	Myopic	16574011.425	34.3737494997999	859	507.514551804424	404.286379511059	301.998835855646	205.515715948778	740.85
	Rewards Adjustment	16494851.315	34.6938775510204	867	502.20184544406	402.500576701269	290.035755478662	212.166089965398	734.63
	Adjustment with Penalty & Rewards	16499200.97	34.6938775510204	867	512.118800461361	412.535178777393	299.79815455594	212.320645905421	733.94
65	Myopic	16356629.925	35.28	882	493.683673469388	399.692743764172	282.630385487528	211.053287981859	730.85
	Rewards Adjustment	16287599.735	35.56	889	493.946006749156	398.728908886389	284.610798650169	209.335208098988	729.47
	Adjustment with Penalty & Rewards	16137962.34	36.2	905	505.116022099447	414.590055248619	289.534806629834	215.581215469613	720.68
66	Myopic	15789676.95	37.6	940	491.773404255319	402.821276595745	273.025531914894	218.747872340426	701.9
	Rewards Adjustment	15602026.085	38.36	959	489.843587069865	399.745568300313	278.713242961418	211.130344108446	696.17
	Adjustment with Penalty & Rewards	15667208.905	38.12	953	495.948583420777	413.139559286464	265.305351521511	230.643231899265	695.81
67	Myopic	16309934.78	35.48	887	507.727170236753	404.895152198422	265.314543404735	242.412626832018	727.56
	Rewards Adjustment	16279728.16	35.6	890	506.960674157303	403.065168539326	266.759550561798	240.201123595506	728.32
	Adjustment with Penalty & Rewards	16334236.525	35.4	885	510.166101694915	415.531073446328	264.962711864407	245.203389830508	728.05
68	Myopic	16072072.85	36.44	911	508.371020856202	398.931942919868	291.386388583974	216.984632272228	718.7
	Rewards Adjustment	16092905.325	36.36	909	509.117711771177	401.645764576458	291.193619361936	217.924092409241	714.65
	Adjustment with Penalty & Rewards	16156970.485	36.12	903	517.619047619048	413.31007751938	290.148394241418	227.47065337763	721.97
69	Myopic	16241823.835	35.734293717487	893	508.808510638298	406.406494960806	271.297872340426	237.510638297872	726.67
	Rewards Adjustment	16161901.425	36.0544217687075	901	504.249722530522	402.972253052164	274.882352941176	229.367369589345	724.85
	Adjustment with Penalty & Rewards	16310378.12	35.4541816726691	886	505.075620767494	406.351015801354	270.3769751693	234.698645598194	729.24
70	Myopic	16287939.095	35.56	889	501.980877390326	399.492688413948	291.402699662542	210.578177727784	729.19
	Rewards Adjustment	16270760.59	35.64	891	508.112233445567	404.930415263749	295.037037037037	213.07519640853	728.18
	Adjustment with Penalty & Rewards	16262710.495	35.68	892	508.008968609865	408.850896860987	286.137892376682	221.871076233184	725.99
71	Myopic	15711361.995	37.92	948	488.061181434599	402.978902953587	276.086497890295	211.974683544304	699.99
	Rewards Adjustment	15602948.565	38.36	959	486.980187695516	401.666319082378	276.464025026069	210.516162669447	699.13
	Adjustment with Penalty & Rewards	15676857.66	38.08	952	496.013655462185	412.832983193277	277.955882352941	218.057773109244	698.32
72	Myopic	16074923.23	36.44	911	497.097694840834	405.190998902305	286.402854006586	210.694840834248	717.46
	Rewards Adjustment	15943364.705	36.96	924	489.470779220779	396.124458874459	280.977272727273	208.493506493506	710.41
	Adjustment with Penalty & Rewards	16021106.12	36.68	917	505.016357688113	416.028353326063	284.384950926936	220.631406761178	714.24
73	Myopic	16122160.345	36.24	906	507.040838852097	401.326710816777	303.451434878587	203.58940397351	718.69
	Rewards Adjustment	16122068.745	36.24	906	505.363134657837	401.122516556291	293.466887417219	211.896247240618	720.49
	Adjustment with Penalty & Rewards	16162469.76	36.08	902	506.716186252772	403.789356984479	294.70509977827	212.011086474501	721.52
74	Myopic	16189844	35.96	899	501.776418242492	399.291434927697	287.129032258064	214.647385984427	725
	Rewards Adjustment	16219701.345	35.84	896	500.353794642857	400.3125	285.760044642857	214.59375	722.69
	Adjustment with Penalty & Rewards	16451027.74	34.92	873	511.958762886598	413.8843069874	285.281786941581	226.676975945017	734.48
75	Myopic	15934057.675	37	925	497.086486486487	397.191351351351	282.652972972973	214.433513513513	713.35
	Rewards Adjustment	16004374.21	36.72	918	502.095860566449	400.907407407407	291.16339869281	210.932461873638	715.42
	Adjustment with Penalty & Rewards	16088215.375	36.4	910	510.22967032967	412.872527472528	286.465934065934	223.763736263736	716.75
76	Myopic	16166816.2	36.0544217687075	901	513.997780244173	413.885682574917	295.334073251942	218.663706992231	721.4
	Rewards Adjustment	16055895.33	36.4945978391357	912	505.269736842105	406.88048245614	296.845394736842	208.424342105263	715.66
	Adjustment with Penalty & Rewards	16419722.305	35.0140056022409	875	507.530285714286	409.96	287.611428571429	219.918857142857	729.61
77	Myopic	16064339.06	36.48	912	502.899122807018	403.462719298246	294.048245614035	208.850877192982	720.12
	Rewards Adjustment	15945283.68	36.96	924	498.668831168831	400.275974025974	294.672077922078	203.996753246753	712.36
	Adjustment with Penalty & Rewards	16010961.045	36.72	918	509.920479302832	415.254901960784	291.114379084967	218.806100217865	718.09
78	Myopic	16085912.635	36.4	910	508.492307692308	407.812087912088	292.268131868132	216.224175824176	716.27
	Rewards Adjustment	16170814.35	36.04	901	500.795782463929	400.562708102109	293.482796892342	207.312985571587	721.7
	Adjustment with Penalty & Rewards	16247294.175	35.76	894	513.482102908277	418.190156599553	285.804250559284	227.677852348993	726.35
79	Myopic	15907651.165	37.12	928	502.504310344828	403.658405172414	274.922413793103	227.581896551724	707.33
	Rewards Adjustment	15723030.845	37.88	947	505.525871172123	406.927138331573	274.88489968321	230.640971488912	701.69
	Adjustment with Penalty & Rewards	15909441.695	37.16	929	521.922497308934	428.606027987083	266.616792249731	255.305705059203	708.39
80	Myopic	16357709.03	35.28	882	518.857142857143	402.138321995465	313.456916099773	205.40022675737	732.06
	Rewards Adjustment	16564749.47	34.44	861	522.39837398374	405.066202090592	317.044134727062	205.354239256678	736.94
	Adjustment with Penalty & Rewards	16422767.385	35.04	876	531.84703196347	416.438356164384	316.834474885845	215.012557077626	734.77

References

[1] Javier Alonso-Mora, Alex Wallar, and Daniela Rus. “Predictive routing for autonomous mobility-on-demand systems with ride-sharing”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 3583–3590. DOI: 10.1109/IROS.2017.8206203.

[2] Javier Alonso-Mora et al. “On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment”. In: *Proceedings of the National Academy of Sciences* 114.3 (2017), pp. 462–467. ISSN: 0027-8424. DOI: 10.1073/pnas.1611675114. eprint: https://www.pnas.org/content/114/3/462.full.pdf. URL: https://www.pnas.org/content/114/3/462.

[3] N. Azi, M. Gendreau, and J. Potvin. “A dynamic vehicle routing problem with multiple delivery routes”. In: *Annals of Operations Research* 199 (2012), pp. 103–112.

[4] Bbike. “Open Street Map Open Source Repository”. In: (). URL: http://download.bbbike.org/osm/bbbike/.

[5] Richard Bellman. “Dynamic Programming and Lagrange Multipliers”. In: *Proceedings of the National Academy of Sciences of the United States of America* 42.10 (1956), pp. 767–769. ISSN: 00278424. DOI: 10.2307/89535. URL: http://www.jstor.org/stable/89535.

[6] Russell Bent and Pascal Van Hentenryck. “Regrets Only! Online Stochastic Optimization under Time Constraints”. In: *Proceedings of the 19th National Conference on Artifical Intelligence*. AAAI’04. San Jose, California: AAAI Press, 2004, pp. 501–506. ISBN: 0262511835. DOI: 10.5555/1597148.1597230.

[7] Russell Bent and Pascal Van Hentenryck. “Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers”. In: *Operations Research* 52 (Dec. 2004), pp. 977–987. DOI: 10.1287/opre.1040.0124.

[8] Gerardo Berbeglia, Jean-François Cordeau, and Gilbert Laporte. “Dynamic pickup and delivery problems”. In: *European Journal of Operational Research* 202.1 (2010), pp. 8–15. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2009.04.024. URL: https://www.sciencedirect.com/science/article/pii/S0377221709002999.

[9] Jürgen Branke et al. “Waiting Strategies for Dynamic Vehicle Routing”. In: *Transportation Science* 39.3 (2005), pp. 298–312. DOI: 10.1287/trsc.1040.0095. eprint: https://pubsonline.informs.org/doi/pdf/10.1287/trsc.1040.0095. URL: https://pubsonline.informs.org/doi/abs/10.1287/trsc.1040.0095.

[10] Jean-François Cordeau. “A Branch-and-Cut Algorithm for the Dial-a-Ride Problem”. In: *Operations Research* 54.3 (2006), pp. 573–586. DOI: 10.1287/opre.1060.0283. eprint: https://doi.org/10.1287/opre.1060.0283. URL: https://doi.org/10.1287/opre.1060.0283.

[11] Jean-François Cordeau and Gilbert Laporte. “The dial-a-ride problem (DARP): Models and algorithms”. In: *Annals OR* 153 (June 2007), pp. 29–46. DOI: 10.1007/s10479-007-0170-8.

[12] G. B. Dantzig and J. H. Ramser. “The Truck Dispatching Problem”. In: *Management Science* 6.1 (1959), pp. 80–91. DOI: 10.1287/mnsc.6.1.80. eprint: https://doi.org/10.1287/mnsc.6.1.80. URL: https://doi.org/10.1287/mnsc.6.1.80.

[13] Edsger W. Dijkstra. “A note on two problems in connexion with graphs”. In: *Numerische Mathematik* 1 (1959), pp. 269–271.

[14] L.R Engelen. “The same-day pickup and delivery problem”. In: *Masters thesis, TU Eindhoven* (2018).

[15] Andres Fielbaum, Maximilian Kronmueller, and Javier Alonso-Mora. “Anticipatory routing methods for a ridesharing on-demand mobility system (Manuscript submitted for publication)”. en. In: *Transportation Science* (2021).

[16] Michael R Garey and David S Johnson. *Computers and intractability*. Vol. 174. freeman San Francisco, 1979.

[17] Tobias Gawor and Kai Hoberg. “Customers’ valuation of time and convenience in e-fulfillment”. In: *International Journal of Physical Distribution Logistics Management* 49 (Aug. 2018). DOI: 10.1108/IJPDLM-09-2017-0275.

[18] Michel Gendreau et al. “Parallel Tabu Search for Real-Time Vehicle Routing and Dispatching”. In: *Transportation Science* 33 (Nov. 1999), pp. 381–390. DOI: 10.1287/trsc.33.4.381.

[19] Gianpaolo Ghiani, Emanuele Manni, and Barrett Thomas. “A Comparison of Anticipatory Algorithms for the Dynamic and Stochastic Traveling Salesman Problem”. In: *Transportation Science* 46 (Aug. 2012), pp. 374–387. DOI: 10.1287/trsc.1110.0374.

[20] Gianpaolo Ghiani et al. “Anticipatory algorithms for same-day courier dispatching”. In: *Transportation Research Part E: Logistics and Transportation Review* 45.1 (2009), pp. 96–106. ISSN: 1366-5545. DOI: https://doi.org/10.1016/j.tre.2008.08.003. URL: https://www.sciencedirect.com/science/article/pii/S1366554508001142.

[21] Fred Glover. “Tabu search - Part I”. In: *INFORMS Journal on Computing* 2 (Jan. 1990), pp. 4–32.

[22] Fred Glover. “Tabu search—part II”. In: *ORSA Journal on Computing* 2 (Feb. 1990), pp. 4–32. DOI: 10.1287/ijoc.2.1.4.

[23] Jano van Hemert and J. la. “Dynamic Routing Problems with Fruitful Regions: Models and Evolutionary Computation”. In: Sept. 2004. ISBN: 978-3-540-23092-2. DOI: 10.1007/978-3-540-30217-9_70.

[24] Soumia Ichoua, Michel Gendreau, and Jean-Yves Potvin. “Diversion Issues in Real-Time Vehicle Dispatching”. In: *Transportation Science* 34.4 (2000), pp. 426–438. DOI: 10.1287/trsc.34.4.426.12325. URL: https://doi.org/10.1287/trsc.34.4.426.12325.

[25] Soumia Ichoua, Michel Gendreau, and Jean-Yves Potvin. “Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching”. In: *Transportation Science* 40.2 (2006), pp. 211–225. DOI: 10.1287/trsc.1050.0114. URL: https://pubsonline.informs.org/doi/abs/10.1287/trsc.1050.0114.

[26] Petr Kalina and Jiří Vokřínek. “Algorithm for Vehicle Routing Problem with Time Windows Based on Agent Negotiation”. In: *Proceedings of the 7th Workshop on Agents In Traffic and Transportation, AAMAS* 8291 (Dec. 2013). DOI: 10.1007/978-3-642-44927-7_11.

[27] D Keyes. “Walmart to offer same-day delivery in New York City”. In: *Business Insider* (Oct. 2017).

[28] Maximilian Kronmueller, Andres Fielbaum, and Javier Alonso-Mora. “On-Demand Grocery Delivery From Multiple Local Stores With Autonomous Robots”. In: *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. 2021, pp. 29–37. DOI: 10.1109/MRS50823.2021.9620599.

[29] Gilbert Laporte. “What You Should Know About the Vehicle Routing Problem”. In: *Naval Research Logistics (NRL)* 54 (Dec. 2007), pp. 811–819. DOI: 10.1002/nav.20261.

[30] J Lochem. “Dynamic Vehicle Routing with Stochastic Customers”. In: *TU Delft Repository* (2019).

[31] J. v. Lochem et al. “Anticipatory Vehicle Routing for Same-Day Pick-up and Delivery using Historical Data Clustering”. In: *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. 2020, pp. 1–6. DOI: 10.1109/ITSC45102.2020.9294424.

[32] Jelmer van Lochem. “Improving Dynamic Route Optimisation by making use of Historical Data”. In: *TU Delft Repository* (2020). URL: http://resolver.tudelft.nl/uuid:554bf03a-b57e-4932-962b-defb4af2e50b.

[33] Ning Ma et al. “Improved ant colony optimisation for the dynamic multi-depot vehicle routing problem”. In: *International Journal of Logistics* 16 (Apr. 2013). DOI: 10.1080/13675567.2013.810712.

[34] Jesse Maida. “Online Grocery Delivery Services Market to Grow Over $ 630 Billion During 2020-2024 | 55% Growth to Originate From APAC | Technavio”. In: *Business Insider* (2020).

[35] Stephan Meisel. *Anticipatory optimization for dynamic decision making*. Vol. 51. Springer Science & Business Media, 2011.

[36] Snezana Minic, Ramesh Krishnamurti, and Gilbert Laporte. “Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows”. In: *Transportation Research Part B: Methodological* 38 (Sept. 2004), pp. 669–685. DOI: 10.1016/j.trb.2003.09.001.

[37] Snežana Mitrović-Minić and Gilbert Laporte. “Waiting strategies for the dynamic pickup and delivery problem with time windows”. In: *Transportation Research Part B: Methodological* 38.7 (2004), pp. 635–655. ISSN: 0191-2615. DOI: https://doi.org/10.1016/j.trb.2003.09.002. URL: https://www.sciencedirect.com/science/article/pii/S0191261503001073.

[38] Jairo Montoya-Torres et al. “A literature review on the vehicle routing problem with multiple depots”. In: *Computers Industrial Engineering* 79 (Nov. 2014). DOI: 10.1016/j.cie.2014.10.029.

[39] Victor Pillac et al. “A review of dynamic vehicle routing problems”. In: *European Journal of Operational Research* 225.1 (2013), pp. 1–11. ISSN: 0377-2217. DOI: https://doi.org/10.1016/j.ejor.2012.08.015. URL: https://www.sciencedirect.com/science/article/pii/S0377221712006388.

[40] Jean-Yves Potvin, Ying Xu, and Ilham Benyahia. “Vehicle routing and scheduling with dynamic travel times”. In: *Computers & Operations Research* 33.4 (2006), pp. 1129–1137.

[41] Warren Powell. “Approximate Dynamic Programming: Solving the Curses of Dimensionality”. In: (Aug. 2011). DOI: 10.1002/9781118029176.

[42] Harilaos Psaraftis, Min Wen, and Christos Kontovas. “Dynamic Vehicle Routing Problems: Three Decades and Counting”. In: *Networks* 67 (Aug. 2015). DOI: 10.1002/net.21628.

[43] PYMNTS. “Millennials Lead Grocery’s Digital Shift”. In: (2021). URL: https://www.pymnts.com/news/retail/2021/millennials-lead-grocerys-digital-shift/.

[44] Rahul. “The Many Same-Day Delivery Challenges and Their Solutions”. In: *Route4me* (2020).

[45] Ted K Ralphs et al. “On the capacitated vehicle routing problem”. In: *Mathematical programming* 94.2 (2003), pp. 343–359.

[46] Valuates Reports. “Last Mile Delivery Market Size is Projected to Reach USD 66,000 Million by 2026 at CAGR 8.9%”. In: *PR Newswire* (Dec. 2020).

[47] GlobalTranz Resources. “Last Mile Distribution: The Global State of Last Mile in 2020”. In: *GlobalTranz* (May 2020). URL: https://www.globaltranz.com/last-mile-distribution/.

[48] Damián Reyes et al. “The Meal Delivery Routing Problem”. In: 2018.

[49] Ulrike Ritzinger, Jakob Puchinger, and Richard Hartl. “A survey on dynamic and stochastic vehicle routing problems”. In: *International Journal of Production Research* 54 (Jan. 2016), pp. 215–231. DOI: 10.1080/00207543.2015.1043403.

[50] Yves Rochat and Éric Taillard. “Taillard, E.D.: Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. Journal of Heuristics 1(1), 147-167”. In: *Journal of Heuristics* 1 (Sept. 1995), pp. 147–167. DOI: 10.1007/BF02430370.

[51] Tomasz Rokicki. “E-COMMERCE MARKET IN EUROPE IN B2C; Information Systems in Management, Vol. 7, No. 2, pp. 133-140.” In: 7 (June 2018), pp. 133–140. DOI: 10.22630/ISIM.2018.7.2.12.

[52] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis. “Approximate algorithms for the traveling salesperson problem”. In: *15th Annual Symposium on Switching and Automata Theory (swat 1974)*. 1974, pp. 33–42. DOI: 10.1109/SWAT.1974.4.

[53] C. Schneeweiss. *Distributed Decision Making*. Springer Berlin Heidelberg, 2012. ISBN: 9783540247241. URL: https://books.google.co.in/books?id=d0UBCAAAQBAJ.

[54] Steffen Schorpp. “Dynamic Fleet Management for International Truck Transportation”. In: Jan. 2012, pp. 593–598. DOI: 10.1007/978-3-642-29210-1_94.

[55] Marius M. Solomon. “Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints”. In: *Oper. Res.* 35 (1987), pp. 254–265.

[56] M.Grazia Speranza. “Trends in transportation and logistics”. In: *European Journal of Operational Research* 264 (Aug. 2016). DOI: 10.1016/j.ejor.2016.08.032.

[57] Barrett W. Thomas. “Waiting Strategies for Anticipating Service Requests from Known Customer Locations”. In: *Transportation Science* 41.3 (2007), pp. 319–331. DOI: 10.1287/trsc.1060.0183. eprint: https://pubsonline.informs.org/doi/pdf/10.1287/trsc.1060.0183. URL: https://pubsonline.informs.org/doi/abs/10.1287/trsc.1060.0183.

[58] Paolo Toth and Daniele Vigo. *The Vehicle Routing Problem*. Ed. by Paolo Toth and Daniele Vigo. Society for Industrial and Applied Mathematics, 2002. DOI: 10.1137/1.9780898718515. eprint: https://epubs.siam.org/doi/pdf/10.1137/1.9780898718515. URL: https://epubs.siam.org/doi/abs/10.1137/1.9780898718515.

[59] Marlin Ulmer. “Approximate Dynamic Programming for Dynamic Vehicle Routing”. In: *Operations Research/ Computer Science Interfaces Series* 61 (Jan. 2017). DOI: 10.1007/978-3-319-55511-9.

[60] Marlin Ulmer, Jan Brinkmann, and Dirk Mattfeld. “Anticipatory Planning for Courier, Express and Parcel Services”. In: Jan. 2015, pp. 313–324. DOI: 10.1007/978-3-319-13177-1_25.

[61] Marlin Ulmer, Dirk Mattfeld, and Felix Köster. “Budgeting Time for Dynamic Vehicle Routing with Stochastic Customer Requests”. In: *Transportation Science* 52 (Mar. 2017). DOI: 10.1287/trsc.2016.0719.

[62] Marlin Ulmer and Barrett Thomas. “Same-Day Delivery with a Heterogeneous Fleet of Drones and Vehicles”. In: *Networks* (Aug. 2018). DOI: 10.1002/net.21855.

[63] Marlin Ulmer, Barrett Thomas, and Dirk Mattfeld. “Preemptive Depot Returns for Dynamic Same-Day Delivery”. In: *EURO Journal on Transportation and Logistics* 8 (Apr. 2018). DOI: 10.1007/s13676-018-0124-0.

[64] Marlin W. Ulmer et al. “On modeling stochastic dynamic vehicle routing problems”. In: *EURO Journal on Transportation and Logistics* 9.2 (2020), p. 100008. ISSN: 2192-4376. DOI: https://doi.org/10.1016/j.ejtl.2020.100008. URL: https://www.sciencedirect.com/science/article/pii/S219243762030008X.

[65] Rinde RS Van Lon et al. “Measures of dynamism and urgency in logistics”. In: *European Journal of Operational Research* 253.3 (2016), pp. 614–624.

[66] Stacy A. Voccia, Ann Melissa Campbell, and Barrett W. Thomas. “The Same-Day Delivery Problem for Online Purchases”. In: *Transportation Science* 53.1 (2019), pp. 167–184. DOI: 10.1287/trsc.2016.0732. eprint: https://doi.org/10.1287/trsc.2016.0732. URL: https://doi.org/10.1287/trsc.2016.0732.

[67] Alex Wallar et al. “Vehicle Rebalancing for Mobility-on-Demand Systems with Ride-Sharing”. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2018, pp. 4539–4546. DOI: 10.1109/IROS.2018.8593743.

[68] Haitao Xu, Pan Pu, and Feng Duan. “A Hybrid Ant Colony Optimization for Dynamic Multidepot Vehicle Routing Problem”. In: *Discrete Dynamics in Nature and Society* 2018 (Sept. 2018), pp. 1–10. DOI: 10.1155/2018/3624728.

[69] Zizhen Zhang et al. “Manpower allocation and vehicle routing problem in non-emergency ambulance transfer service”. In: *Transportation research part E: logistics and transportation review* 106 (2017), pp. 45–59.