A Fair Pick-up and Delivery Problem with Time Window (PDPTW) Model for Meal Delivery in Davao City

Amantiad, Blaisdel Kate
Flores, Symon John
Gamao, Jobie May

ATENEO DE DAVAO UNIVERSITY
SCHOOL OF ARTS AND SCIENCES
DAVAO CITY

October 2022

Abstract

Meal delivery operations have grown in popularity as on-demand delivery has gained more momentum over the past few years, and meal delivery operations' stakeholders, especially riders, have faced fairness concerns regarding the food delivery assignment system. The authors proposed a fair two-phase solution approach to the Pick-up and Delivery Problem with Time Window (PDPTW) for meal deliveries in the local context of Davao City: (1) order and rider clustering through constrained k-means, (2) a fair PDPTW Model through mathematical programming and optimization in CPLEX. The mathematical model has an objective of cost minimization function and was subjected to fairness constraints from the riders' perspective: (1) prioritizing riders with longer idle time and (2) maximum ride distance. The model returned the fairest, most optimal, shortest, and least-cost routes and order assignments to riders. Results show that the proposed fair PDPTW model was faster and more scalable than the base PDPTW model from the literature. Also, the study demonstrated how order and rider bundling pre-processing steps through constrained k-means clustering made the whole system more efficient.

Table of Contents

Acknowledgements

Words cannot express the proponents' deepest gratitude to everyone who has made the completion of this significant academic milestone possible.

We are forever profoundly indebted to Ms. Grace S. Tacadao, MScIT, our thesis mentor and adviser, who generously lent us her most precious time, guidance, knowledge, and expertise. This endeavor would not have been possible without your constant guidance in steering us toward the right direction to take for this study. We are eternally grateful for always being there every step of the way, for the weekly meetings, and for all your helpful suggestions despite your busy schedule. We could not have imagined having a better adviser and mentor for our undergraduate thesis study.

We are also grateful to Fr. Ramon Prudencio S. Toledo, SJ, our thesis coordinator, who has always been regularly hands-on in ensuring we accomplish and deliver the thesis at the right time. We are thankful for your valuable insights and advice on further improving our paper and presentation.

We would also like to express our gratitude to Mr. Adrian D. Ablazo, Mr. Antonio G. Bulao II, and Mr. Patrick Angelo P. Paasa for their valuable insights and comments on improving our study during the thesis defense.

Additionally, thanks go to the Department of Science and Technology - Science Education Institute (DOST-SEI), which supported and helped fund this study.

We would also be remiss in not mentioning our families, especially our parents and siblings. Their belief, understanding, and financial and moral support have kept our spirits, will, and motivation high during this process.

Finally, we would like to mention our Almighty Father, who provided us with our needs, strength, and determination to overcome the difficulties we have encountered along our thesis journey.

We are eternally appreciative of all the time, support, and efforts we received from everyone, especially those mentioned above, who took part in this significant endeavor of ours'.

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background of The Study

In the rapidly evolving on-demand delivery industry, where customers place orders at a volatile and unpredictable rate, meal delivery is a significant player. Since COVID-19 came into the picture, food delivery operations have grown in popularity as on-demand delivery has gained more momentum over the past few years. It has been one of the services people all over the globe have relied on since the pandemic, making it one of the essentials, nowadays. In the Philippines, this service ranges from takeaway fast food or restaurant orders to delivering groceries to the doorsteps to store in the pantry. More and more people opt for this convenient meal delivery service since it helps them be safe from the risks of the COVID-19 virus outside and aids new sets of entrepreneurs with their latest food enterprise endeavors. With this, the meal delivery industry in the country has expanded, totaling to an amount of $1.2 billion in terms of the gross merchandise value (GMV) by the final part of 2020, with a 183% increase in the gross meal delivery sector across Southeast Asia (SEA) in the same year [28]. Emerging providers invest heavily in deploying food delivery networks that guarantee businesses and consumers a dependable, quick, and cost-effective delivery method as they strive to capitalize and take advantage of this market opportunity. The long-term viability of these integrated meal delivery networks that emphasize speed and reliability is dependent on turning efficiency potential into actual revenues. Thus, it is necessary to develop appropriate optimization models to address exponentially growing pick-up and delivery situations in near-real-time and recommend high-quality judgments to keep costs down and meet remarkably increased service requirements.

Moreover, according to a research report by the Business Research Company in terms on the worldwide online meal delivery industry, the international online meal delivery industry is predicted to reach $192.16 billion in 2025 at a compound annual growth rate (CAGR) of 11% [32]. The increase is primarily because of businesses commencing operational processes and acclimating to the post-pandemic life while trying to recover from the COVID-19 strike, that had earlier resulted in constrictive preventative guidelines such as the shutdown of business operations, social distancing, and working remotely, all of which created operational challenges [32]. And in the local context, some delivery services noted an upsurge of about 50 percent in their bookings while the City of Davao was under community quarantine due to the coronavirus disease, just in 2020 alone [18]. A manager from Lobby Cafe in Davao city, Yvonne Alverez, remarked that as the number of third-party food delivery services grows, the public may expect increasingly convenient meal deliveries [35]. These figures show the market potential for food delivery and thus, researching on how to improve meal delivery processes can be of great contribution to the industry.

A number of works [38] [45] have already addressed the problem of restaurant meal delivery to customers, while not to the same extent as more traditional routing issues. Such area of research captures many facets of the restaurant delivery operations. Generally, four stakeholders are involved: the food delivery riders, the restaurant that prepares the food, the clients who order the meals, and the organization that coordinates the other stakeholders. For instance, customers expect their meal orders to arrive on schedule and in good condition, and delivery riders want to earn profits, hence they can be selective about whose orders they accept, which can add to the risk of the operation. Also, restaurants want a rider assigned to their area to ensure that drivers deliver a prepared meal as soon as possible because they receive a steady stream of requests. And, the organization that aggregates these requests must manage all operational aspects without interruption and make choices in a fast-paced environment. All of these parties have distinct objectives in mind and act accordingly, posing a myriad of challenges.

The successful operation and deployment of meal delivery networks are complex due to the scale of these systems and the urgency of arriving orders [47]. Meal delivery truly represents the pinnacle of difficulty in last-mile distribution processes. Typically, an order is anticipated to be transported within 60 minutes (or less if applicable) and minutes of the order being fully prepared, limiting the opportunity for consolidation and increasing the need for more riders working at the same time and completing more optimized paths. Moreover, delivery services should be capable of adapting to substantial and frequently unexpected changes in the demand, both in terms of geographic distribution and in terms of time. For this research, the proponents have formulated a two-phase solution algorithm which mainly includes a Pick-up and Delivery Problem with Time Windows (PDPTW) Model that incorporates various constraints to account for various conditions and demand by meal delivery services' stakeholders. For the purposes of this discussion, the PDPTW is a complex vehicle routing (pick-up and delivery) concern that involves a wide range of restrictions and variables that interact with one another.

In this study, the emphasis is on creating solid foundations for designing an optimization tool that can scale up to the task. The proponents adopted a deterministic framework of PDPTW and utilized an enhanced PDPTW model to find optimal solutions. The following are the most significant contributions made by this paper: (**1**) a definition of the localized version of the Pick-up and Delivery Problem with Time Windows (PDPTW) in Davao City; (**2**) the incorporation of the fairness aspect to the localized PDPTW model. All of these contributions address the problem of comprehending the PDPTW's complexity of the algorithm and providing a platform for testing input data to better grasp the variability in meal delivery operations, among others.

1.2 Objectives of the Study

This study has a general objective of formulating a fair PDPTW solution approach to the meal delivery context in Davao City. The specific objectives that this research aims to accomplish are:

- To investigate the localized version of the PDPTW in Davao City
- To identify the fairness notion that best fits the localized version of the PDPTW
- To formulate a model for the localized version of the PDPTW while considering the fairness aspect
- To optimize the fair PDPTW model
- To run experimentation testing PDPTW input data from the local context

1.3 Significance of the Study

Aside from the additional contribution to the literature PDPTW solution approaches with integer programming, coming up with an optimized fair PDPTW model for meal delivery services Davao City contribute to the enhanced experience of the different stakeholders in meal delivery routing processes in the local context. The fairness notions discussed in Chapter 2 align with promoting the common good for the majority, which is one of Ateneo's mission and goals as a Jesuit University, because it helps in building a more equitable pick-up and delivery algorithm especially for food delivery riders. Also, route optimization in itself, which is part of solving the PDPTW, brings about several contributions already. Among these are:

- Reduced carbon footprint (Laudato Si')
- Helps attain sustainable growth for food delivery businesses
- Delivery productivity and efficiency
- The minimum variation in transit time and rider load
- Reduced penalties for low quality service
- Maximized collected profit/score
- Reduced operating/fixed costs

1.4 Research Questions

This study aims to investigate and solve the localized version of the Pick-up and Delivery Problem with Time Window (PDPTW) in Davao City meal delivery, while incorporating the fairness aspect. It generally aims to answer the main research question: "What are the adjustments that have to be made on the base model for the default solver or to the heuristic approach to determine a fair PDPTW model for Davao City?" Specifically, the study aims to answer the following research sub-questions:

1. What is the local version of the PDPTW and its specific settings in Davao City?
2. What changes should be applied to the PDPTW model to adapt it to the local version?
3. What are the fairness notions in the literature that are applicable to PDPTW in the local setting?
4. How do we evaluate the fair model?

1.5 Scope and Limitations

The scope of the research was only limited to solving the Pick-up and Delivery Problem with Time Window in Davao City's context of meal delivery. The data collection was conducted through an in-depth interview to randomly selected local food delivery riders, courier companies (through the riders), and customers within the city who represented the population. Delivery riders who do not currently reside in Davao City were not included in the scope of this study. Routes and restaurants were also limited within Davao City. This study did not cover problems and situations which do not fall under the deterministic framework. In a deterministic problem setting, all problem information is assumed to be known with certainty at the time of making a decision. This means that, for example, the exact time it takes to travel between two points is known [1]. This study does not tackle stochastic problem setting where information may be unknown or uncertain at the time of solving the problem. Dynamic instances wherein orders come at any point in time are not part of the scope. Also, actual real road networks are not part of the considerations for this study.

Chapter 2

Review of Related Literature

2.1 Routing Problems

In traditional routing problems, Vehicle Routing Problem (VRP) is a known area of research in terms of transportation and shipment loading [40]. The goal is to build an optimized route for every driver to attend to all requests while minimizing costs. The definition of how a path is constructed and minimized costs are linked to the variant considered, more specifically, to its restrictions. There are many variants of the VRP that target and consider various constraints. As per Van Breedam [46], the VRP, which was described by Dantzig and Ramser as the Truck Dispatching Problem, and later by Laporte and Osman and Fisher, is an essential extension to the Traveling Salesman Problem (TSP) that was first modeled on computers by Dantzig and Ramser, Clarke and Wright, and Lin. TSP entails the following tasks: determining the shortest path a salesperson can consider taking through many city areas, beginning and finishing at the same location, and visiting cities just once. To put it another way, the TSP can be implemented to just about every situation in which the objective is to look for the Hamiltonian cycle with the minimum cost of a weighted graph. In VRP, vehicles begin and conclude their travels at a depot and have a set maximum route length. As time passed, more variants of VRP have been developed to cater to the more modern practical routing problems while considering more constraints and restrictions such as pick-up and delivery problems.

In modern routing problems, the different forms of VRPs (i.e., PDPTW, MDRP, RMDP) occur in food deliveries and product distribution. VRP's general aim was to reduce route costs. It is a combinatorial integer optimization problem [5]. Now it has been a crucial issue to solve these problems with optimum costs within a reasonable time. Specific versions and applications may have different objective purposes. The following are the most prevalent VRP objectives according to Sitek et al. [41]:

- to reduce the number vehicles required to cater all delivery sites;
- to reduce all transportation expenses and distance traveled;

The VRPs look at a variety of practical, real-world applications, generally mostly focus on delivery services. The following is a comprehensive description of VRP:

- The items or commodities are delivered from one or more depot;
- A set of riders/couriers operates a group of vehicles.
- The network of roads and trains is provided.
- A list of delivery nodes/points where the products are delivered is provided.

Another variant of VRP is the dynamic pick-up and delivery problem (DPDP), and one VRP variant comparable to this is the meal delivery routing problem (MDRP) which occurs when commodities must be carried from different pick-up points to different drop-off points [45]. MDRP is a dynamic problem with customer deadlines, a goal function centered on customer service, and a large number of vehicles fulfilling a high number of orders. Ulmer et al. [45] investigated a stochastic dynamic pick-up and delivery problem to dynamically control a fleet of drivers to avoid customer deadline delays. To account for the problem's dynamism and uncertainty, an anticipatory customer assignment (ACA) policy is developed, which relies on a time buffer and delay. Yildiz and Savelsbergh [55] also proposed a meal delivery routing problem model that assumes perfect knowledge of order arrivals. For its solution, they created a simultaneous column and row-generation method. Moreover, Rey et al. [37] developed a method for recovering perishable food delivery problems by identifying the exact solution with the lowest travel costs. To efficiently provide envy-free and cost-effective solutions, a heuristic approach that combines greedy and local search was presented. Furthermore, Wang and Jiang [49] explored a meal delivery problem in which vehicles must make multiple travels to pick up meals from different suppliers and deliver them to customers. For multi-trip routing with soft time windows and numerous refill locations, two standard heuristic methods, iterated local search (LS) and adaptive large neighborhood search (ALNS), were devised. Table 1 presents the comparison between existing VRP literature and this study.

2.2 Pick-up and Delivery Problem (PDP)

As society modernized and last-mile logistics and transportation became more complex, researchers have devised the Pick-up Delivery Problem (PDP) from VRP to address numerous challenging features in many real-life pick-up and delivery problem instances, like start times and time deadlines (time windows). The PDP is a variant of VRP that was introduced by Savelsbergh [40]. One variant of PDP is PDPTW, PDP with Time Windows (PDPTW), which models the circumstances in which a fleet of vehicles must service a collection of transportation requests with a time window and more constraints [27, 13, 40]. Each request has a pick-up and delivery location. Vehicles must be assigned routes to serve all requests, fulfilling time windows and vehicle capacity constraints while optimizing a particular objective function such as total distance traveled. PDPTW can model many core problems emerging in logistics and public transit. Finding good solutions to these problems is essential because it helps planners to utilize the existing fleet cost-effectively to meet customer demands. PDPTW comprises orders indicating a collection of origins attributed with one destination; or one source associated with several ending areas, vehicles with varied starting and ending locations, and real-time transportation requests. This reflects modern society as many practical pick-ups and delivery situations

Table 1: Comparison between the existing VRP literature and this study.

Research	Two-Stage Model	Clustering	Heuristic	Operational Environment
Wang and Jiang (2022)	✓	HAC	GA	Python 3.7
Ulmer et al. (2021)			ACA Policy	Java
Xue et al. (2021)	✓		LNS	C#
Liao et al. (2021)	✓	K-means	GA	
Wang (2018)		Spatial-Temporal Clustering	ILS, ALNS	CPLEX 12.7
Rey et al. (2018)	✓	Multi-objective Pick-up Node Clustering Procedure	Greedy, LS	CPLEX, AMPL, Java
Steever et al. (2018)			Auction-Based	Python
This study	✓	Constrained K-means		CPLEX

are demand-responsive; for instance, new transit sets of pick-up and delivery destinations become readily accessible in real-time and are eligible as soon as they are considered. And thus, once the system deploys the vehicle, the collection of routes and destinations must be re-optimized to accommodate new transportation demands.

2.3 Fairness Notions

This study aims to incorporate the fairness aspect in solving the Pick-up and Delivery Problem with Time Windows (PDPTW) in the local meal delivery context of Davao City. This section tackles the different fairness notions in vehicle routing problems from the perspective of the various stakeholders in delivery services (i.e., company, driver, and customers).

One fairness notion refers to "fairness" from the riders' perspective. A work by Chen et al.[11] on Restaurant Meal Delivery Problem brought up such a fairness aspect. Their Anticipatory Customer Assignment (ACA) policy maintained fairness among riders in that, over time, all riders delivered almost the same number of orders. They considered the fairness of drivers and used the average minimum and maximum numbers of services to evaluate the performance of policies. Also, workload balancing among drivers was another fairness aspect explored by Bowerman et al. [7], Li and Fu [30], as well as in those from the literature on multi-objective VRPs [5, 22, 26, 52]. This relates to Lesmana et al.'s [29] study on fairness in terms of driver profit, wherein they measured fairness as the minimum profit (utility) among all drivers. They only improved the solution from an existing assignment until it satisfied a predefined fairness threshold [29]. Also, Kritikos and Ioannou [24] investigated a VRPTW variation that accounts for fairness by balancing the weights carried by different active vehicle fleet members. The problem's objective function is the weighted total of route expenses, vehicle costs, and vehicle loading imbalance. They suggested a novel way to produce a solution based on the free disposal hull data envelopment analysis method. Implementing the proposed methodology to benchmark problems revealed outstanding results that steadily improve vehicle loading compared to existing methods that deliver the best answers in terms of vehicle count and overall distance traveled. Moreover, Ghiani et al. [15] pointed out how large waiting times would seem a waste of time to both riders and fleet supervisors and would be judged as unacceptable, which somehow tackled fairness. And so, Ghiani et al. [15] imposed that the total waiting time along a route does not exceed a maximum amount to make the solutions feasible in the real world. Similarly, another research about the formulation of a linear programming model for the vehicle routing problem by Cam [9] aimed to minimize idle time. There are also other VRP studies with implementations that implicitly covered notions related to fairness. For example, Xue et al. [54] developed a two-stage approach using mixed-integer programming (MIP) and large neighborhood search (LNS) algorithms to improve the scheduling of riders for food delivery services in the O2O sector. They considered various sensitivity analyses, such as the tightness of the order time windows associated with the orders and riders' familiarity with delivery regions. They discovered that the smaller the familiarity, the larger the actual travel distance is. Thus, a "fairness notion" is that riders should be assigned to a specific zone to reduce the distance traveled and the total delivery cost.

Moreover, in terms of fairness notions on the customer's side, Chen et al. [11] explored the problem of offering a fair same-day-delivery (SDD) service to customers. Their paper considered the second objective of fairness in customer service, which focused on geographical fairness in SDD service availability. This work partitioned the service area and measured fairness as the minimum service rate across the resulting regions. They studied an SDD problem where customers from different designated parts make delivery requests throughout the day. A central decision-maker chooses which customers to serve and what vehicle should serve the customer's request. With fairness in mind, they modeled the problem as a multi-objective Markov decision process of maximizing a weighted combination of the expected overall service rate (utility) and minimal regional service rate (fairness). Their consideration of geographical fairness in services directly reflects the societal problems that neighborhoods that are financially advantageous or geographically closer to facilities receive more public or private resources, while the vulnerable with a disadvantage of location have a far lower chance of receiving the service. And with this study by Chen et al. [11], in studying dynamic delivery problems, Soeffker et al. [42], and Ulmer et al.[45] considered fairness as an evaluation metric. They highlighted that the methods select lucrative customers in central areas and discriminate against customers in more rural areas, especially in "high-quality" policies.

Furthermore, there is a fairness notion from the courier or rider companies' point of view. An actual case study on order allocation across Hong Kong by Zhang et al. [56] illustrated the characterizing features of VRP with outsourcing and fairness. In their research, by “fairness,” they were about profit balancing. The researchers presented two multi-objective local searches (MOLS) algorithms as promising solutions. They conducted extensive experiments on an import company in Hong Kong that uses a decentralized delivery strategy, meaning they laid their eggs in different baskets to hire outsourced transport shippers. A transport shipper’s profit is the difference between the overall transportation cost and the rewards gained by executing the assigned orders. In practice, profitability should be comparable between two businesses with identical circumstances. Otherwise, the less profitable hired shipper may protest and raise the price in the future, putting the company’s long-term gain at risk. The fairness objective here requires balancing the profits among outsourced transport companies as closely as possible, as such fair division of load of work should be considered. The findings demonstrated that the computational results of the improved MOLS algorithm could achieve a suitable solution in optimizing vehicle routing from outsourced employees while accounting for profit-balancing.

2.4 NP-hardness of VRP

VRP and its variants are combinatorial optimization problems that are Non-Polynomial-hard (NP-hard) [3, 25, 6]. As an NP-hard problem, obtaining a perfectly optimized solution to VRP [23] is tedious. Because of its NP-hardness, a universal challenge in solving a variety of VRPs is the exponential increase of computation complexity when the number of entities such as nodes, roads, and vehicles increases [10]. For instance, in Anh et al.’s study about VRPTW in the context of the fresh food distribution center, their system found an optimal solution in 2 minutes for eight nodes, their smallest instance; however, it already took 5 hours to solve 28 nodes, their largest instance [2]. This shows that MIP can optimally solve specific small-scale problem sizes using mathematical or combinatorial optimization tasks [23]. Aside from these methods, researchers use several more approaches to VRPs. This section touched upon different applications of clustering in various industries and discussed heuristics and a combination of clustering and heuristics as solutions to VRPs.

2.4.1 Clustering in VRP

Researchers have already used clustering methods to handle, classify, and consolidate large datasets. Hsieh and Huang [20] used the K-means algorithm to merge online shopping orders and the self-organized map batching heuristics to improve the performance of order-picking systems. Aside from the previously mentioned industry application, Ishizaka et al. [21] also developed a multi-criterion divided hierarchical clustering algorithm that overcomes uncertainty and imprecision issues. It was utilized to cluster financial institutions for US banks with enhanced performance. Also, Tumpa et al. [44] employed the hierarchical clustering approach in assessing the frequent hurdles in the textile industry’s green supply chain adoption. Additionally, Xu et al. [53] established a suitable validity measure for agglomerative hierarchical clustering, demonstrating the effectiveness of determining the ideal number of clusters.

Several works of literature are available on heuristic methods for optimization problems. For instance, Homsi et al. [19] suggested an exact branch-and-price algorithm with a hybrid genetic search to address industrial and tramp ship routing and scheduling problems. Vidal et al. [48] presented an efficient hybrid evolutionary algorithm with sophisticated diversity control for a wide range of time-windowed vehicle routing challenges. Zhou et al. [57] created a hybrid multi-population genetic algorithm to solve a real-world territory design problem for a big dairy corporation to minimize total operational costs. Various heuristic algorithms have been applied in meal delivery routing optimization, including the genetic algorithm and the greedy adaptive neighborhood heuristic [31, 37, 50, 54, 43].

Researchers frequently combine clustering with other heuristics, meta-heuristics, and approximate or exact solution approaches in dealing with optimization problems [58, 34]. For example, Dondo and Cerda [12] proposed a three-phase cluster-based optimization approach based on the cluster first-route second philosophy for the heterogeneous fleet vehicle routing problem. After combining nodes into a few clusters in phase I, such groups are assigned to vehicles and sequenced on the relevant trips in phase II. The detailed routing and scheduling for each tour discovered in phase II were calculated in phase III by solving a small cluster-based mixed-integer linear programming (MILP) model. Moreover, Wang and Vidal [17] introduced a hybrid genetic algorithm that used K-means as a local search together with problem-tailored genetic operators. In addition, Wang and Lin [51] devised a scenario-based heuristic algorithm using an "intimacy degree" index to categorize customers into different clusters. Also, Mendes et al. [33] used a Pearson's and τ-Kendall hierarchical cluster approach to reduce the dimensionality of the multi-objective vehicle routing problem solved by the MOEA/D evolutionary approach. Likewise, Ozdamar and Demir [36] devised a heuristic approach with hierarchical clustering to coordinate vehicle routing in large-scale disaster distribution planning.

2.5 The Base Model

The proponents considered many VRP studies as the point of reference for this research. This section discusses all related literature that the researchers considered for this study. This part discusses the primary VRP studies that the researchers mainly used as the basis for this research after many deliberations and evaluations of said literature.

Initially, the researchers built this study upon Reyes et al.'s [38] "The Meal Delivery Routing Problem." Reyes et al. introduced a dynamic deterministic model of the structure and functioning of meal delivery systems. They presented a linear assignment model, a parallel-insertion bundling approach, and a rolling-horizon repeated-matching algorithm to solve the problem in near real-time. However, their study focused on solving the problem through their heuristics rather than the mathematical model. When the researchers of this study, the researchers of the Fair PDPTW model for Davao City, started working on Reyes et al.'s linear assignment model before doing some model modifications, they had difficulty replicating their solution approach. This struggle was because of the incomplete documentation in Reyes et al.'s paper for their process in order bundle generation, particularly the service delay part, which was a very crucial section they did not wholly detail. This has always been the major issue with most VRP studies. Most of the authors in this area of research utilize algorithms and heuristics as an approach to VRPs, so they are usually loose in presenting the model since they often do not translate it into a Linear Programming (LP) solver or use Mathematical Integer Programming (MIP). Thus, most models of VRP and its variants usually cannot be readily written into a decision optimization system.

From there, the authors considered another paper which was Wang & Jiang's "Two-Stage Solution for Meal Delivery Routing Optimization on Time-Sensitive Customer Satisfaction" [49]. They established a two-stage meal delivery routing problem (MDRP) optimization solution to maximize time-sensitive customer satisfaction using the clustering method and heuristic algorithm. In the first stage, meal orders were hierarchically classified and merged into delivery bundles based on the nearest pick-up point rule. The ward hierarchical agglomerative clustering (HAC) algorithm was applied to improve fast meal delivery efficiency. In the second stage, the proponents established a cluster-based MDRP optimization model for fast delivery services of meal orders in each delivery bundle with the objective of maximizing time-sensitive customer satisfaction. Their numerical simulation results verified that the two-stage routing optimization solution effectively schedules timely meal delivery and improves customer satisfaction. The comparison of the results indicated the superiority of the proposed two-stage solution with HAC and genetic algorithm (GA) on customer satisfaction while ensuring the delivery of all orders within sixty (60) minutes. However, while Wang and Jiang's [49] study got good results, the researchers have found some concerning aspects of the study. First, their dataset did not include some rider details, such as location and maximum rider capacity. Second, they only tested their solution on one instance. Lastly, Wang and Jiang [49] only considered the pick-up points in the bundling of orders which can result in clustering orders with distant delivery points, which may not be optimal.

Researchers of this study then took into account another related literature by Xue et al. entitled "Optimization of Rider Scheduling for a Food Delivery Service in O2O Business"[54]. Similar to Wang & Jiang [49], they had a two-phase approach, but their methods for each phase differed. In their first phase, they formulated a two-stage model to address the problem of configuring the transportation resources of delivery couriers and optimize both quantity and travel routes of the riders for fluctuations of customer demand over time and space in any period in a region. In their second phase, they designed an LNS approach to produce the delivery rider's route provided the defined minimum number of delivery riders in the first stage of their model. Their results revealed that their proposed model and algorithm could optimize the number of delivery riders and improve delivery rider scheduling. However, upon evaluating their paper and testing the model, there were many typographical errors, which may be most likely because the paper was translated to English from a Chinese language. The formulation of their model was also questionable.

The proponents then considered another study, Ahn et al.'s [2] research called: "The vehicle routing problem with time windows: A case study of a fresh food distribution center." Said research was the primary basis of this study on building a fair PDPTW model for Davao City. In Ahn et al.'s study, they also implemented a two-stage process to optimize the scheduling of riders for fresh food distribution: (1) a heuristic-based clustering algorithm that was applied to group locations into a certain number of clusters,

(2) a VRPTW MIP model with an objective of minimizing total delivery service cost that authors implemented on CPLEX to solve the optimal routes for each cluster. Their results and conclusion detailed that not only the proposed routing system offers the minimum traveling distance with fewer trips covered per night, but it also yielded a better-utilized average truck-loading ratio. The authors minimized about 40 percent of the total cost regarding such an outcome. This paper was good enough; however, the proponents did not settle right away with their MIP model since it looked like it could have been more inclusive of the different aspects of VRPs.

Finally, the authors of this study decided to adapt Grandinetti et al.'s [16] mathematical model from their research entitled: "The multi-objective multi-vehicle pick-up and delivery problem with time windows." In their study, they first presented the Single Objective Multi-Vehicle Pick-up and Delivery Problem with Time Windows (SOMV-PDPTW) Model, which was then adapted and modified by the proponents of this study. The proponents added new constraints and considerations. In summary, the main contributions of this study on building a fair PDPTW model for Davao City in the VRP literature include: (1) the introduction of constrained k-means clustering algorithm as a pre-processing step for the bundling of orders, (2) the incorporation of the fairness aspect (in riders' perspective) in the model by considering idle times as a determinant for priority scheduling, (3) the limiting of riders' idle time for fair rider dispatching, (4) the inclusion of the maximum ride distance constraint, (5) the application of the PDPTW solution model to the local meal delivery service context in Davao City.

Chapter 3

Methodology

3.1 Theoretical Framework

In dealing with the thesis problem on pick-up delivery in the context of meal deliveries, the authors considered Dial-a-Ride Problems and a variant of Vehicle Routing Problems (VRP), the Pick-up Delivery Problem. In the Dial-a-Ride Problems (DARP), clients request transportation from a particular pick-up point or origin at a specific drop-off point or destination [14]. Vehicles complete transportation requests that offer a shared service in the sense that some passengers might be in a vehicle altogether. The objective is to design a collection of low-cost vehicle paths able to meet as many requests as possible while adhering to various constraints such as time windows, maximum travel distance, and etc [1]. Allowing clients to enforce a time limit on their departure and arrival timings is a typical pattern in DARP models; however, this may be overly restrictive for the carrier, especially if these time constraints are tight. DARP models commonly consider maximum ride distance aside from time windows which makes it a suitable candidate theory for this thesis. However, DARP models mainly concern transportation, so it generally represents people instead of goods or food delivery which is the subject of this thesis. Thus, the authors introduced the Pick-up Delivery Problem because the problem is one of transporting items rather than people.

PDPs, Pick-up and Delivery Problems, are a class of VRPs (Vehicle Routing Problem) in which a group of riders deliver items from pick-up points to delivery locations. Each transportation order request is defined by a pick-up point, a delivery area, and a load [14]. In PDP with Time window (PDPTW), each point has a time window to be followed. Rejection of an order demand is not allowed, and the entire request has to be served by a single rider. The proponents introduced PDPTW to address different complex features found in many practical real-life pick-up and delivery instances. This includes order requests detailing a bunch of origins associated with a single destination, or a single origin related to a set of destinations vehicles with different beginning and ending points, with transportation requests evolving in real-time. PDPTWs are well suited in dealing with the sub-problems that occur in demand responsive routing problems, making them suitable for the researchers' study [1].

3.2 Conceptual Framework

This section identifies the sources of the concepts that were used for this research. The main theory adapted for the conceptual framework in this study is the PDPTW, specifically the concept of constraints (i.e., time window, vehicle capacity), while incorporating the fairness notion aspect.

3.2.1 Localizing the Problem

The first step was localizing the problem by obtaining information from food delivery companies. An interview with the food delivery drivers was conducted as part of this process. This is essential because there is a need to understand PDPTW's specific settings in Davao City to provide a more realistic and practical solution to the problem that suits the local context. The researchers were then able to determine what adjustments should be made to the PDPTW model to adapt it to its local version. Results from the interview helped the researchers know what fairness constraints to include and figure out the values to set for the model parameters.

3.2.2 Formulating the Mathematical Problem

The next phase was to develop the mathematical model for the PDPTW's deterministic framework. This is important because constraints such as the time frame, maximum route distance, vehicle capacity, and the fairness concept must be considered. This section addresses the sub-questions on the local version of the PDPTW, its unique circumstances in Davao City, and the fairness principles in the literature that would be relevant. The overall goal was to minimize operating costs while figuring out what fairness notion was the most applicable in the local context and how it could be utilized since the fairness aspect could be another objective function or a constraint. In the end, the researchers incorporated the fairness aspect as an additional constraint.

Time Window

This constraint considers that each pick-up and delivery must occur within its time window. This concept is from the theory of the PDPTW [13].

Maximum Ride Distance

The proponents have incorporated a maximum ride distance in the model. This constraint takes into account the maximum allowable travel distance from the origin to the destination. The distance between the pick-up and delivery node must not exceed this value. In this study, the ride distance is measured between the starting pick-up node and the ending drop-off node. This concept was from the VRP concept applied by Andersson & Tomas [1].

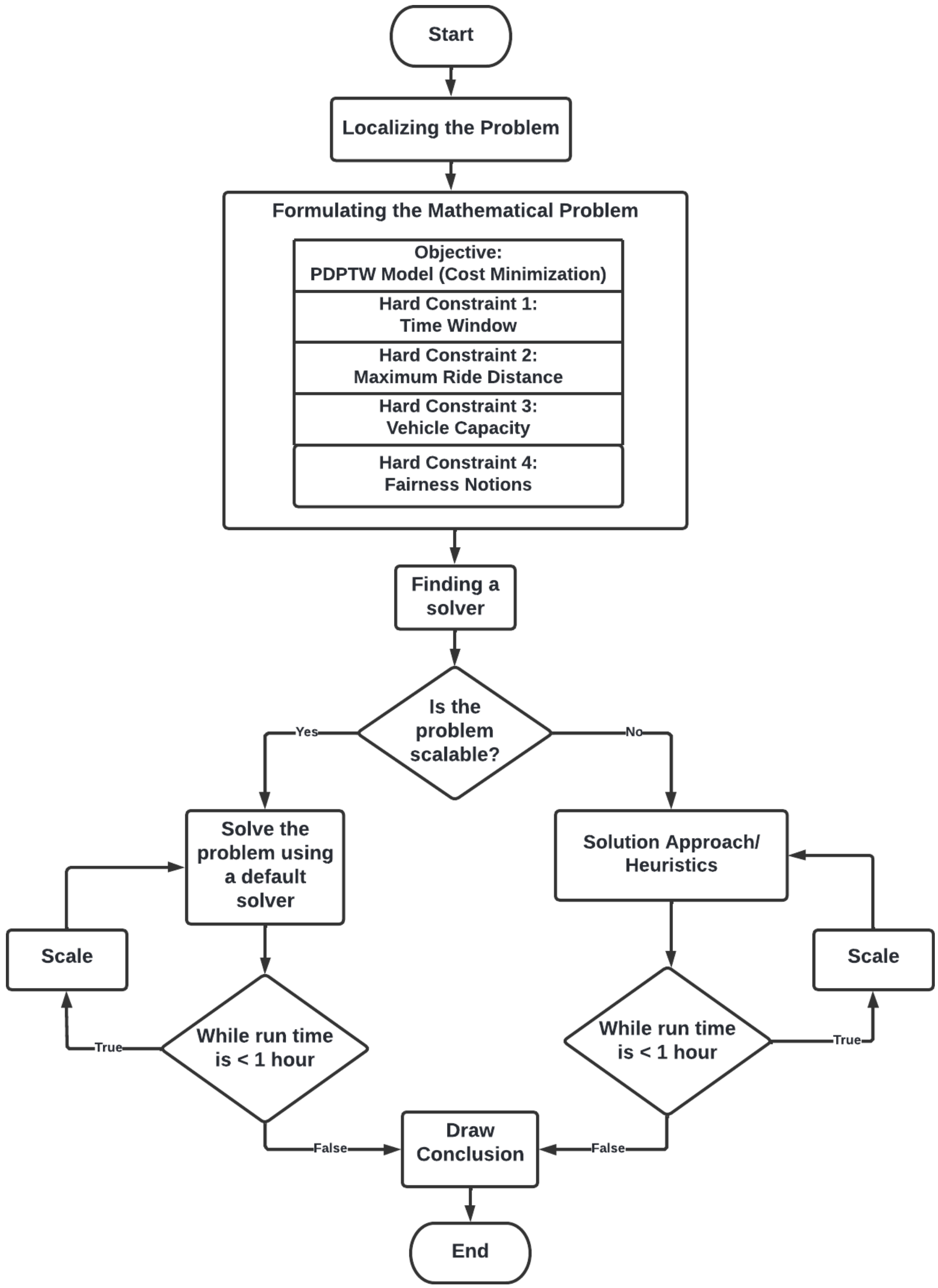

Figure 3.1: Conceptual Framework

Vehicle Capacity

This constraint takes into account the capacity of the rider's vehicle load. This was included in the theory of the PDPTW [13].

Fairness Notions

The fairness notions considered were mainly for the riders. The fairness notions are:

1. Riders with longer idle times should be assigned orders first.
2. A maximum ride distance must be set to ensure that riders would not be placed outside their zones.

These fairness notions were chosen based on present related literature as well as the result of the in-depth interview with the riders which can be found in Appendix C.1.1. The first fairness aspect above was a concern from the riders since according to them, the assignment of orders is random. Thus, the proponents have devised a model that somehow prioritizes riders that are idle for a longer period of time. This fairness constraint accounts for a more reasonable order assignment, as well as the reduction of overall riders' waiting time since this is a waste of time and is deemed unacceptable as per Ghiani et al. in their study: "Anticipatory algorithms for same-day courier dispatching" [15]. For the second fairness notion, the drivers detailed about having a pick-up and delivery zone. From this, the proponents have ensured that the model will ensure that the riders will not be placed outside their assigned area since it would be unfair to the riders if they keep getting assignments to very far pick-up and delivery nodes. This will also allow the rider to return to his preferred area before getting assigned again by the system. This is also relevant to control instances where riders are assigned in unfamiliar locations because it has been discovered in previous studies (i.e., Xue et al. [54]) that riders travel significantly less distance when they are more familiar with the task sub-region.

Using a Default Solver

This step involves running the PDPTW problem in a solver to generate optimal solutions. For this study, the researchers used CPLEX as a solver. The researchers chose said platform as an optimization tool for it is a preferred platform utilized in operations research, most especially on VRP research as seen on Table 1 on Chapter 2. It is also the more flexible optimization tool. The proposed approach was coded in this platform and different instances of data were considered.

3.3 Methodology

The first step was localizing the pick-up and delivery problem by gathering data from food delivery companies. This involved an interview with the food delivery drivers. In the interview with the drivers, the researchers tried to get their food delivery companies' details to know the side of the companies as well. Same goes for the customers. This was essential to cater the inputs of the different stakeholders. This whole process followed the necessary steps, etiquette, and ethical considerations in interviews and data gathering procedures. Through this, the researchers were able to know what changes can be applied to the PDPTW model to adapt to its local version. This was necessary because the proponents needed to know PDPTW's specific settings in Davao City for a more realistic and practical solution to the problem that fits the local setting. This answers the sub question about the local version of the PDPTW and the changes to be applied to the PDPTW model to adapt to said local version. The duration for this step was 4 weeks. Week 1 was allocated for searching of delivery riders that can be interviewed. Week 2-3 was allocated for interviews and research for valuable information through online resources in addition to the data needed from food delivery companies. Week 4 was for the summarizing of data and interpretation.

The following steps took take 3-4 months to complete, working step one after another and back overlapping time frames. The second step of this research was to formulate the mathematical model for the deterministic framework of the PDPTW. This was necessary because the constraints such as time window, maximum distance, vehicle capacity, as well as the fairness notions have to be accounted for. This answers the sub questions on what is the local version of the PDPTW and its specific settings in Davao City, and what fairness notions in the literature will be applicable. The formulation for the mathematical model was worked on for four (4) weeks with overlapping weeks with the next steps. This period also included the preparation of data sets to be used in testing the model. For this study, the researchers utilized Grandinetti et al.'s [16] extended PDPTW mathematical formulation by Ropke and Cordeau [39].

The researchers then modified and made adjustments to the model by considering the fairness notions that have been derived from literature and interviews that are applicable in Davao. The fairness constraints added to the model were: (1) riders' idle times such that riders who have been idle for a long time get orders first, and (2) maximum route distance such that deliveries will not take too long and too far.

After formulating the mathematical model, the next step was utilizing the solver. This took around five (5) weeks. This stage involved deciding on a default solver and translating the mathematical model into the tool. For this study, the researchers have selected to run their model through Python with CPLEX, a known optimization platform. In this step, the mathematical model was translated into a code in CPLEX and was executed to produce optimal solutions. The running of the model in the solver is vital as it is where the assigning of orders to the rider and route generation happens. Also, this determined the best riders assigned on specific orders and helped discover if the system can bundle the order from other orders and if the orders may be postponed.

Lastly, simulation and experimentation were done to test the input data which took about three (3) weeks. Overall, there were two (2) instances used which were all generated based on random actual locations in Davao City. Afterwards, results were presented and conclusions were made with regard to the best-found solution to the PDPTW considering the fairness notions.

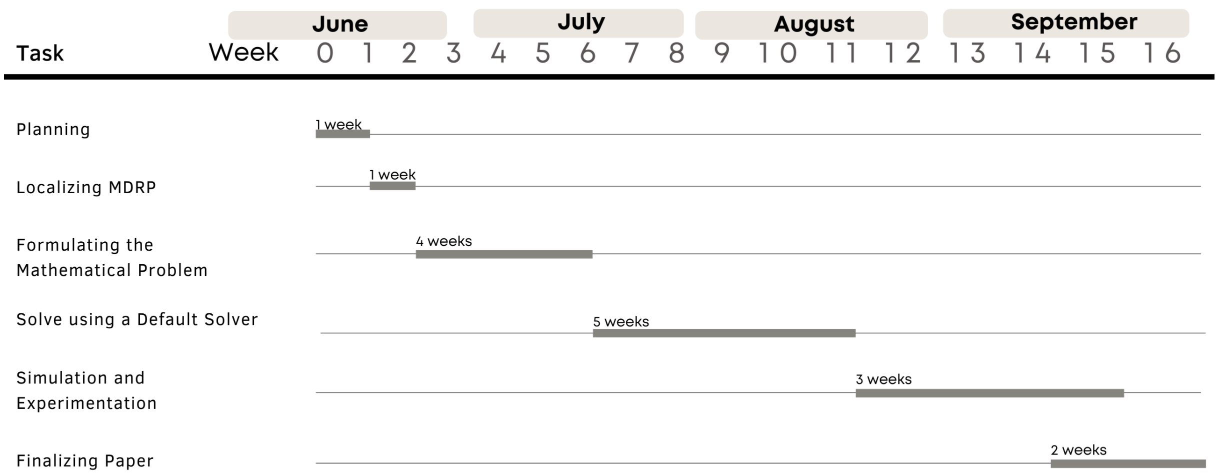

Figure 3.2: GANTT Chart.

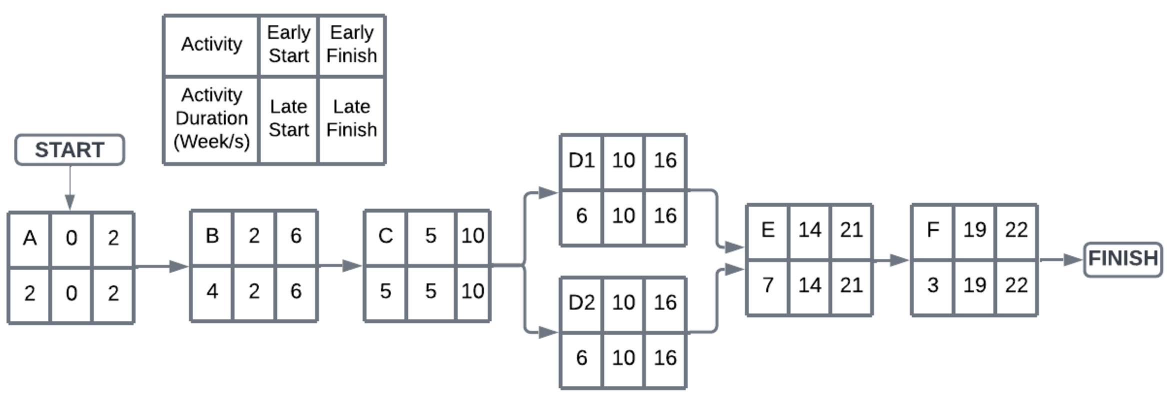

Figure 3.3: PERT-CPM for the Study.

3.3.1 Implementation

Phase I: Order-Rider Clustering and Allocation

Before doing the actual Mathematical Integer Programming on CPLEX, the mathematical model needs inputs. For this study, the inputs needed were the order bundle and rider-to-bundle assignments.

Algorithm 1 Cop-Kmeans Algorithm

1: Let $C_1 ... C_k$ be the initial cluster centers.
2: For each point d_i in D, assign it to the closest cluster C_j such that VIOLATE-CONSTRAINT$(d_i, C_j, Con \neq)$ is false. If no such cluster exists, fail (return {}).
3: For each cluster C_i, update its center by averaging all of the points d_j that have been assigned to it.
4: Iterate between (2) and (3) until convergence.
5: Return $C_1 ... C_K$

As a pre-processing step, the researchers used Bradley et al.'s [8] constrained k-means clustering approach to group the orders into bundles. The proponents utilized Babaki's [4] package for it. Such clustering method was utilized in order to make it possible for pairs of nodes (pick-up and delivery) to be linked with each other for clustering, a feature that basic K-means clustering does not have. Taking into account all nodes associated in an order is a more appropriate approach to clustering orders rather than choosing only one node as basis, either pick-up or delivery node, which is what most studies do, since considering only one node could lead into instances wherein its associated node happens to be so distant.

After coming up with the order bundles, the nearby riders' locations were clustered by defining its centers based on the order bundles centers, to ensure that the riders' location match best with the order bundles. Simple k-means clustering was used here as only one location was being considered unlike the orders which have a pair, the pick-up and delivery nodes.

Phase II: Route Assignment Generation through Mathematical Integer Programming

The mathematical model, which was adapted and modified from the PDPTW mathematical model [16], is as follows:

Sets

P := set of restaurant pick-up nodes

D := set of customer drop-off nodes, such that $P \cap D = \emptyset$

R := set of riders

R_s := set of assigned riders in clustered subregion s

O_s := set of orders in subregion s, denoted by (i, j), where $i \in P$ and $j \in D$

$N = P \cup D \cup R$:= set of all nodes in O_s including the pickup (restaurant) locations, delivery (customer) locations, and riders' starting location 0

Parameters

a := the time the order is ready to be picked up at p

b := the latest arrival time the order is expected to arrive at d

q := the demand at node

Q := the maximum vehicle capacity

st := the average service time set at each node

t^r_{ij} := the travel time from node i to node j

d^r_{ij} := the distance between node i and node j

W := a large non-negative number

γ_1 := fixed cost for each rider

γ_2 := travel cost per meter

δ := the total idle time

β_1 := minimum idle time requirement for order-rider assignment

β_2 := minimum idle time requirement for assignment prioritization

TL := route time limit, the difference between a and b

DL := route distance limit

Decision variables

B^r_i := continuous variable for the arrival time of rider r at node i

Q^r_i := continuous variable for the load quantity of rider r as they arrived at node i

x^r_{ij} := binary variable. 1 if rider r travels from node i to node j, 0 otherwise

y^r := binary variable. 1 if rider r is assign, 0 otherwise

Objective

$$\min \sum_{r\in R_s}\sum_{i\in N}\sum_{j\in N|j\neq 0} \gamma_2 d^r_{ij} x^r_{ij} + \sum_{r\in R_s} \gamma_1 y^r \tag{3.1}$$

Subject to the following constraints:

$$\sum_{r\in R_s}\sum_{j\in N|j\neq i} x^r_{ij} = 1 \quad \forall i \in O_s(i,j) \tag{3.2}$$

$$\sum_{j\in N|j\neq p} x^r_{pj} - \sum_{j\in N|j\neq d} x^r_{dj} = 0 \quad \forall p,d \in O_s(p,d), r \in R_s \tag{3.3}$$

$$\sum_{j\in N|j\neq 0} x^r_{0j} = y^r \quad \forall r \in R_s \tag{3.4}$$

$$\sum_{d\in D_s|d\neq 0} x^r_{d0} = y^r \quad \forall r \in R_s \tag{3.5}$$

$$\sum_{i \in N} \sum_{j \in N | j \neq i} x_{ij}^r \leq W y^r \quad \forall r \in R_s \tag{3.6}$$

$$\sum_{i \in N} \sum_{j \in N | j \neq i} x_{ij}^r \geq y^r \quad \forall r \in R_s \tag{3.7}$$

$$\sum_{j \in N | i \neq j} x_{ji}^r - \sum_{j \in N | i \neq j} x_{ij}^r = 0 \quad \forall r \in R_s, i \in P \cup D \tag{3.8}$$

$$B_j^r - B_i^r \geq (st + t_{ij}^r) x_{ij}^r - W(1 - x_{ij}^r) \quad \forall i, j \in N | i \neq j, r \in R_s \tag{3.9}$$

$$Q_j^r - Q_i^r \geq q_j x_{ij}^r - W(1 - x_{ij}^r) \quad \forall i, j \in N | i \neq j, r \in R_s \tag{3.10}$$

$$B_p^r - B_d^r + t_{pd} \sum_{j \in N} x_{pj}^r \leq 0 \quad \forall p, d \in O_s(p, d), r \in R_s \tag{3.11}$$

$$B_i^r \geq a_i \sum_{j \in N} x_{ij}^r \quad \forall i \in N, r \in R_s \tag{3.12}$$

$$B_i^r \geq b_i \sum_{j \in N} x_{ij}^r \quad \forall i \in N, r \in R_s \tag{3.13}$$

$$\max\{0, q_i\} \sum_{j \in N} x_{ij}^r \leq Q_i^r \leq min\{Q, Q + q_i\} \sum_{j \in N} x_{ij}^r \quad \forall i \in N, r \in R_s \tag{3.14}$$

$$0 \leq B_i^r \leq TL \quad \forall i \in N, r \in R_s \tag{3.15}$$

$$0 \leq Q_i^r \leq Q \quad \forall i \in N, r \in R_s \tag{3.16}$$

$$x_{ij}^r \in \{0, 1\} \quad \forall i, j \in N | i \neq j, r \in R_s \tag{3.17}$$

$$y^r \in \{0, 1\} \quad \forall r \in R_s \tag{3.18}$$

The objective function of the model, as be seen at formula (3.1), aims to minimize the total cost of the assignment of orders and routes to riders, while also minimizing the number of riders. Travel cost, represented by γ_2, is multiplied by the total distance of the routes and fixed cost per rider, γ_1, multiplied by the total used riders that is assumed equal for all vehicles [16], are added together. Adjustments were made in the original objective function to localize the PDPTW. The proponents removed the calculation from last delivery to depot, as outsourced food delivery riders of Davao City, do not have depots; as such, there is no need to return to their prior location before the rider's delivery started. Constraint (3.2) ensures that all of the order requests from restaurants must be picked up while constraint (3.3) assures that once a rider is assigned to pick up the order request from node i, it must also be the one to deliver it to the customer at node j (delivery point of node i). The constraint (3.4) assures that the first arc in a route must be from the rider's initial location denoted by 0, while constraint (3.5) denotes the end of a route. The constraints (3.6) and (3.7) link together x and y decision variables:

once there is a true value for x^r_{ij}, then $y^r = 1$, which indicates the rider is assigned; on the other hand, if for all x variable of rider r will not be traversed, then y^r should be 0, meaning the rider is not assigned to any routes. Constraint (3.8) imposes the flow conservation constraint: the number of entering arc (i, j) has to be equal to the number of outgoing arcs. The constraint (3.9) ensures that node i must be served before its next node j. Constraint (3.10) updates the load for each node. The constraint (3.11) covers the precedence constraints and ensures that the route sequence is consistent. Constraints (3.12) and (3.13) assure the time window for pick-up and delivery of order: the rider r should leave node i on or after the time the order is ready a_i and should arrive at its delivery point before the time b_i. Constraint (3.14) assures that the payload of rider r never exceed its capacity Q. Finally, formulas (3.15-3.18) are the decision variables of PDPTW model.

Upon testing the model using CPLEX, the following constraints, 3.19-3.23, are added by the researchers to ensure the continuity of the routes and consistency with the solutions aimed for. Constraints (3.19) and (3.20) restrict the jump from starting location of r to delivery node directly, and pickup node to end of route directly, respectively. The constraints (3.21) and (3.22) restrict arcs that loop, or nodes that return to each other. The constraint (3.23) ensures the consistency of the x and y decision variables; once y^r is false, then there should not be any routes for r.

$$x^r_{0,d} = 0 \quad \forall d \in O_s(p, d), r \in R_s \tag{3.19}$$

$$x^r_{p,0} = 0 \quad \forall p \in O_s(p, d), r \in R_s \tag{3.20}$$

$$x^r_{i,i} = 0 \quad \forall i \in N, r \in R_s \tag{3.21}$$

$$\text{if} \quad x^r_{ij} = 1 \Rightarrow x^r_{ji} = 0 \quad \forall i \in N, \forall j \in N, r \in R_s \tag{3.22}$$

$$\text{if} \quad y^r = 0 \Rightarrow \sum_{r \in R_s} x^r_{i,j} = 0 \quad \forall i \in N, \forall j \in N \tag{3.23}$$

To extend the PDPTW model, the proponents added constraints 3.24 - 3.26 which are fairness notions in the perspective of the riders in Davao City. Fairness constraint (3.24), ensure that the riders with an idle time of greater than β_2 will be priority assigned with an order route on the time of order dispatching. Constraint (3.25) constrict the idle time of riders. Constraint (3.26) includes the maximum ride distance constraint to ensure that riders would not reach out too far from their zones.

$$\text{if} \quad \delta_r \geq \beta_2 \Rightarrow y^r = 1 \tag{3.24}$$

$$\text{if} \quad \delta_r \leq \beta_1 \Rightarrow y^r = 0 \tag{3.25}$$

$$\sum_{i \in N, j \in N | j \neq 0} d^r_{ij} x^r_{ij} \leq DL \tag{3.26}$$

Chapter 4

Results and Discussions

The proposed approach was implemented in Python with CPLEX using the DOcplex library. The experiments were carried out on a Macbook Pro 2019, 2.4 GHz Quad-Core Intel Core i5 Processor (i5-8279U), 16 GB 2133 MHz LPDDR3 Memory.

A computational analysis was derived from testing the model on two datasets with two instances. Each instance of data consists of randomly generated node locations for orders, restaurants, and riders based on the actual map of Davao City and other necessary details to be discussed further on the next sections. Moreover, the model was subjected to these model parameter values:

1. a 5-minute minimum idle time before order-rider assignment β_1,
2. a 30-minute minimum idle time for assignment prioritization β_2,
3. a 40-minute route time limit TL,
4. a 4-km and 5.5-km route distance limit or maximum ride distance value DL,
5. a 10-unit maximum capacity Q,
6. a 10 km/hr average rider's speed assumption derived from Wang & Jiang's model [49] for calculating travel time t_{ij}^r,
7. an average service time of 2.5 minutes per node [49] st,
8. γ_1 = PHP 45 based on Grab Food Philippine rate,
9. γ_2 = PHP 0.007 based on Grab Food Philippine rate

4.1 Dataset

4.1.1 Instance 1

Table 2 presents the meal orders in the first instance which is composed of 20 orders or 40 nodes in total, 20 nodes for pick-up points and 20 nodes for drop-off points. Each order has a latitude and longitude value based on Davao City's map with its equivalent

Table 2: Order Data for Instance 1.

order	latitude	longitude	x	y	restaurant	a	b	demand
1	7.095531	125.608058	788100.8347	785119.8897	r28	15	55	10
2	7.074685	125.616519	789049.1211	782818.3121	r1	0	40	3
3	7.071755	125.618612	789282.2755	782495.3935	r19	10	50	4
4	7.071106	125.613302	788695.7102	782420.2676	r14	5	45	4
5	7.052925	125.608601	788187.3458	780405.4831	r26	15	55	1
6	7.100940	125.607677	788055.3615	785718.1564	r1	10	50	2
7	7.069194	125.620536	789496.5418	782213.2616	r6	15	55	3
8	7.075784	125.603760	787637.9367	782932.0138	r10	10	50	4
9	7.083734	125.604500	787714.8402	783812.2775	r30	10	50	2
10	7.073886	125.600803	787312.2869	782720.2136	r14	5	45	1
11	7.094888	125.605406	787808.0937	785047.0322	r20	15	55	3
12	7.083422	125.598931	787099.3897	783774.2458	r26	5	45	2
13	7.084290	125.597424	786932.3092	783869.4129	r20	15	55	1
14	7.070961	125.601696	787412.7813	782397.0589	r30	5	45	5
15	7.067071	125.618257	788274.2717	780360.4996	r30	15	55	2
16	7.062153	125.611842	788539.8945	781428.6466	r25	0	40	2
17	7.063084	125.613006	788667.9727	781532.4087	r9	10	50	4
18	7.058451	125.610554	788399.7818	781018.2035	r29	5	45	3
19	7.074666	125.604199	787687.2473	782808.5760	r24	10	50	4
20	7.078720	125.600181	787240.5359	783254.6761	r1	15	55	3

Universal Transverse Mercator (UTM) x and y coordinates. An order also includes an assigned restaurant, a time window [a,b], and a demand. The demand is an order's capacity requirement in the rider's food bag.

In Table 3, the restaurants' data of Instance 1 was presented. There is a total of 30 restaurants nodes. Each restaurant node has its latitude and longitude values based on Davao City's actual map, as well as its corresponding x and y coordinates using UTM.

Additionally, Table 4 presents the riders' data for Instance 1. There is a total of 30 riders. Riders' data also include the rider's location represented by latitude and longitude values in Davao City's map with its corresponding UTM x and y coordinates. It also includes their idle times.

Table 3: Restaurant Data for Instance 1.

restaurants	latitude	longitude	x	y
r1	7.090931	125.611100	788439.9872	784612.7097
r2	7.086312	125.611749	788514.5504	784102.0120
r3	7.088889	125.612886	788638.6652	784387.8376
r4	7.084936	125.610151	788338.7816	783948.7772
r5	7.084302	125.612857	788638.3504	783880.2310
r6	7.083088	125.611865	788529.4029	783745.3010
r7	7.082619	125.613700	788732.5089	783694.6021
r8	7.081882	125.611941	788538.5878	783611.9263
r9	7.082382	125.609855	788307.6716	783665.9734
r10	7.080141	125.609020	788216.6836	783417.4585
r11	7.077560	125.605316	787808.8531	783129.5602
r12	7.081795	125.602417	787485.8047	783596.4002
r13	7.067987	125.606099	787901.3686	782070.6807
r14	7.065722	125.606665	787965.3747	781820.4368
r15	7.063125	125.611602	788512.7705	781536.0840
r16	7.079060	125.613195	788678.9777	783300.4268
r17	7.078075	125.613674	788732.4421	783191.7425
r18	7.076670	125.616588	789055.4570	783038.0083
r19	7.077099	125.618374	789252.6564	783086.6761
r20	7.078711	125.617923	789201.7846	783264.7304
r21	7.075971	125.611544	788498.3897	782957.5943
r22	7.078505	125.609135	788230.5235	783236.4790
r23	7.077055	125.608315	788140.6786	783075.4691
r24	7.077225	125.607187	788015.9009	783093.5931
r25	7.074297	125.608459	788158.3427	782770.4099
r26	7.072175	125.604578	787730.6323	782533.2133
r27	7.066666	125.609324	788258.7675	781926.5403
r28	7.066764	125.608766	788196.9112	781936.9903
r29	7.069271	125.615041	788889.0591	782218.2728
r30	7.066683	125.613155	788682.2197	781930.7717

Table 4: Rider Data for Instance 1.

rider	latitude	longitude	x	y	idle
rd1	7.067959	125.610163	788754.4800	782015.0500	10
rd2	7.091507	125.610549	788378.7086	784676.1711	35
rd3	7.071783	125.605811	787867.2224	782490.5621	2
rd4	7.069472	125.604202	787690.7418	782233.8940	6
rd5	7.066994	125.605965	787887.1704	781960.7728	10
rd6	7.065441	125.606363	787932.1400	781789.1100	2
rd7	7.065128	125.611986	788553.9331	781758.0135	18
rd8	7.070846	125.616052	788999.8859	782393.2452	4
rd9	7.077791	125.607534	788053.8859	783156.5001	3
rd10	7.078994	125.607393	788037.6296	783289.5192	40
rd11	7.075132	125.607725	788076.6950	782862.3839	5
rd12	7.081393	125.617620	789166.5840	783561.3051	19
rd13	7.083969	125.617587	789161.4173	783846.4069	1
rd14	7.087280	125.614852	788856.9278	784211.1165	7
rd15	7.085151	125.612105	788554.6376	783973.7717	10
rd16	7.085715	125.612995	788652.7247	784036.7691	4
rd17	7.087046	125.611257	788459.7615	784182.9593	15
rd18	7.085226	125.611118	788445.4743	783981.4048	2
rd19	7.084108	125.609369	788252.8524	783856.6089	3
rd20	7.066119	125.613113	788677.8938	781868.3641	2
rd21	7.078378	125.613516	788714.7875	783225.1648	7
rd22	7.078559	125.614608	788835.4451	783245.8389	45
rd23	7.090304	125.611450	788479.0781	784543.5993	17
rd24	7.066893	125.608092	788122.4330	781950.8472	6
rd25	7.067095	125.609831	788314.4493	781974.3110	3
rd26	7.066321	125.614693	788852.4600	781891.6600	40
rd27	7.078075	125.618362	789250.7300	783194.6300	1
rd28	7.091148	125.611609	788496.0800	784637.0800	12
rd29	7.090669	125.610557	788380.0900	784583.4200	2
rd30	7.082845	125.612698	788621.6400	783718.9500	10

4.1.2 Instance 2

For the second instance, Table 5 presents the meal orders in the second instance which is composed of 27 orders or 54 nodes in total, 27 nodes for pick-up points and 27 nodes for drop-off points. Each order has a latitude and longitude value based on Davao City's map with its equivalent Universal Transverse Mercator (UTM) x and y coordinates. Similarly to the Instance 1 dataset, every order in Instance 2 has an assigned restaurant, a time window [a,b], and a demand. The demand is an order's capacity requirement in the rider's food bag.

Table 5: Order Data for Instance 2.

order	latitude	longitude	x	y	restaurant	a	b	demand
1	7.050741	125.592871	786449.70	780154.12	r1	0	40	10
2	7.067896	125.582743	785319.53	782046.22	r4	5	45	4
3	7.079672	125.589223	786028.61	783353.30	r3	20	60	2
4	7.099803	125.635958	791182.24	785610.08	r6	10	50	4
5	7.072121	125.616989	789102.66	782534.91	r10	0	40	4
6	7.081022	125.603471	787602.79	783511.49	r9	20	60	2
7	7.081168	125.619392	789362.66	783537.54	r8	20	60	4
8	7.051952	125.554762	782236.09	780264.87	r11	10	50	5
9	7.051934	125.560770	782900.27	780266.53	r11	20	60	3
10	7.056637	125.541544	780772.08	780775.30	r12	20	60	4
11	7.052415	125.596046	786799.66	780341.31	r2	15	55	5
12	7.055481	125.579545	784973.62	780670.46	r2	25	65	4
13	7.102997	125.622568	789700.09	785955.12	r5	5	45	4
14	7.097163	125.626044	790087.99	785311.71	r6	25	65	2
15	7.067522	125.612998	788664.33	782023.51	r7	25	65	3
16	7.104280	125.600580	787269.00	786083.00	r5	25	65	1
17	7.043860	125.538969	780495.12	779359.94	r12	30	70	2
18	7.100540	125.605560	787822.00	785673.00	r5	30	70	3
19	7.101549	125.585253	785576.26	785771.70	r3	20	60	2
20	7.097453	125.598729	787068.41	785326.78	r5	5	45	3
21	7.075316	125.575683	784534.54	782862.96	r11	20	60	5
22	7.066923	125.613604	788731.70	781957.60	r1	25	65	4
23	7.127960	125.615868	788943.85	788713.33	r13	0	60	2
24	7.133634	125.601068	787304.43	789331.96	r14	30	70	3
25	7.118389	125.617375	789116.42	787655.15	r14	20	65	1
26	7.140277	125.626044	790060.87	790082.72	r13	5	45	2
27	7.062687	125.527897	779259.86	781436.54	r12	25	65	3

For Table 6, it exhibits the riders' data for Instance 2. There is a total of 34 riders. Riders' data also include the rider's location represented by latitude and longitude values in Davao City's map with its corresponding UTM x and y coordinates. It also includes the riders' idle times.

Moreover, Table 7 shows the restaurants' data for the second instance. There are 14 restaurants in total. Each restaurant node has its latitude and longitude values based on Davao City's actual map, as well as its corresponding x and y coordinates using UTM.

Table 6: Rider Data for Instance 2.

rider	latitude	longitude	x	y	idle
rd1	7.050030	125.585639	785650.64	780071.00	7
rd2	7.046921	125.588408	785958.66	779728.67	9
rd3	7.047930	125.595703	786764.51	779844.80	4
rd4	7.052670	125.580468	785077.38	780359.97	5
rd5	7.065320	125.592141	786360.02	781766.94	33
rd6	7.061700	125.593450	786506.96	781367.17	2
rd7	7.069566	125.578408	784839.30	782228.36	6
rd8	7.077360	125.583515	785399.06	783093.95	12
rd9	7.099581	125.627954	790297.61	785580.48	8
rd10	7.106404	125.632567	790803.24	786338.42	6
rd11	7.096089	125.633726	790937.85	785197.68	7
rd12	7.102529	125.638318	791441.40	785913.22	8
rd13	7.079617	125.608835	788196.61	783359.34	35
rd14	7.073612	125.612912	788651.04	782697.37	8
rd15	7.088518	125.608921	788200.59	784344.37	4
rd16	7.083468	125.615444	788924.80	783789.60	9
rd17	7.061066	125.561714	782999.07	781277.60	3
rd18	7.058663	125.550127	781719.65	781004.67	9
rd19	7.050078	125.541801	780804.44	780049.68	15
rd20	7.053041	125.566220	783502.07	780392.33	1
rd21	7.106470	125.645940	792281.00	786354.00	32
rd22	7.089710	125.630220	790554.00	784490.00	23
rd23	7.051356	125.589164	786039.51	780219.90	35
rd24	7.112485	125.613245	788663.61	786999.23	6
rd25	7.106362	125.611818	788509.70	786320.77	40
rd26	7.109078	125.614918	788850.67	786623.26	50
rd27	7.110620	125.606840	787956.79	786788.86	10
rd28	7.115653	125.613808	788723.86	787350.15	2
rd29	7.065155	125.530257	779519.25	781711.05	15
rd30	7.060465	125.587978	785902.81	781227.14	12
rd31	7.048007	125.544913	781149.71	779822.39	3
rd32	7.061262	125.562884	783128.29	781300.00	40
rd33	7.049327	125.590575	786196.75	779996.24	1
rd34	7.047049	125.535901	780154.05	779710.97	10

Table 7: Restaurant Data for Instance 2.

restaurant	latitude	longitude	x	y
r1	7.050136	125.588601	785978.02	780084.55
r2	7.047687	125.591240	786271.27	779815.17
r3	7.062104	125.592055	786352.50	781411.02
r4	7.063920	125.596626	786856.69	781614.79
r5	7.098581	125.631022	790637.37	785471.75
r6	7.102541	125.632975	790850.77	785911.19
r7	7.090988	125.611453	788478.94	784619.27
r8	7.086048	125.612097	788553.21	784073.02
r9	7.078032	125.614157	788785.92	783187.25
r10	7.071312	125.607977	788106.92	782439.79
r11	7.061133	125.559182	782719.13	781283.47
r12	7.049311	125.544977	781156.00	779966.73
r13	7.111600	125.613065	788644.26	786901.19
r14	7.110498	125.611595	788482.46	786778.32

4.2 Results

4.2.1 Instance 1

Phase I: Order-Rider Clustering and Allocation

For the first instance, the most optimal number of bundles is four which was derived using the Elbow Method, a known approach used for determining the most optimal value of k, the number of clusters to be generated, in k-means. The method's graphical representation can be found in the appendix section of the paper [A.1.3]. This means Instance 1 can be clustered to 4 sub-regions. Figure 4.1 shows the placement of the nodes. Blue pins represent the order nodes (delivery nodes) while red pins are restaurants.

After running constrained k-means clustering on the orders and k-means clustering on the riders who are within a 3-km range, it resulted to Bundle 1 having four orders in total namely, orders 4, 5, 10, and 12, with five possible riders: rd3, rd4, rd5, rd6, and rd24. Bundle 2 includes five orders which are orders 1, 9, 14, 15, and 18, with six possible possible riders: rd1, rd7, rd8, rd20, rd25, rd26. For Bundle 3, it contains six orders which are orders 2, 6, 7, 8, 17, and 20, with twelve possible riders: rd2, rd13, rd14, rd15, rd16, rd17, rd18, rd19, rd23, rd28, rd29, and rd30. Lastly, for Bundle 4, it contains five orders namely, orders 3, 11, 13, 16, and 19, with seven possible riders: rd9, rd10, rd11, rd12, rd21, rd22, and rd27.

Phase II: Route Assignments

Table 8 shows the order-rider assignments of the first instance, derived from the model's most optimal solutions. It displays the riders that were assigned with at least one order. Additionally, it also shows the riders' assigned cluster, the riders' routes from one node to another. The table also includes the rider's load, the total cost of each route in PHP, the total distance in meters traveled by each rider, and the total run times of the calculation in seconds.

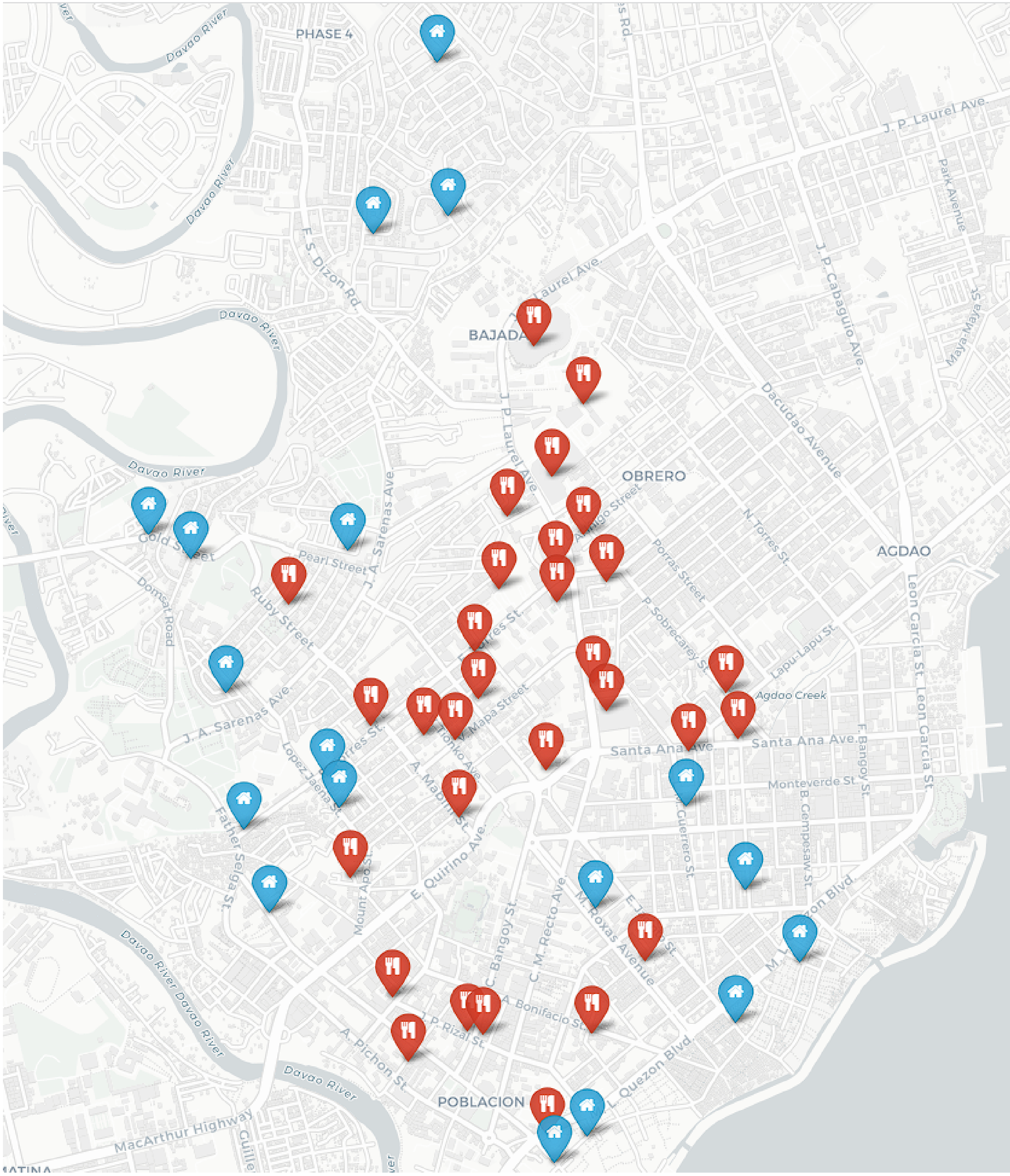

Figure 4.1: Nodes of Instance 1.

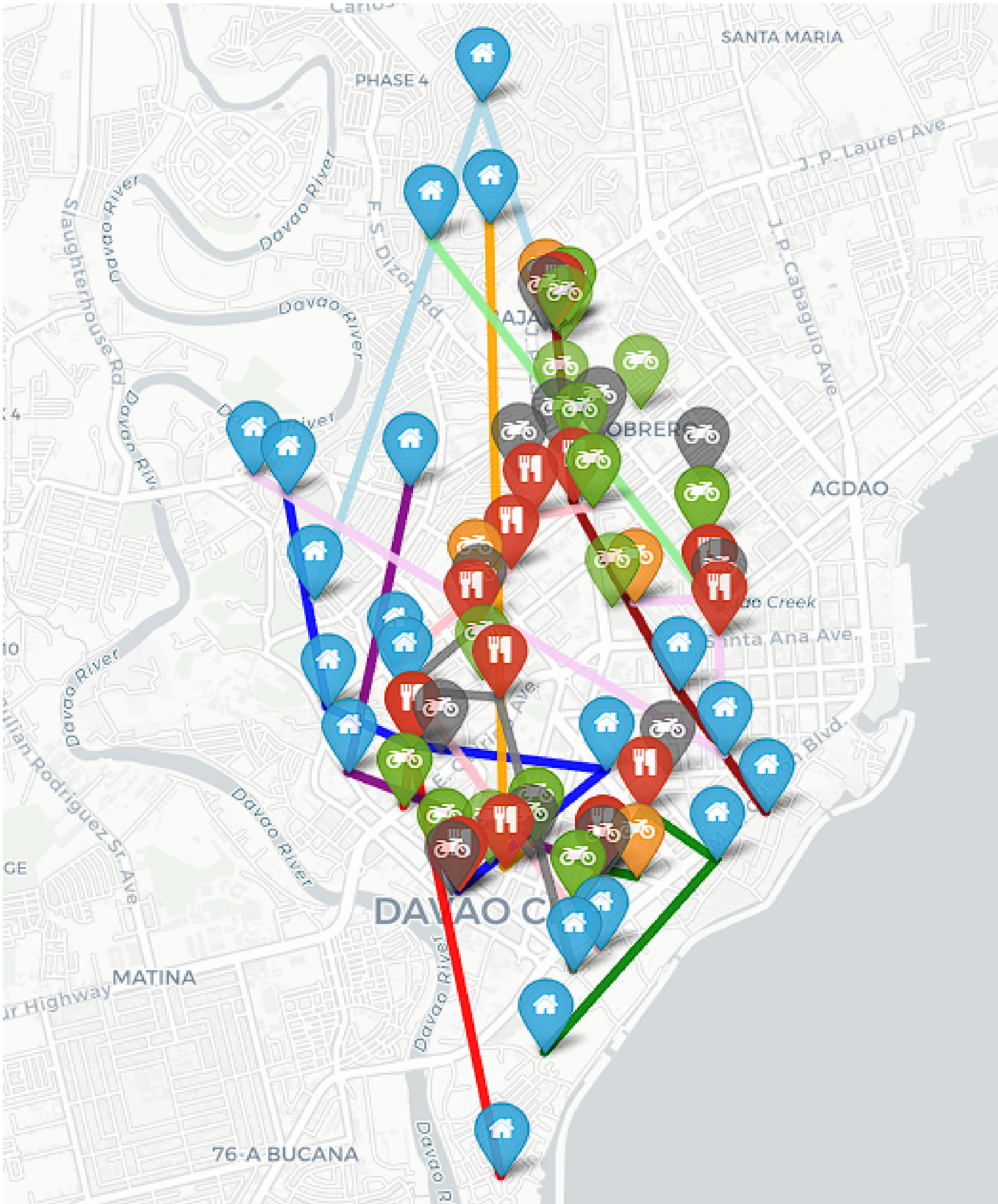

Figure 4.2: Routes for Instance 1.

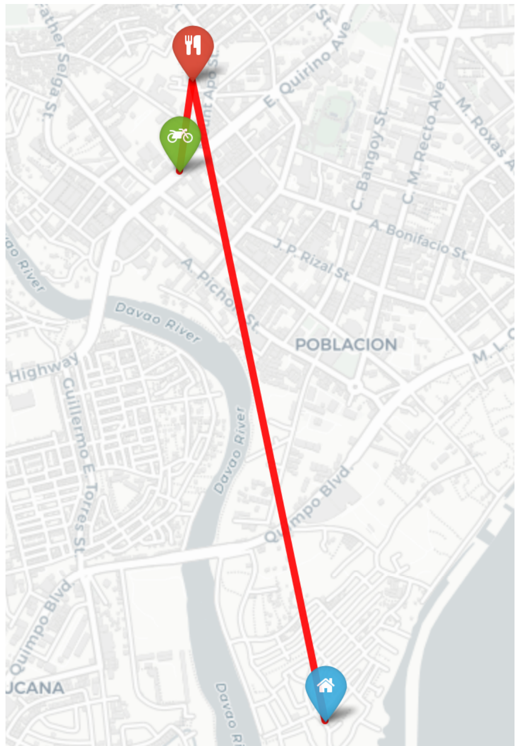

Figure 4.3: Route of rd4.

From Quirino Ave. to Purok 13 Nodes: 0-5-105

In Table 8, row 1 shows the assignment for rider 'rd4'. The route column displays the nodes that rider 'rd4' will traverse, 0-5-105. The route starts from 0 which denotes the rider's initial location when the assignment was given. Nodes < 100 are the pick-up nodes and the delivery point of those pick-up (p) nodes are denoted by $p + 100$. As so, order 5's delivery point is 105. From the initial location of rd4, the rider will travel to node 5 to pick-up the delivery with load demand of 1 unit, in ratio with the rider's food bag. The load of the rider while on route can be seen on the load column of the Table 8. For rd4 it is 0-1-0. As a rider starts his route, the load of his bag is 0, and ends at 0. When rd4 travels to delivery points, the load of the rider gets lighter by how much was the demand (quantity) ordered by the customer as the orders at the drop-off nodes are now removed from the delivery bag. The riders' load is part of the constraints that the model has to consider, as the courier should not be assigned to multiple orders which could not add up in the rider's delivery bag. The total cost for rd4 is PHP 62.34 derived from the base fare plus total distance traveled, with a total distance traveled of 2476.61 meters. The total distance traveled is below the set distance limit DL, which means the fairness constraint (3.26) is followed. From Table 4, we could see that rd4 has an idle time of 6 minutes before the rider got assigned. This means that the model results is consistent with our fairness constraint on idle times as our set β_1 is 5 minutes. Figure 4.3 displays the route sequence of rd4 from Quirino Ave. to Purok 13.

For Instance 1 from the Table 8, the model returns a total minimum cost of PHP 728.22 for all the riders' routes and a minimum distance of about 33314.14 meters. The

Table 8: Route Results for Instance 1.

Cluster	Rider	Route	Load	Cost	Distance	Run Time
1	rd4	0-5-105	0-1-0	62.34	2476.61	0.2148
	rd5	0-10-4-104-12-110-112	0-1-5-1-3-2-0	70.26	3608.70	
2	rd7	0-9-14-114-109	0-2-7-2-0	66.09	3012.76	0.2873
	rd1	0-1-101	0-10-0	68.70	3385.59	
	rd26	0-15-18-115-118	0-2-5-3-0	60.66	2237.0	
3	rd28	0-2-7-102-107	0-3-6-3-0	64.23	2746.68	4.7201
	rd30	0-17-8-108-117	0-4-8-4-0	66.52	3074.28	
	rd2	0-20-6-106-120	0-3-5-3-0	71.96	3851.0	
4	rd22	0-13-3-103-113	0-1-5-1-0	71.57	3863.78	0.4084
	rd10	0-19-119-16-116	0-4-0-2-0	62.49	2498.18	
	rd12	0-11-111	0-3-0	63.39	2559.56	
Total				728.22	33314.14	5.6306

computational times for Bundles 1 to 4 in the first instance are about 0.21 seconds, 0.29 seconds, 4.72 seconds, and 0.41 seconds, respectively. Instance 1 has a total calculation time of about 5.63 seconds. The mapping for Instance 1 is displayed in Figure 4.2. The gray rider pins are those who were excluded by the model in phase II due to the fairness constraints. They are the ones who have only been idle for less than 5 minutes because they just finished an order delivery. Orange rider pins are those riders that are prioritized for they have been idle for the longest time, while the green rider pins denotes the rest of viable riders that can be chosen. The mapping visualization for this test instance can be downloaded and found at 4.2.2.

4.2.2 Instance 2

Phase I: Order-Rider Clustering and Allocation

There are five bundles in total that resulted from the Elbow Method, the approach used to solve the most optimal number of clusters for Instance 2 [A.1.3]. Utilizing constrained k-means and standard k-means on orders and riders, respectively, resulted to Bundle 1 having seven orders namely, orders 1, 2, 3, 11, 12, 19, and 22, with eleven possible riders: rd1, rd2, rd3, rd4, rd5, rd6, rd7, rd8, rd23, rd30, and rd33. Bundle 2 has four orders namely, orders 23, 24, 25, and 26, with five potential riders: rd24, rd25, rd26, rd27, and rd28. For Bundle 3, there are seven orders specifically, orders 4, 13, 14, 15, 16, 18, and 20, with six candidate riders: rd9, rd10, rd11, rd12, rd21, and rd22. Bundle 4 consists of six orders namely, orders 8, 9, 10, 17, 21, and 27, with eight probable riders: rd17, rd18, rd19, rd20, rd29, rd31, rd32, and rd34. Lastly, Bundle 5 includes three orders which are orders 5, 6, and 7, with four possible riders: rd13, rd14, rd15, and rd16. [Note: Similar legend representations previously mentioned in the first instance apply on Figures 4.5 and 4.6 on Instance 2.]

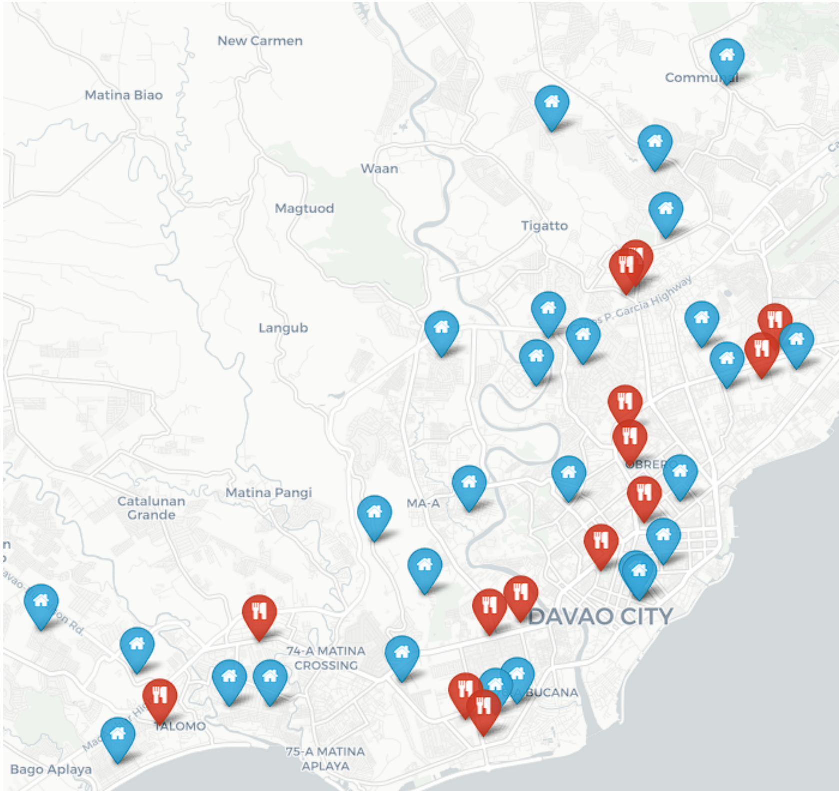

Figure 4.5: Nodes of Instance 2.

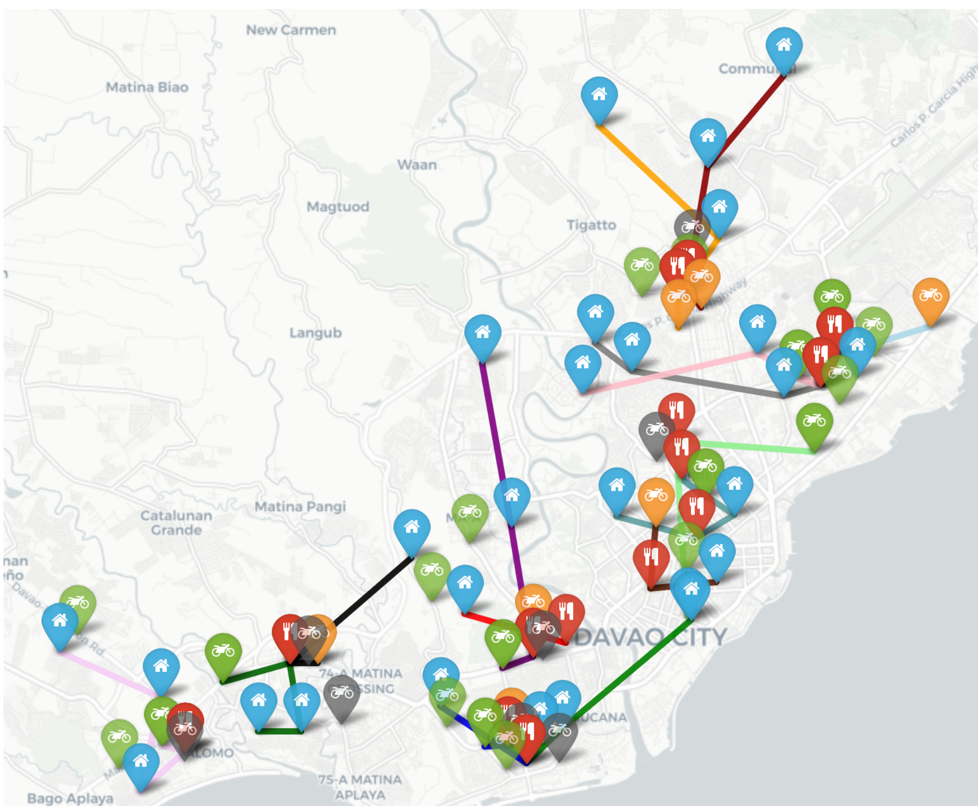

Figure 4.6: Routes for Instance 2.

Phase II: Route Assignments

Table 9 presents the order-rider assignments of the second instance which are the most optimal solutions returned by the model. It includes the riders that were assigned with at least one order. It also shows the rider's inter-node routes which was demonstrated in Figure 4.6. Moreover, it includes the rider's assigned cluster, riders' load, the total cost of each route, the total distance traveled, and the run time in finding the optimal solution in each bundle. The model returns a total minimum cost of about PHP 1038.26 for all the riders' routes and a minimum distance of around 51895.38 meters. The computational times of Bundles 1 to 5 in the second instance are around 3.33 seconds, 0.14 seconds, 8.74 seconds, 1.62 seconds, and 0.15 seconds, respectively. This results to a total run time of 13.99 seconds.

Table 9: Route Results for Instance 2.

Cluster	Rider	Route	Load	Cost	Distance	Run Time
1	rd23	0-22-11-111-122	0-4-9-4-0	71.66	3809.11	3.3321
	rd1	0-1-101-12-112	0-10-0-4-0	64.18	2739.97	
	rd5	0-2-102	0-4-0	59.8	2114.7	
	rd30	0-19-3-103-119	0-2-4-2-0	79.39	4912.26	
2	rd25	0-25-24-125-124	0-1-4-3-0	73.05	4006.65	0.1442
	rd26	0-26-23-123-126	0-2-4-2-0	72.63	3947.62	
3	rd21	0-4-104	0-4-0	58.61	1944.24	8.7434
	rd9	0-13-20-113-120	0-4-7-3-0	73.8	4114.38	
	rd22	0-15-115	0-3-0	77.75	4678.81	
	rd10	0-14-16-18-114-118-116	0-2-3-6-4-1-0	76.3	4471.44	
4	rd19	0-27-10-17-117-110-127	0-3-7-9-7-3-0	75.44	4348.58	1.6194
	rd32	0-21-121	0-5-0	64.7	2814.16	
	rd18	0-8-9-109-108	0-5-8-5-0	64.13	2733.16	
5	rd16	0-7-107-6-106	0-4-0-2-0	68.36	3337.32	0.1478
	rd13	0-5-105	0-4-0	58.46	1922.98	
Total				1038.26	51895.38	13.99

To access the map visualization of the results in this chapter, the following can be downloaded once clicked:

1. Instance One Map
2. Instance Two Map

Snapshots of the results are also available at Appendix A.1.

As seen in the model's results from the two instances above, it indeed exhibited the conditions for Savelsbergh's [40] PDPTW. The proposed model constructed routes for a group of vehicles to fulfill order requests. It also considered the vehicle and order's capacity, the start location, end location, and the size of load to be delivered. Additionally, orders were transported by one rider only which satisfies Savelsbergh's condition about the PDPTW not permitting "transshipments" or the transferring of loads from one transporter to another. And, time window was also one of the considerations. All these are the aspects that make up a Pick-up and Delivery Problem with Time Windows (PDPTW) as per literature.

4.3 Comparison with PDPTW

This section presents a comparison between the performance of the Base PDPTW Model and the Proposed Fair PDPTW Model. Areas of comparison includes: (1) run time, (2) fairness performance metrics, (3) average total cost of per route.

4.3.1 Run Time

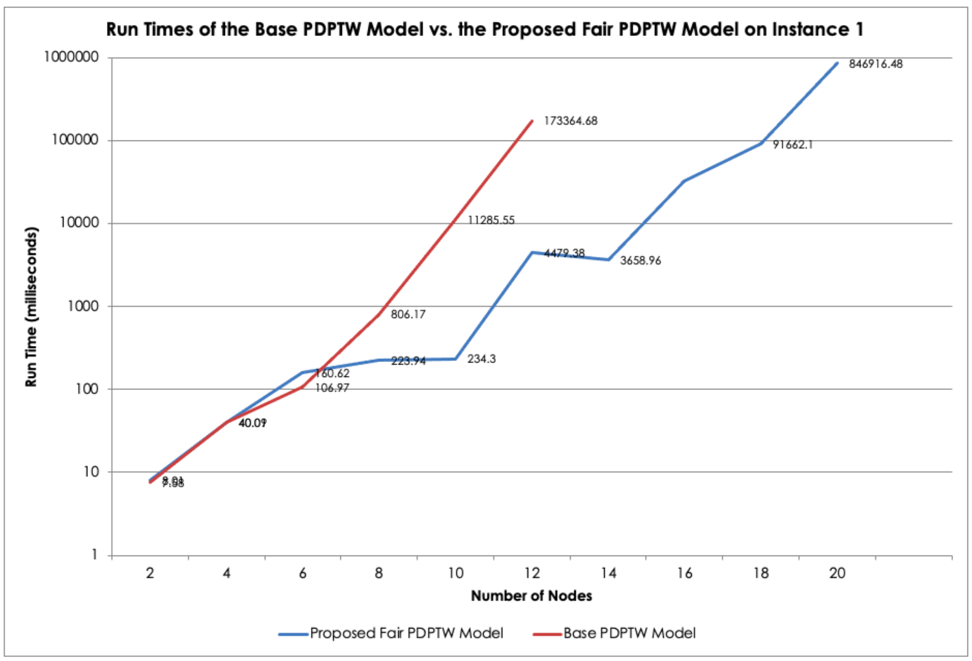

Figure 4.7: Run Times of the Base Model and Fair PDPTW Model on Instance 1.

To compare the performance of the two models in terms of run time, the proponents tested both the base optimisation model and the proposed model on different number of nodes on each of the instance. The models were first tested with two random nodes and then the node count was scaled up by intervals of 2. Generally, the proponents only set a maximum waiting time of 1 hour. Once the model ran past sixty minutes of run time in looking for an optimal solution, the proponents stopped running the program. Then, the proponents recorded the computational run times (in milliseconds) on each. It can be seen that in both the original PDPTW model and the proponents' fair PDPTW model in the two instances, there was indeed an exponential increase of run times as nodes increased.

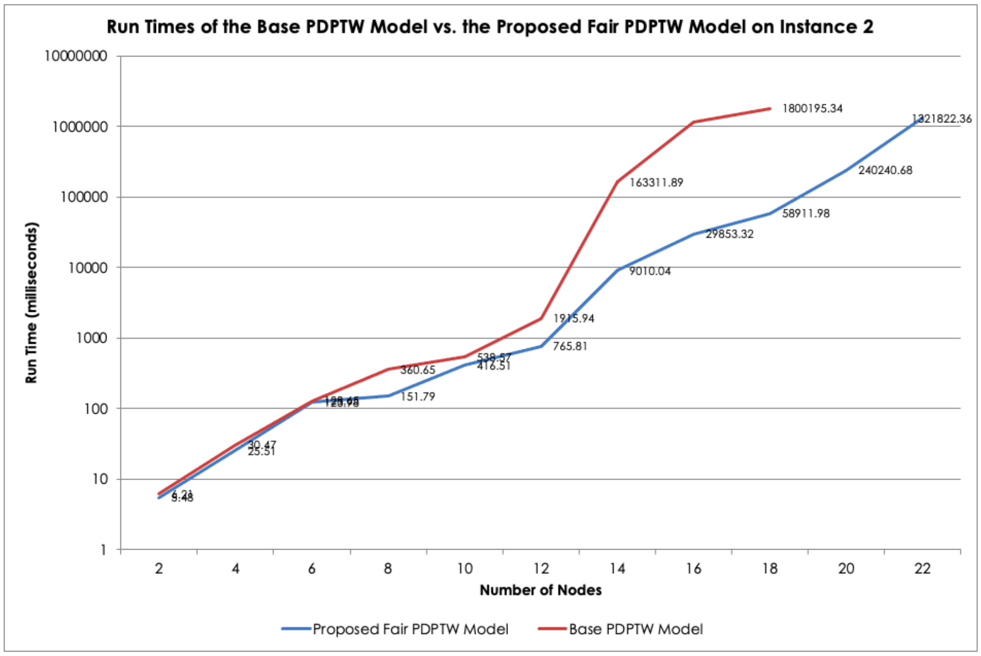

Figure 4.8: Run Times of the Base Model and Fair PDPTW Model on Instance 2.

For Instance 1, as seen in Figure 4.7, in the base PDPTW model, running 2 nodes took 7.58 milliseconds (ms). For 4 nodes, the base model spent 40.09ms before it found an optimal solution. It then took 106.97ms for 6 nodes. For 8 nodes, the run time was 806.17s (s), 11.29 seconds for 10 nodes, and about 3 minutes for 12 nodes. By the time the model was fed with 14 nodes, it already went over an hour and yet it still did not find an optimal solution. As seen from the results, a 12-node increase from 2 nodes to 14 nodes resulted to a massive increase of run time from 7.58ms to more than an hour (3,600,000ms). Meanwhile, in the proposed fair PDPTW model, it took 8.01ms before it found an optimal solution for 2 nodes. It then took 40.01ms for 4 nodes. For 6 nodes, it spent 160.62ms before the model finished running. As for 8 nodes, it had a run time of 223.94ms, 234.3ms for 10 nodes, 4.47s for 12 nodes, 3.66s for 14 nodes, 32.04s for 16 nodes, 91.67s for 18 nodes, and around 14 minutes for 20 nodes. Once 22 nodes were fed into the proposed model, it already spent more than an hour looking for the most optimal solution to the instance. It can be observed how there was such a tremendous rise in run times from 8.01ms to more than an hour (3,600,000ms), over a 20-node increase from 2 nodes to 22 nodes.

For Instance 2, Figure 4.8 presents that for 2 nodes, it took 6.21ms for the base PDPTW model to find an optimal solution. It then spent 30.47ms to solve 4 nodes, 128.65ms for 6 nodes, 360.65s for 8 nodes, 538.57s for 10 nodes, 1.92 minutes for 12 nodes, around 2.72 minutes for 14 nodes, around 19.24 minutes for 16 nodes, and about 30 minutes for 18 nodes. By the time 20 nodes were tested, the base model could not find an optimal solution within the 60-minute maximum time limit anymore. On the other hand, the fair PDPTW model generally returned the most optimal solutions faster. The proposed model only took 5.48ms for 2 nodes, 25.51s for 4 nodes, 123.98s for 6 nodes, 151.70s for 8 nodes, 416.51s for 10 nodes, 765.81s for 12 nodes, about 9s for 14 nodes, about 30s for 16 nodes, around 59s for 18 nodes, around 4 minutes for 20 nodes, and about 22 minutes for 22 nodes. For 24 nodes, the fair PDPTW model was not able to find an optimal solution within one hour. It can be observed that in second instance, there was a significant difference with the number of nodes the base model and the proposed model could solve within 60 minutes which were 18 and 22 nodes, respectively.

These results reveal that PDPTWs are indeed NP-hard, as demonstrated by how the instances were solved with exponential time as nodes increased. The run-time increase is more massive and significant than the increase in the nodes. Also, the proponents observed that the proposed fair PDPTW model generally returns optimal solutions faster than the base model and is more scalable because the proposed fair PDPTW model can handle it better as the nodes increase. Given a waiting time of sixty minutes, the proposed fair PDPTW model found an optimal solution for more nodes for both instances. For Instance 1, the proposed fair model found an optimal solution within 14 minutes for 20 nodes, while the base model only found a solution for 12 nodes within the given 1-hr time frame. In Instance 2, the fair model was able to return an optimal solution for 22 nodes in 22 minutes, while the base model only solved 18 nodes within the one-hour time frame (in 30 minutes).

4.3.2 Fairness Performance Metrics

Idle Time

Idle time was one of the riders' concerns with regard to fairness. According to the riders [Appendix C.1.1], they would want a dispatch system that would prioritize those who have been waiting for orders for a longer time since the current system assigns orders to riders randomly. Also, as per VRP literature, researchers (i.e., Ghiani et al.[15]) have pointed out how a long waiting time is deemed "unacceptable." And so, proponents of this study devised a model that reduces the riders' total idle time. The test conducted on Instance 1 showed that compared to the base PDPTW model, the total idle time of the unassigned riders for the fair PDPTW model is lower. This means that the cumulative idle time of the riders was reduced in the proposed fair model. As shown on Table 10, the fair PDPTW model results to a total idle time of 96 minutes while the base model's total unassigned riders' waiting time is 313 minutes. Similarly on Instance 2, Table 11 presents a total idle time of 116 minutes for the fair PDPTW model, while the base PDPTW model has a total waiting time of 272 minutes. On average, based on the idle time results on both Instance 1 and Instance 2, the proposed fair PDPTW model reduces idle time by around 63%. Thus, the idle time fairness constraint in the proposed model is effective.

Table 10: Total Idle Time of the Fair PDPTW Model vs. the Base PDPTW Model on Instance 1.

Proposed Fair Model		**PDPTW Model**	
Idle Riders	Idle Time	Idle Riders	Idle Time
rd3	2	rd1	10
rd6	2	rd2	35
rd8	4	rd4	6
rd9	3	rd5	10
rd11	5	rd7	18
rd13	1	rd10	40
rd14	7	rd11	5
rd15	10	rd12	19
rd16	4	rd13	1
rd17	15	rd14	7
rd18	2	rd15	10
rd19	3	rd16	4
rd20	2	rd17	15
rd21	7	rd18	2
rd23	17	rd19	3
rd24	6	rd21	7
rd25	3	rd22	45
rd27	1	rd23	17
rd29	2	rd24	6
		rd25	3
		rd26	40
		rd30	10
Total	96	**Total**	313

Table 11: Total Idle Time of the Fair PDPTW Model vs. the Base PDPTW Model on Instance 2.

Proposed Fair Model		**PDPTW Model**	
Idle Riders	Idle Time	Idle Riders	Idle Time
rd2	9	rd1	7
rd3	4	rd2	9
rd4	5	rd3	4
rd6	2	rd4	5
rd7	6	rd7	6
rd8	12	rd8	12
rd11	7	rd12	8
rd12	8	rd13	35
rd14	8	rd15	4
rd15	4	rd16	9
rd17	3	rd18	9
rd20	1	rd19	15
rd24	6	rd20	1
rd27	10	rd21	32
rd28	2	rd22	23
rd29	15	rd25	40
rd31	3	rd27	10
rd33	1	rd28	2
rd34	10	rd29	15
		rd30	12
		rd31	3
		rd33	1
		rd34	10
Total	116	**Total**	272

Distance

One of the goals of the proposed fair model was to reduce the total distance traveled by the riders. As seen in Table 12, the average distance of the generated routes by the fair PDPTW model for Instance 1 is 3028.56 meters which is smaller than the average distance of the base model which is 3978.41 meters. Likewise, Table 13 presents an average distance of 3459.69 meters and 4842.21 meters for the proposed model and the base model, respectively. In average, the proposed fair PDPTW model reduced the average total distance traveled by around 26%. The reduced distance returned by the proposed fair PDPTW model was due to the fact that the routes were balance because of the maximum distance constraint.

Table 12: Average Distance of the Proposed and Base Model on Instance 1.

Instance 1	**Proposed Fair Model**		**PDPTW Model**	
Bundle	Rider	Distance	Rider	Distance
1	rd4	2476.61	rd3	2317.84
	rd5	3608.70	rd6	3493.83
2	rd7	3012.76	rd8	4656.45
	rd1	3385.59	rd20	4859.28
	rd26	2237.0		
3	rd2	3851.0	rd28	4802.22
	rd30	3074.28	rd29	4122.68
	rd28	2746.68		
4	rd12	2559.56	rd27	5200.35
	rd10	2498.18	rd9	2374.62
	rd22	3863.78		
Avg Distance		3028.56		3978.41

Table 13: Average Distance of the Proposed and Base Model on Instance 2.

Instance 2	**Proposed Fair Model**		**PDPTW Model**	
Bundle	Rider	Distance	Rider	Distance
1	rd5	2114.70	rd5	7843.22
	rd23	3809.11	rd23	2561.08
	rd1	2739.97	rd6	4587.20
	rd30	4912.26		
2	rd25	4006.65	rd24	3701.52
	rd26	3947.62	rd26	3947.85
3	rd9	4114.38	rd9	3997.14
	rd10	4471.44	rd10	2010.58
	rd21	1944.24	rd11	7968.97
	rd22	4678.81		
4	rd32	2814.16	rd32	4507.72
	rd19	4348.58	rd17	6513.91
	rd18	2733.16		
5	rd16	3337.32	rd14	5625.12
	rd13	1922.98		
Avg Distance		3459.69		4842.21

4.3.3 Cost

The objective function of both the base and the proposed model is cost minimization. In terms of the total cost of all the routes generated by the models, test results in Table 14 and Table 15 reveal that the proposed fair PDPTW model yields less overall cost than the base PDPTW model for both Instance 1 and Instance 2. For Instance 1 in Table 14, the average cost of each route produced by the proposed model is about PHP 66.20 while for the base model, it has an average cost of 72.85. Comparably, for Instance 2 presented in Table 15, it shows an average cost of PHP 69.22 for the fair PDPTW model and a PHP 78.89 average cost for the base model. On average, the average cost of the proposed model decreased by about 11% compared to the average cost from the base model. The reduced average cost produced by the proposed fair PDPTW model is expected since the cost is dependent on the distance. As the distance in the proposed fair PDPTW model decreased, the cost would also decrease.

Table 14: Average Cost of the Proposed and Base Model on Instance 1.

Instance 1	**Proposed Fair Model**		**PDPTW Model**	
Bundle	Rider	Cost	Rider	Cost
1	rd4	62.34	rd3	61.22
	rd5	70.26	rd6	69.46
2	rd7	66.09	rd8	77.60
	rd1	68.70	rd20	79.01
	rd26	60.66		
3	rd2	71.96	rd28	78.62
	rd30	66.52	rd29	73.86
	rd28	64.23		
4	rd12	62.92	rd27	81.40
	rd10	62.49	rd9	61.62
	rd22	72.05		
Avg Cost		66.20		72.85

Table 15: Average Cost of the Proposed and Base Model on Instance 2.

Instance 2	**Proposed Fair Model**		**PDPTW Model**	
Bundle	Rider	Cost	Rider	Cost
1	rd5	59.80	rd5	99.9
	rd23	71.66	rd23	62.93
	rd1	64.18	rd6	77.11
	rd30	79.39		
2	rd25	73.05	rd24	70.91
	rd26	72.63	rd26	72.63
3	rd9	73.80	rd9	72.98
	rd10	76.30	rd10	59.07
	rd21	58.61	rd11	100.78
	rd22	77.75		
4	rd32	64.7	rd32	76.55
	rd19	75.44	rd17	90.60
	rd18	64.13		
5	rd16	68.36	rd14	84.38
	rd13	58.46		
Avg Cost		69.22		78.89

Chapter 5

Conclusion and Recommendations

5.1 Conclusion

This paper presented a fair two-stage solution approach for PDPTW in the context of meal deliveries in Davao City. But before the actual implementation of the two-stage solution approach, the proponents conducted an in-depth interview to understand the local context of PDPTW and decide upon the fairness constraints that will be added to the model and the value to use for the parameters. The participants' responses mainly exemplified just what the literature describes PDPTW to be. Thus, the Davao City version of PDPTW also consists of instances where a fleet of riders were to deliver orders in a given time window considering capacity constraints without "transshipment," from the orders' starting points to delivery points, while considering the orders' loads. The fairness constraints determined were the fair management of idle times through rider prioritization and the controlling of maximum ride distance.

The first phase of the solution involved the bundling of orders and riders wherein the orders were clustered first, followed by allocating a group of riders to each order cluster based on the order clusters' centers. PDPTW formulated modified model by Grandinetti et al. [16] from the research of Ropke et al. [39], was then utilized and modified by the proponents to localize the model and introduce fairness aspects for the benefit of one of the stakeholders in meal deliveries, the delivery riders. The second phase was the actual running of the proposed modified PDPTW optimization model in the CPLEX solver to acquire the most optimal assignments with the least cost, having the additional constraints considered.

Results show that the proposed fair PDPTW model was generally faster in finding the most optimal solutions than the base PDPTW model. It was also more scalable since in Instance 1, it was able to solve 20 nodes in 14 minutes, while the base model was not able to find an optimal solution for over an hour for 14 nodes. Meanwhile for Instance 2, the proposed fair PDPTW model found an optimal solution for 22 nodes in 22 minutes, while the base model only solved 18 nodes within the one-hour time frame. Also, the proposed two-stage solution approach was found to be more efficient than not running a prior clustering step since it only took an overall model run time of 5.63 seconds for

Instance 1 and 13.99 seconds for Instance 2 for the proposed model to find the most optimal route assignments because of the bundling stage. Without the clustering step, both the original PDPTW model and the proposed fair PDPTW model never found an optimal solution in more than 48 hours because of the large number of nodes processed all at once. This means that the clustering step indeed made the solution approach efficient. The struggle of the model to find an optimal solution quickly for a large number of nodes is attributed to the NP-hard nature of VRPs. The trend of the model's run times from the node increase presented above, exhibited an exponential rise despite the minimal increase in the number of nodes.

In summary, the proposed PDPTW model was found to be more efficient and effective in terms of fairness as shown in Chapter 4. The fair PDPTW model returns optimal routes with lesser average distance and cost, in a shorter period of time, compared to the base model. It also yields lesser idle or waiting time. In conclusion, experimentation and test results revealed that the proposed fair PDPTW model outperforms the base PDPTW model in terms of algorithm speed, scalability, fairness, and cost minimization.

5.2 Recommendations

There are several directions for future research. Firstly, further studies could apply more fairness notion constraints that may fit their context. Researchers of this study only focused on the riders for the fairness constraint so, future studies may also consider customers and companies' perspectives. Additionally, actual road networks maybe taken into account in further studies. Davao food delivery services are facing complex instances in practice, such as traffic congestion, road complexity, or severe weather conditions, which may be considered in the routing optimization. Also, despite the fixed number of meal orders in this study, the real demand for Davao takeout meals is changing in real-time since the customers can place their online orders through meal delivery platforms anytime and anywhere. Thus, the stochasticity or uncertainty of new meal orders may be considered in future research on delivery scheduling optimization. Additionally, customer satisfaction and delivery cost are crucial for Davao meal delivery platforms. Therefore, multiple objectives might also be applied in future studies on meal delivery routing optimization. Researchers of future studies may also consider customers' time sensitivity coefficients and set different ones for each meal order in future realistic studies. Lastly, in-depth studies on population-based algorithms (i.e., ant colony optimization) and heuristics for VRPs may further improve meal delivery optimization.

Appendix A

Order-Rider Assignment Route Solutions

A.1

The following are the graphical representations for Chapter 4:

A.1.1 Instance 1 Clusters 1-4

A.1.2 Instance 1 Routes

A.1.3 Instance 1 Elbow Method for KMeans Clustering

A.1.4 Instance 2 Elbow Method for KMeans Clustering

A.1.5 Instance 2 Clusters 1-5

A.1.6 Instance 2 Routes

A.1.7 Base PDPTW Simulation

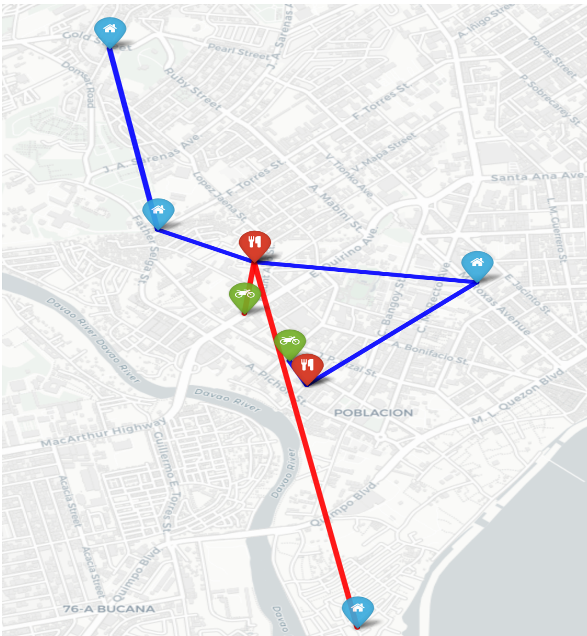

Figure A.1: Instance 1 Bundle 1

rd4 (Red): 0-5-105, rd5 (Blue): 0-10-4-104-12-110-112

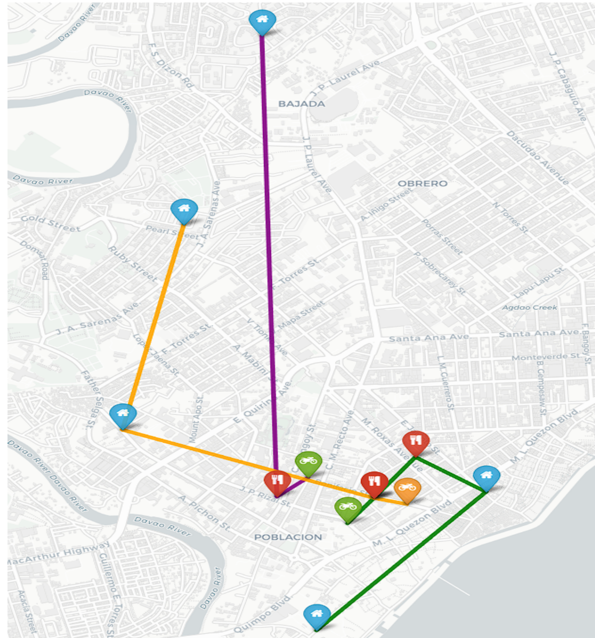

Figure A.2: Instance 1 Bundle 2

rd26 (Green): 0-15-18-115-118, rd1 (Purple): 0-1-101, rd7 (Orange): 0-14-9-114-109

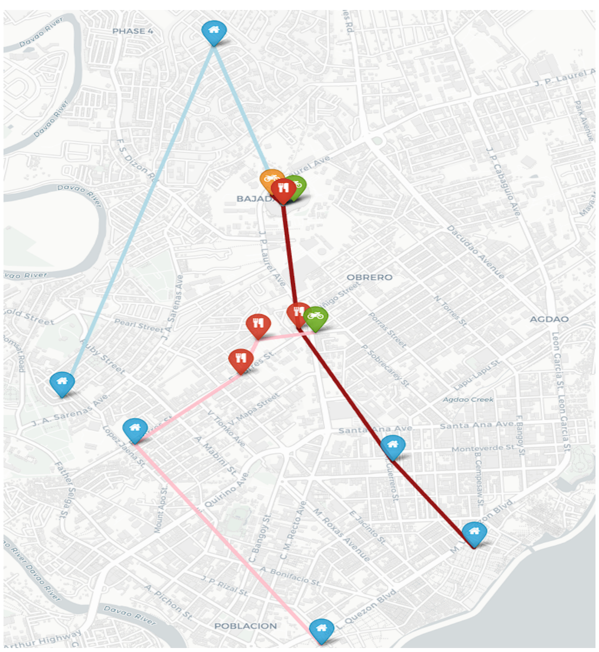

Figure A.3: Instance 1 Bundle 3

rd28 (Dark Red): 0-2-7-102-107, rd30 (Pink): 0-17-8-108-117, rd2 (Light Blue): 0-20-6-106-120

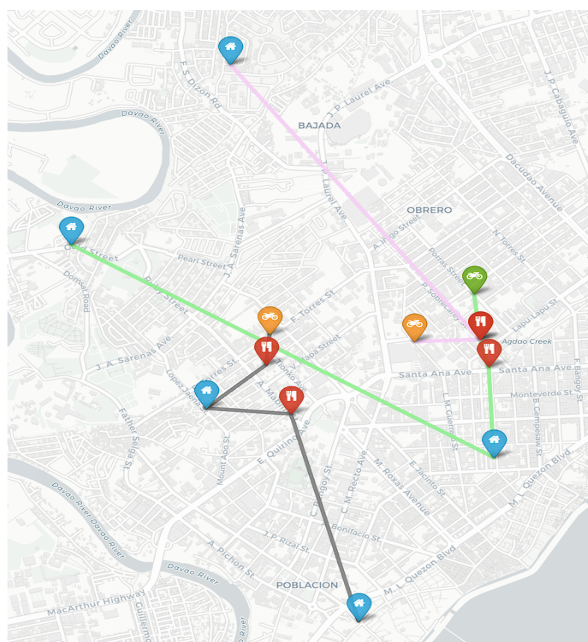

Figure A.4: Instance 1 Bundle 4

rd22 (Light Green): 0-13-3-103-113, rd10 (Gray): 0-19-119-16-116, rd12 (Light Pink): 0-11-111

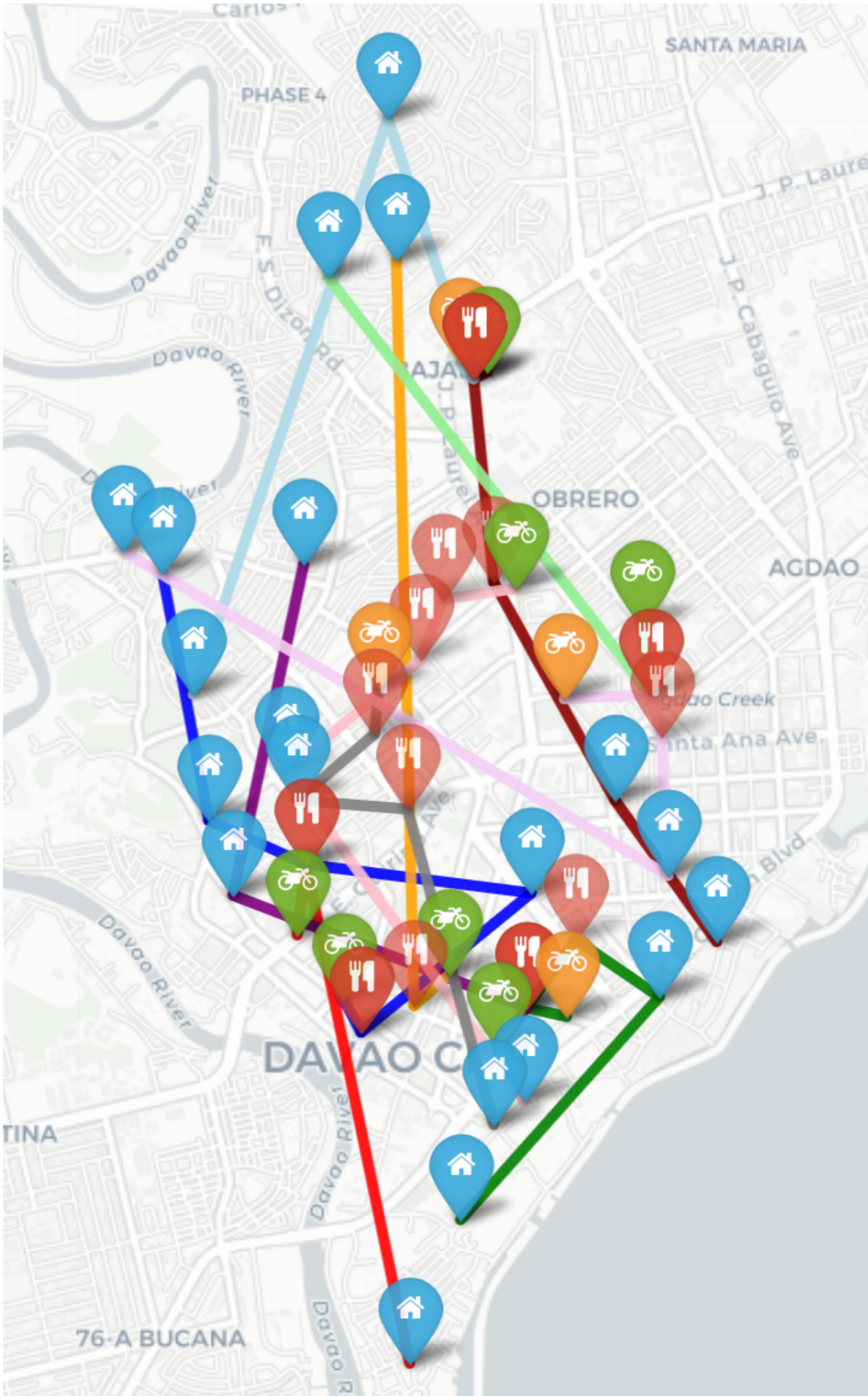

Figure A.5: Instance 1 Routes

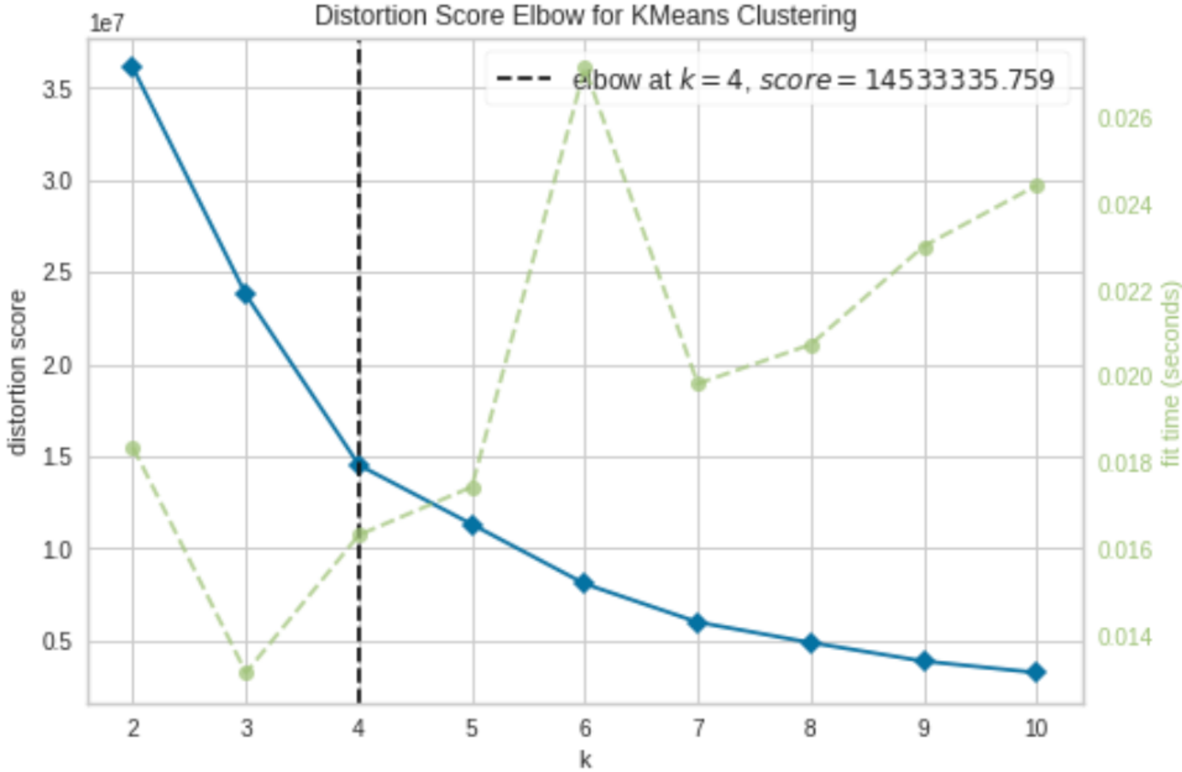

Figure A.6: Elbow Method of Instance 1

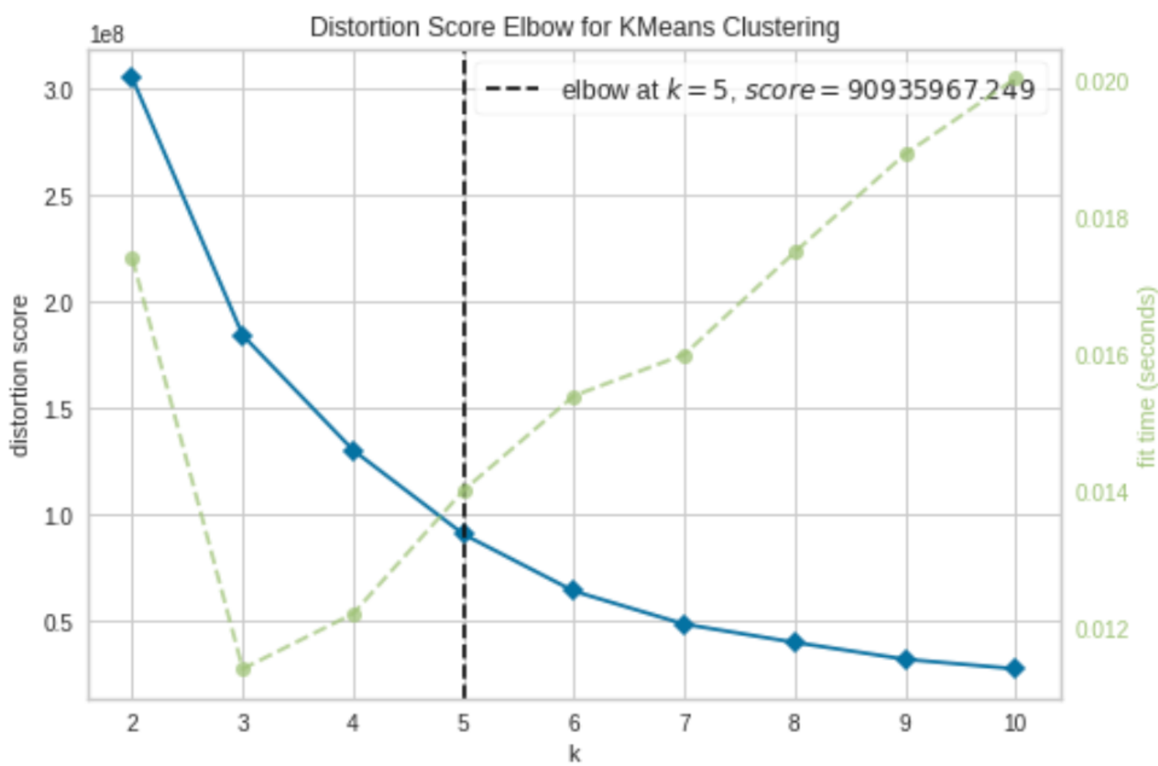

Figure A.7: Elbow Method of Instance 2

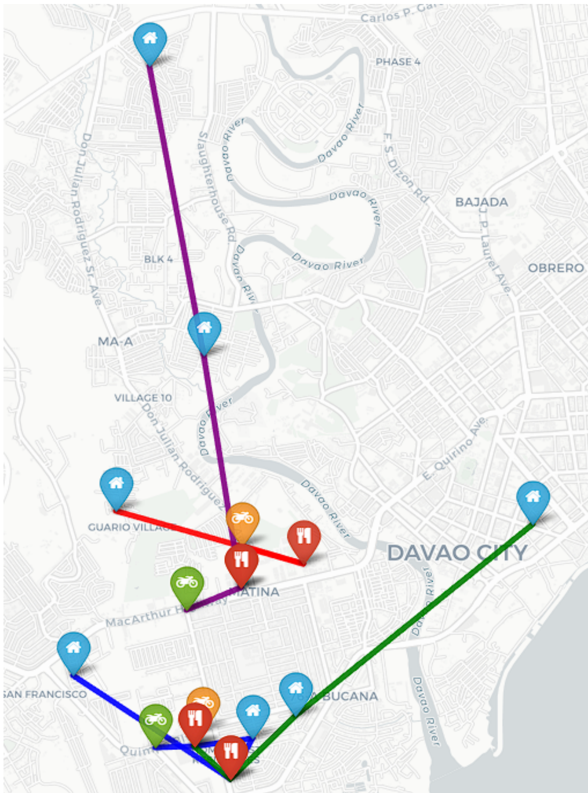

Figure A.8: Instance 2 Bundle 1

rd5 (Red): 0-2-102, rd1 (Blue): 0-1-101-12-112, rd23 (Green): 0-22-11-111-122, rd30 (Purple) (0-19–103-119)

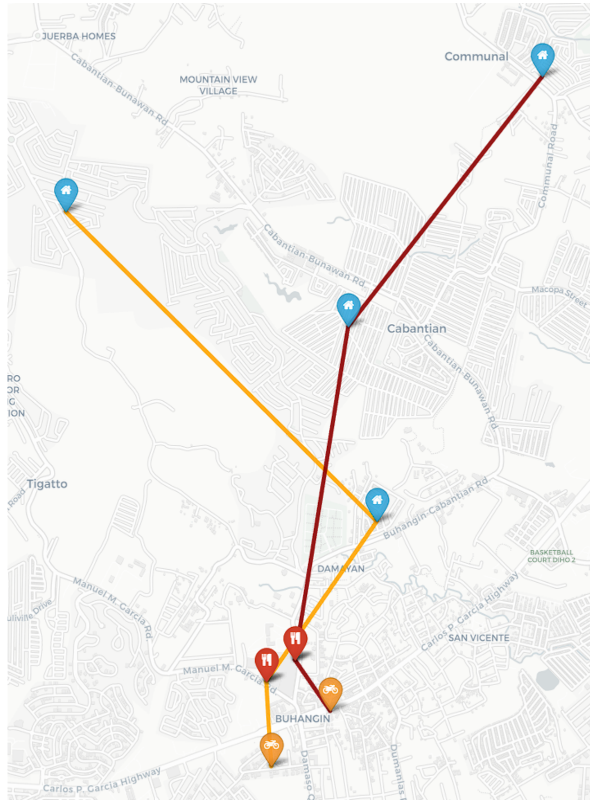

Figure A.9: Instance 2 Bundle 2

rd25 (Orange): 0-25-24-125-124, rd26 (Dark Red): 0-26-23-123-126

Table 16: Route Results for Base PDPTW Instance 1

Cluster	Rider	Route	Load	Cost	Distance	Run Time
1	rd6	0-4-10-104-12-110-112	0-4-5-1-3-2-0	69.46	3493.83	0.7067
	rd3	0-5-105	0-1-0	61.22	2317.84	
2	rd20	0-15-115-1-101	0-2-0-10-0	79.01	4859.27	12.8463
	rd8	0-18-14-9-118-114-109	0-3-8-10-7-2-0	77.6	4656.45	
3	rd29	0-2-7-17-102-107-117	0-3-6-10-7-4-0	73.86	4122.68	189.5696
	rd28	0-6-20-106-8-108-120	0-2-5-3-7-3-0	78.62	4802.22	
4	rd9	0-19-119-16-116	0-4-0-2-0	61.62	2374.62	17.2191
	rd27	0-3-103-11-13-111-113	0-4-0-3-4-1-0	81.4	5200.35	
Total				582.79	31827.26	

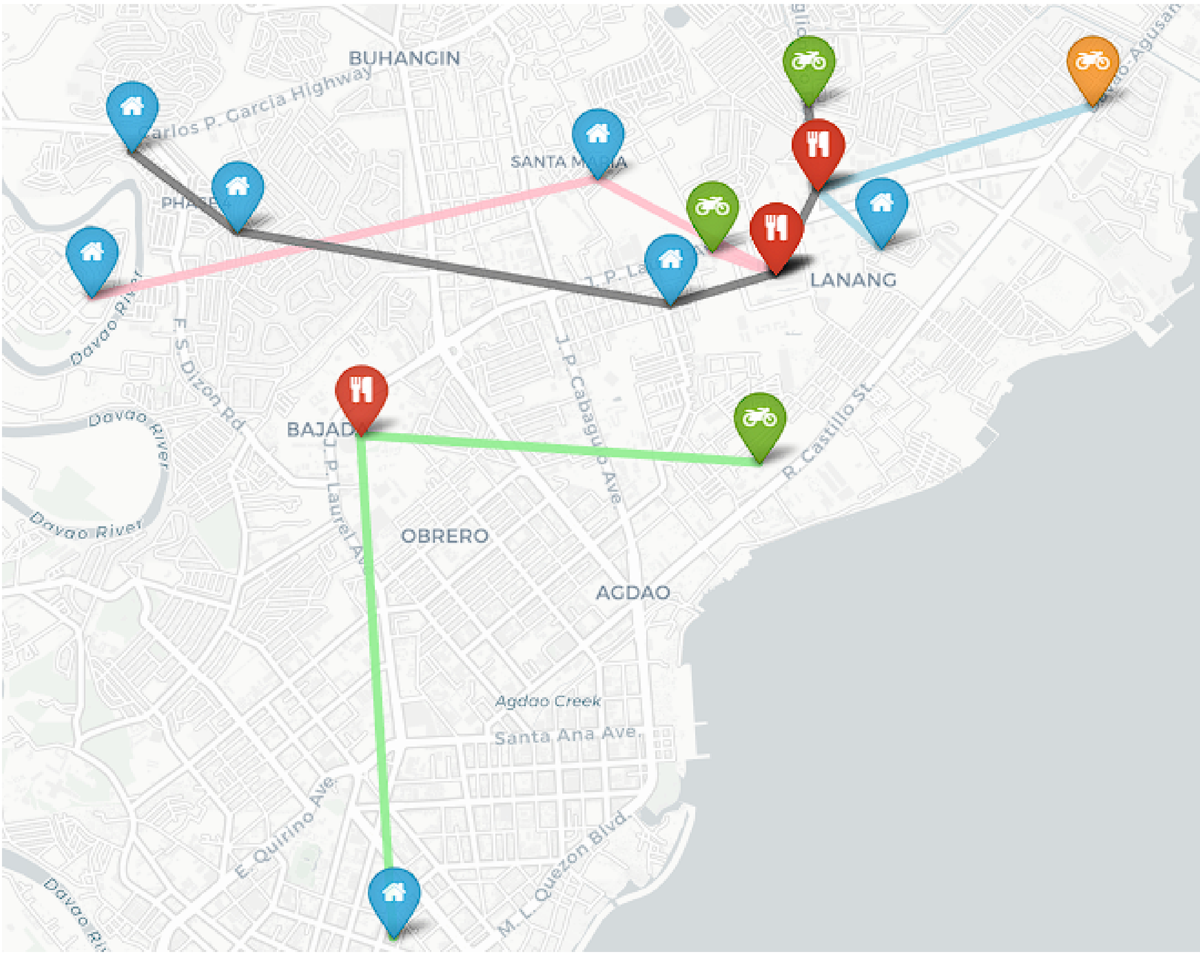

Figure A.10: Instance 2 Bundle 3

rd9 (Pink): 0-13-20-113-120, rd21 (Light Blue): 0-4-104, rd22 (Light Green): 0-15-115, rd10 (Gray): 0-14-16-18-114-118-116

Table 17: Route Results for Base PDPTW Instance 2

Cluster	Rider	Route	Load	Cost	Distance	Run Time
1	rd23	0-1-101-12-112	0-10-4-0	62.93	2561.08	98.6459
	rd5	0-2-102-22-11-111-122	0-4-4-9-4-0	99.9	7843.22	
	rd6	0-19-3-103-119	0-2-4-2-0	77.11	4587.2	
2	rd24	0-26-23-123-126	0-2-4-2-0	70.91	3701.52	0.3698
	rd26	0-24-25-125-124	0-3-4-3-0	72.63	3947.85	
3	rd11	0-20-15-120-115	0-3-6-3-0	100.78	7968.97	164.5794
	rd10	0-4-14-104-114	0-4-6-2-0	59.07	2010.58	
	rd9	0-13-16-18-113-118-116	0-4-5-8-4-1-0	72.98	3997.14	
4	rd32	0-21-9-109-121	0-5-8-5-0	76.55	4507.72	6.3794
	rd17	0-8-108-10-17-27-117-110-127	0-5-4-6-9-7-3-0	90.6	6513.91	
5	rd14	0-5-105-6-106-7-107	0-4-2-4-0	84.38	5625.12	0.1257
Total				867.84	53264.31	

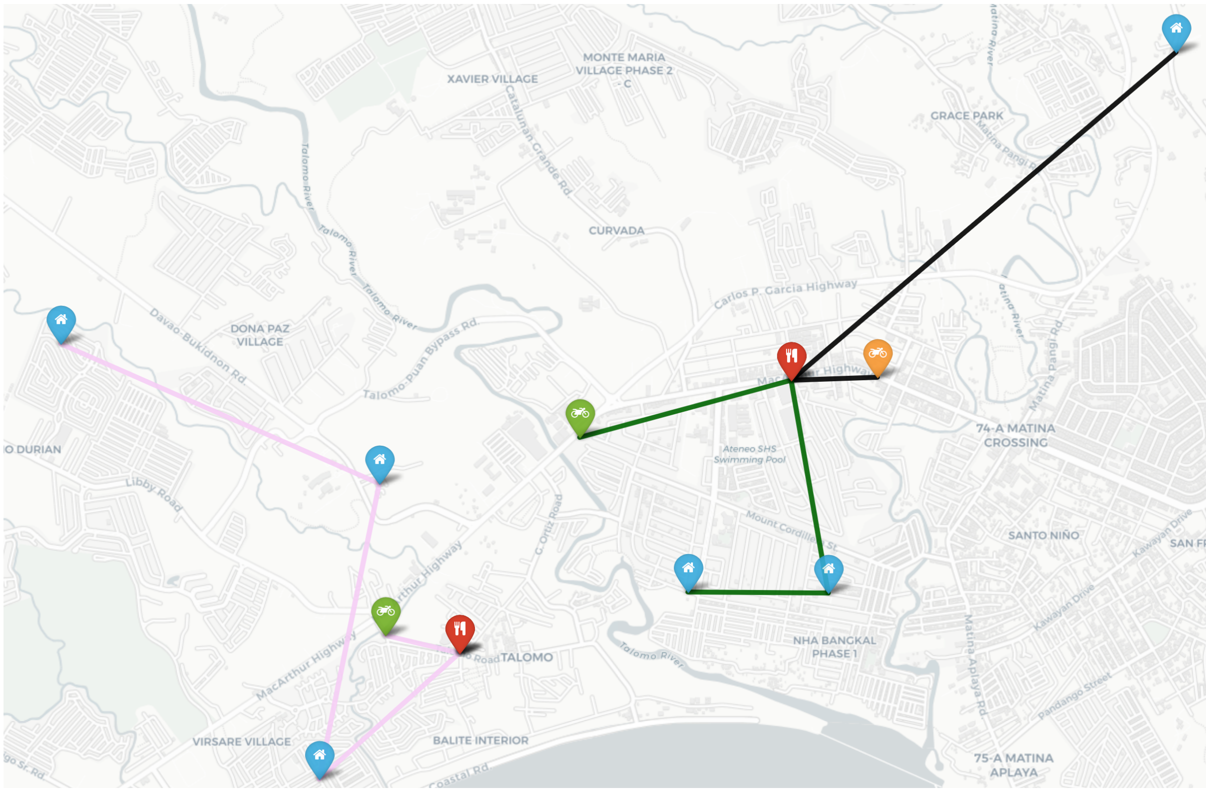

Figure A.11: Instance 2 Bundle 4

rd19 (Light Pink): 0-27-10-17-117-110-127, rd32 (Black): 0-21-121, rd18 (Dark Green): 0-8-9-109-108

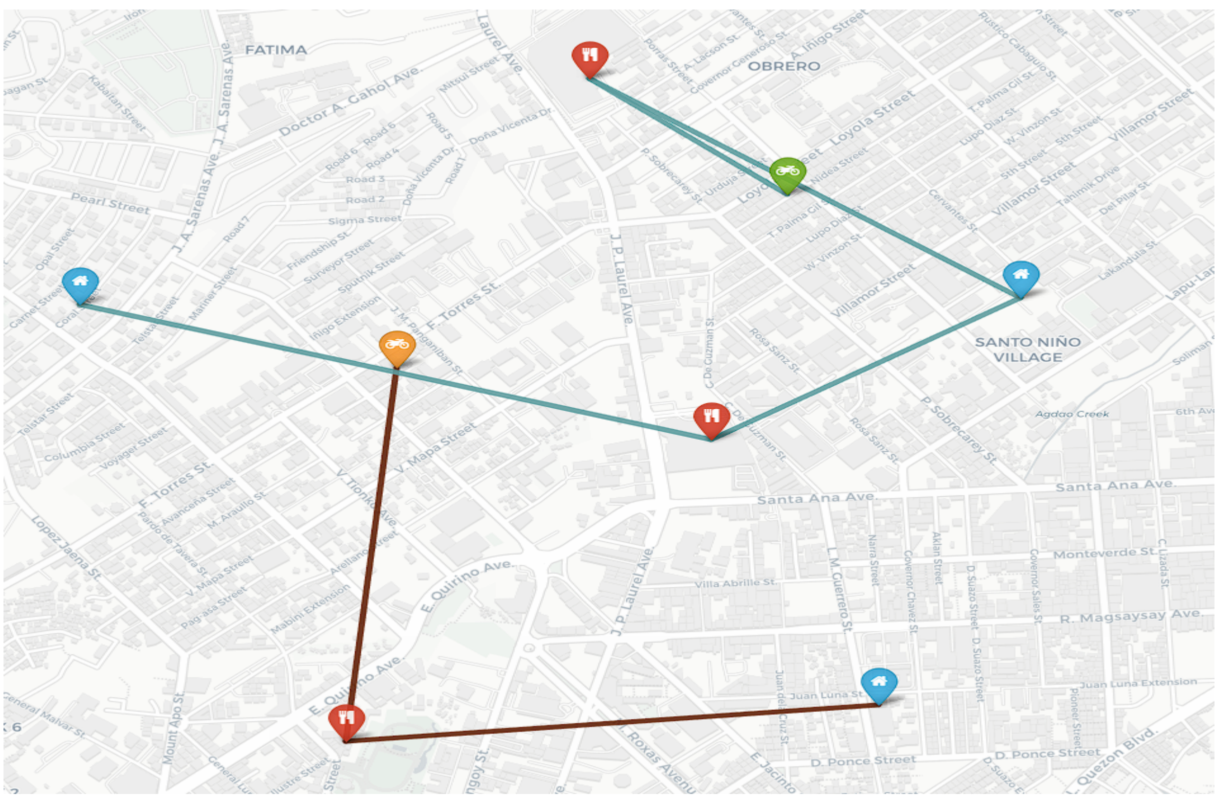

Figure A.12: Instance 2 Bundle 5

rd16 (Cadet Blue): 0-7-107-6-106, rd13 (Brown): 0-5-105

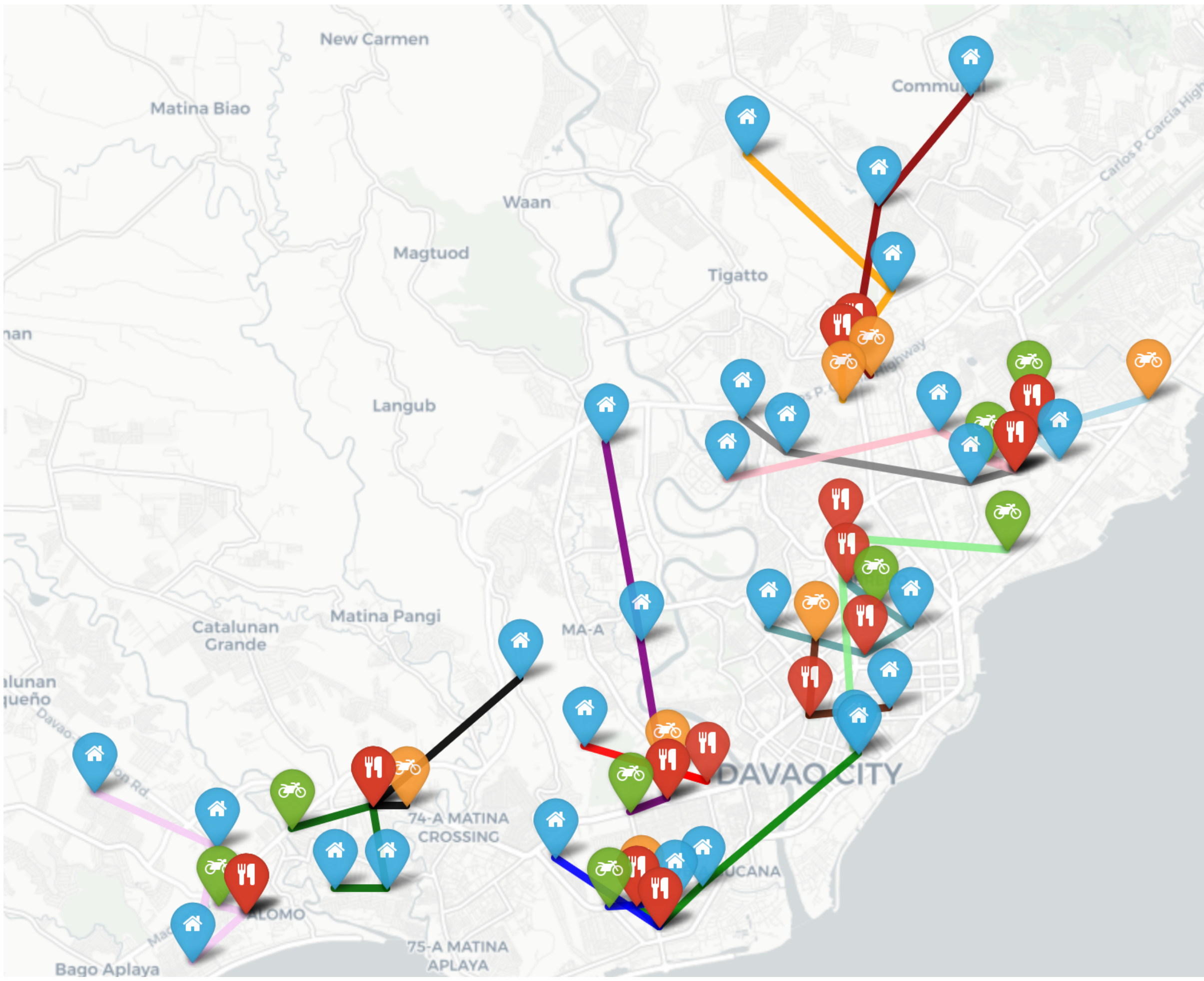

Figure A.13: Instance 2 Routes

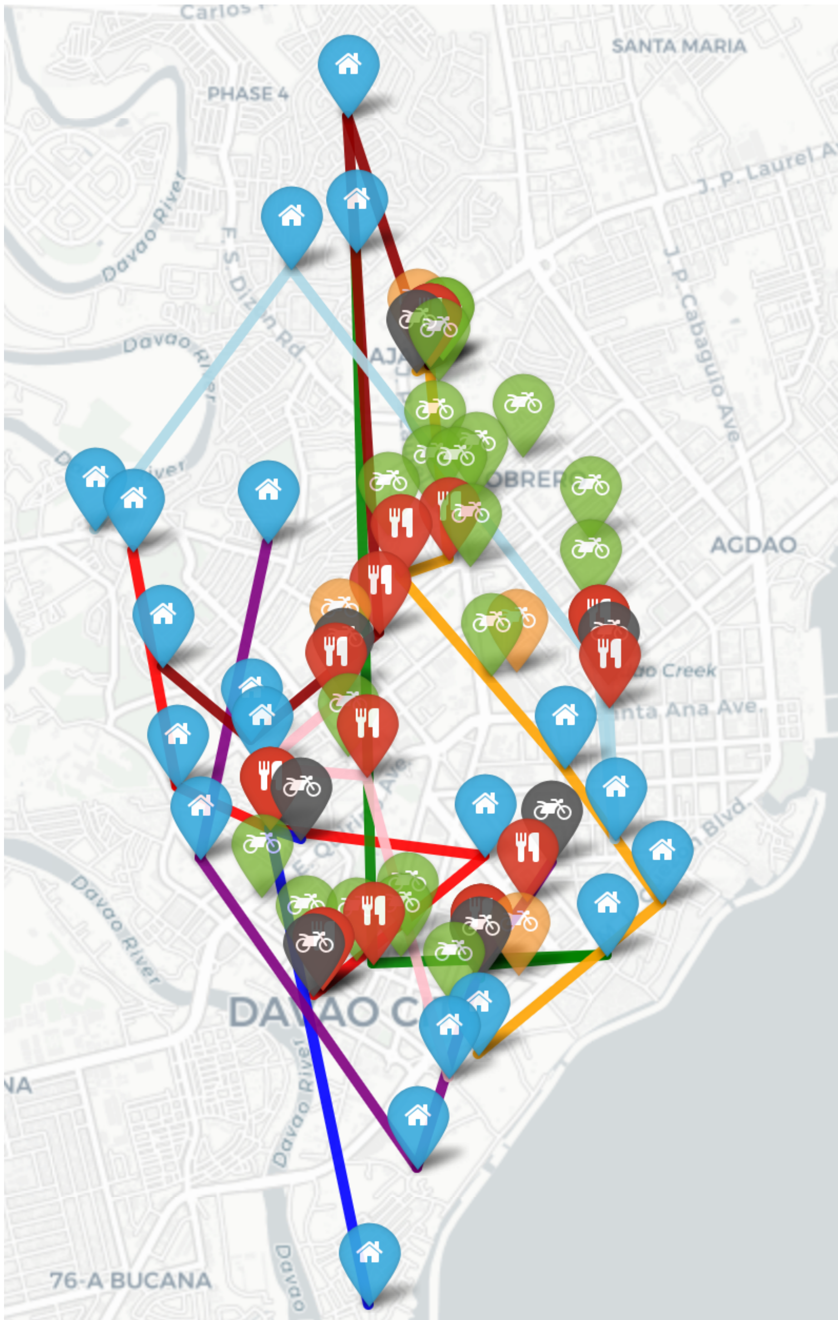

Figure A.14: PDPTW Model of Instance 1

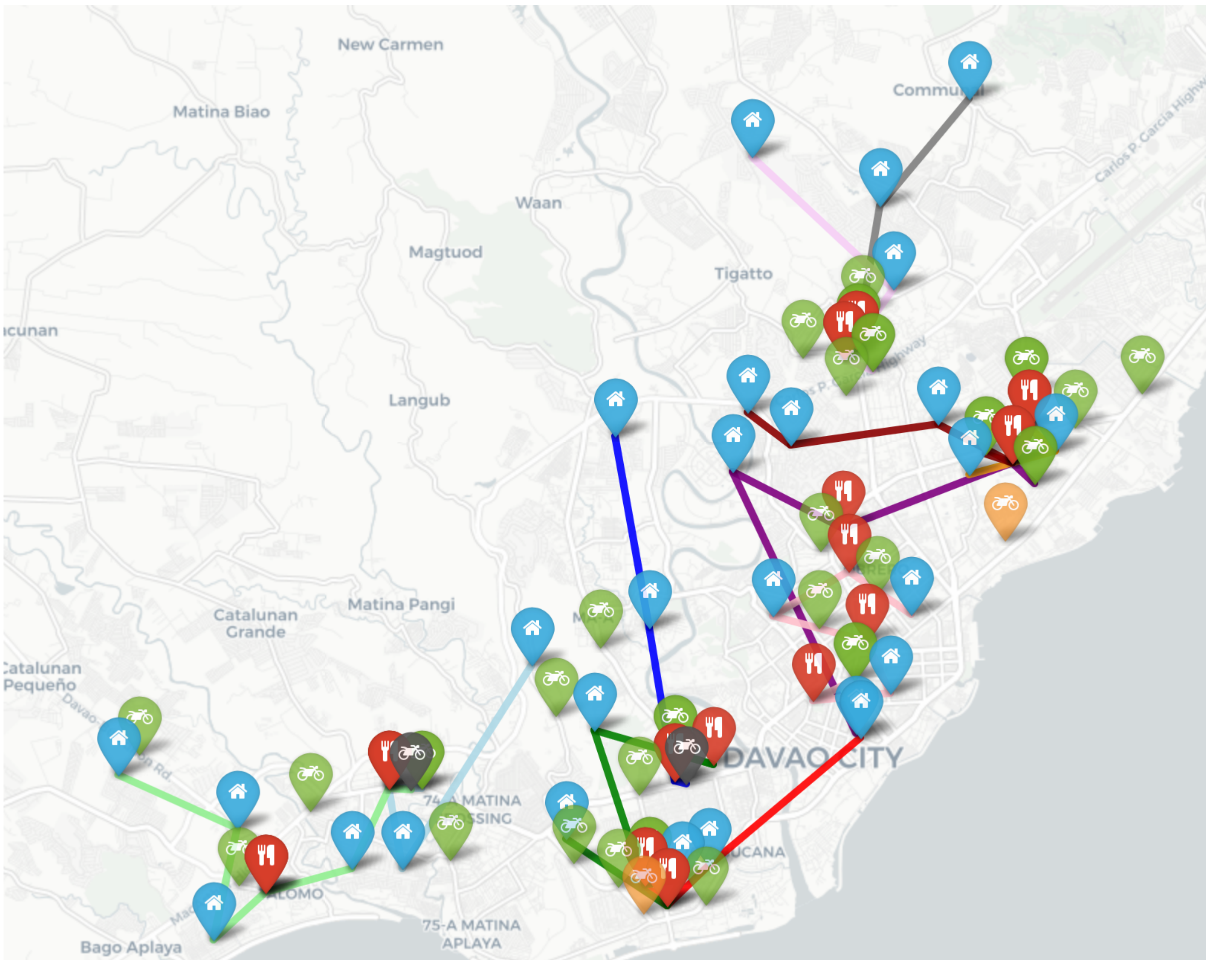

Figure A.15: PDPTW Model of Instance 2

Appendix B

Code Snippets

B.1

This chapter presents code snippets for the methodology.

B.1.1 Snippet: Constrained Clustering of Orders & Riders

B.1.2 Snippet: CPLEX Model

```
# Constrained Clustering
clusters, centers = cop_kmeans(
                      dataset=locations, k=elbow_val, ml=must_link)
# Convert centers to lat and long
centers_LL = list()
for c in range(0,len(centers)):
  (lat,longi)= utm.to_latlon(centers[c][0],centers[c][1], 51, 'N')
  centers_LL += [[lat,longi]]
```

```
#remove outlier riders
from numpy.ma.core import sort

riders_copy = riders
dist = []
for r in riders_copy.index:
  lat = riders.loc[r].latitude
  longi = riders.loc[r].longitude
  dist = []
  for c in range(0,len(centers)):
    dist += [math.ceil(GD((lat,longi),
            (centers_LL[c][0],centers_LL[c][1])).km)]
  dist = sort(dist)
  mini=dist[0]
  if mini > 3:
    riders_copy = riders.drop(r)
# Cluster riders
riders = riders_copy
kmeans_rider = KMeans(
    n_clusters=elbow_val, init=centers, max_iter=1
    ).fit(riders)
```

```
# Model
from docplex.mp.model import Model
mdl = Model('SOMV-PDPTW')

## Decision Var
x = mdl.binary_var_cube(R, N, N, name="%s_from_%s_to_%s")
y = {r: mdl.binary_var(name="assign_%s" % (r)) for r in R}
B = mdl.continuous_var_matrix(R, N,lb=0, name='arrivetime_%s_at_%s')
Q = mdl.continuous_var_matrix(R, N, lb=0, ub=max_cap, name = 'load_of_%s_at_%s')
```

```
### Fair Const
if (fairmodel):
    for r in R:
        if all_riders.loc[r].idle <= beta_2:
            mdl.add_constraint(y[r]==0)

    for r in R:
        if all_riders.loc[r].idle >= beta_1:
            mdl.add_constraint(y[r]==1)

    mdl.add_constraints_(
        (mdl.sum(distance[r,i,j] * x[r,i,j] for i in N for j in N if j!=0) <= max_dist)
        for r in R
    )

## Obj
obj = mdl.sum(fcost_travel*(distance[r,i,j])*(x[r,i,j]) for i in N for j in N if j!=0 for r in R) \
        + fcost_rider * mdl.sum(y[r] for r in R)
mdl.set_objective('min', obj)
```

Appendix C

Interview Data Summary

C.1

This chapter presents the summary of relevant points extracted from the in-depth interviews with the participants, the riders. The fairness notions applied in the model were based on these interview results.

C.1.1 Interview Summary of Grab and FoodPanda Riders

A FAIR PDPTW MODEL FOR DAVAO CITY MEAL DELIVERY SERVICE
DATA COLLECTION INITIAL STEP
INTERVIEW DATA SUMMARY

GRAB	FOODPANDA
1. Order assignment is done randomly. 2. There is no prioritization of order assignments. 3. Riders don't do multiple pick-ups to multiple delivery locations in one route. 4. Grab is very customer biased. 5. Deliveries can reach up to 3-4 bookings in 1 hour. 6. Deliveries can reach up to 24 deliveries on busy days. On non-busy days, there are 10-13 deliveries usually. 7. They can receive orders from within 3 kilometers.	1. They have their pick-up and delivery zones. 2. Order assignment is random. 3. There is no prioritization in the assignment of orders to riders. 4. Thirty minutes lead time is long already. 5. Prefers for the system to prioritize those who waited longer in the assignment of orders. 6. They would want to try multiple bookings as long as all the orders are in one route. 7. Concerned regarding equal profit and order distribution among riders.

Glossary

Anticipatory Commitment Assignment (ACA). a commitment strategy that postpones the task assignment for selected customers, permitting more flexibility in assignments, to represent stochasticity in the MDRP

Demand. an order's capacity requirement in the rider's food bag

Deterministic. a VRP scenario where all available information is assumed to be known with absolute certainty

Dynamic. a VRP scenario where it is assumed that new meal orders may be revealed as time progresses, and the optimizer should try to create new solutions that cater to these requests

Fair MDRP Model. an MDRP model that considers the fairness notions discussed in Chapter 3.2.2

Fairness Notions. fairness from the customers' perspective could mean consideration of geographical fairness in services, while equal distribution of profit for drivers and courier companies

Meal Delivery Routing Problem (MDRP). a problem in which an online restaurant aggregator receives orders from restaurants and matches couriers that perform the pick-up and drop-off of these requests

Node. a location point that can be a pick-up node or delivery node

Pick-up and Delivery Problem (PDP). represents an important family of routing problems in which goods or passengers have to be transported from different origins to different destinations

Run Time. the wall time it takes for the solver to produce a solution from the model

Scalability. the measure of the system's ability to handle problems with more nodes

Stochastic. a VRP scenario where some information, such as travel times, may be uncertain during optimization

Traveling Salesman Problem (TSP). the problem of finding the shortest yet most efficient route for a person to take given a list of specific destinations

Vehicle Routing Problem (VRP). a combinatorial optimization that involves finding an optimal design of routes traveled by a fleet of vehicles to serve a set of customers

Bibliography

[1] Andersson, T. A comparative study on a dynamic pickup and delivery problem: Improving routing and order assignment in same-day courier operations, 2021.

[2] Anh, P. T., Cuong, C. T., and Phuc, P. N. K. The vehicle routing problem with time windows: A case study of fresh food distribution center. In *2019 11th International Conference on Knowledge and Systems Engineering (KSE)* (2019), IEEE, pp. 1–5.

[3] Archetti, C., Feillet, D., Gendreau, M., and Speranza, M. G. Complexity of the vrp and sdvrp. *Transportation Research Part C: Emerging Technologies 19*, 5 (2011), 741–750.

[4] Babaki, B. Cop k-means constrained k-means, 2018.

[5] Baños, R., Ortega, J., Gil, C., Márquez, A. L., and De Toro, F. A hybrid meta-heuristic for multi-objective vehicle routing problems with time windows. *Computers & industrial engineering 65*, 2 (2013), 286–296.

[6] Bansal, S. Issues in solving vehicle routing problem with time window and its variants using meta heuristics - a survey.

[7] Bowerman, R., Hall, B., and Calamai, P. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. *Transportation Research Part A: Policy and Practice 29*, 2 (1995), 107–123.

[8] Bradley, P. S., Bennett, K. P., and Demiriz, A. Constrained k-means clustering. *Microsoft Research, Redmond 20*, 0 (2000), 0.

[9] Çam, Ö. N., and Sezen, H. K. The formulation of a linear programming model for the vehicle routing problem in order to minimize idle time. *Decision Making: Applications in Management and Engineering 3*, 1 (2020), 22–29.

[10] Chen, X. An improved efficient algorithm for time dependent vehicle routing. *Operations and Supply Chain Management: An International Journal 11*, 2 (2018), 55–65.

[11] Chen, X., Wang, T., Thomas, B. W., and Ulmer, M. W. Same-day delivery with fairness. *arXiv preprint arXiv:2007.09541* (2020).

[12] Dondo, R., and Cerdá, J. A cluster-based optimization approach for the multi-depot heterogeneous fleet vehicle routing problem with time windows. *European journal of operational research 176*, 3 (2007), 1478–1507.

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

[14] Furtado, M. G. S., Munari, P., and Morabito, R. Pickup and delivery problem with time windows: a new compact two-index formulation. *Operations Research Letters 45*, 4 (2017), 334–341.

[15] Ghiani, G., Manni, E., Quaranta, A., and Triki, C. Anticipatory algorithms for same-day courier dispatching. *Transportation Research Part E: Logistics and Transportation Review 45*, 1 (2009), 96–106.

[16] Grandinetti, L., Guerriero, F., Pezzella, F., and Pisacane, O. The multi-objective multi-vehicle pickup and delivery problem with time windows. *Procedia-Social and Behavioral Sciences 111* (2014), 203–212.

[17] Gribel, D., and Vidal, T. Hg-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering. *Pattern Recognition 88* (2019), 569–583.

[18] Gumba, R. A. Delivery services see increase in booking, Mar 2020.

[19] Homsi, G., Martinelli, R., Vidal, T., and Fagerholt, K. Industrial and tramp ship routing problems: Closing the gap for real-scale instances. *European Journal of Operational Research 283*, 3 (2020), 972–990.

[20] Hsieh, L.-F., and Huang, Y.-C. New batch construction heuristics to optimise the performance of order picking systems. *International Journal of Production Economics 131*, 2 (2011), 618–630.

[21] Ishizaka, A., Lokman, B., and Tasiou, M. A stochastic multi-criteria divisive hierarchical clustering algorithm. *Omega 103* (2021), 102370.

[22] Jozefowiez, N., Semet, F., and Talbi, E.-G. An evolutionary algorithm for the vehicle routing problem with route balancing. *European Journal of Operational Research 195*, 3 (2009), 761–769.

[23] Kaja, S. C. A new approach for solving the disruption in vehicle routing problem during the delivery: A comparative analysis of vrp meta-heuristics, 2020.

[24] Kritikos, M. N., and Ioannou, G. The balanced cargo vehicle routing problem with time windows. *International Journal of Production Economics 123*, 1 (2010), 42–51.

[25] Kumar, S. N., and Panneerselvam, R. A survey on the vehicle routing problem and its variants.

[26] Kumar, V. S., Thansekhar, M., Saravanan, R., and Amali, S. M. J. Solving multi-objective vehicle routing problem with time windows by faga. *Procedia Engineering 97* (2014), 2176–2185.

[27] LAU, H. C., AND LIANG, Z. Pickup and delivery with time windows: Algorithms and test case generation. *International Journal on Artificial Intelligence Tools 11*, 03 (2002), 455–472.

[28] LEGASPI, J. This service tops philippines' food delivery business, Apr 2021.

[29] LESMANA, N. S., ZHANG, X., AND BEI, X. Balancing efficiency and fairness in on-demand ridesourcing. *Advances in Neural Information Processing Systems 32* (2019).

[30] LI, L. Y., AND FU, Z. The school bus routing problem: a case study. *Journal of the Operational Research Society 53*, 5 (2002), 552–558.

[31] LIAO, W., ZHANG, L., AND WEI, Z. Multi-objective green meal delivery routing problem based on a two-stage solution strategy. *Journal of Cleaner Production 258* (2020), 120627.

[32] LTD, T. B. R. P. Online food delivery services market growth to accelerate at rate of 11 percent as per the business research company's online food delivery services global market report 2021, Dec 2021.

[33] MENDES, R. S., LUSH, V., WANNER, E. F., MARTINS, F. V., SARUBBI, J. F., AND DEB, K. Online clustering reduction based on parametric and non-parametric correlation for a many-objective vehicle routing problem with demand responsive transport. *Expert Systems with Applications 170* (2021), 114467.

[34] MESTRIA, M. New hybrid heuristic algorithm for the clustered traveling salesman problem. *Computers & Industrial Engineering 116* (2018), 1–12.

[35] ORQUIZA, J. R. Davao food industry: A year to shine, Jan 2021.

[36] ÖZDAMAR, L., AND DEMIR, O. A hierarchical clustering and routing procedure for large scale disaster relief logistics planning. *Transportation Research Part E: Logistics and Transportation Review 48*, 3 (2012), 591–602.

[37] REY, D., ALMI'ANI, K., AND NAIR, D. J. Exact and heuristic algorithms for finding envy-free allocations in food rescue pickup and delivery logistics. *Transportation Research Part E: Logistics and Transportation Review 112* (2018), 19–46.

[38] REYES, D., ERERA, A., SAVELSBERGH, M., SAHASRABUDHE, S., AND O'NEIL, R. The meal delivery routing problem. *Optimization Online 6571* (2018).

[39] ROPKE, S., AND CORDEAU, J.-F. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science 43*, 3 (2009), 267–286.

[40] SAVELSBERGH, M. W., AND SOL, M. The general pickup and delivery problem. *Transportation science 29*, 1 (1995), 17–29.

[41] SITEK, P., AND WIKAREK, J. Capacitated vehicle routing problem with pick-up and alternative delivery (cvrppad): model and implementation using hybrid approach. *Annals of Operations Research 273*, 1 (2019), 257–277.

[42] Soeffker, N., Ulmer, M. W., and Mattfeld, D. On fairness aspects of customer acceptance mechanisms in dynamic vehicle routing. *Proceedings of logistik-management 2017* (2017), 17–24.

[43] Teng, R., Hong-bo, X., Kang-ning, J., Tian-yu, L., Ling, W., and Li-ning, X. Optimisation of takeaway delivery routes considering the mutual satisfactions of merchants and customers. *Computers & Industrial Engineering 162* (2021), 107728.

[44] Tumpa, T. J., Ali, S. M., Rahman, M. H., Paul, S. K., Chowdhury, P., and Khan, S. A. R. Barriers to green supply chain management: An emerging economy context. *Journal of Cleaner Production 236* (2019), 117617.

[45] Ulmer, M. W., Thomas, B. W., Campbell, A. M., and Woyak, N. The restaurant meal delivery problem: Dynamic pickup and delivery with deadlines and random ready times. *Transportation Science 55*, 1 (2021), 75–100.

[46] Van Breedam, A. Vehicle routing: Bridging the gap between theory and practice. *JORBEL-Belgian Journal of Operations Research, Statistics, and Computer Science 35*, 1 (1995), 63–80.

[47] Van Lon, R. R., Ferrante, E., Turgut, A. E., Wenseleers, T., Berghe, G. V., and Holvoet, T. Measures of dynamism and urgency in logistics. *European Journal of Operational Research 253*, 3 (2016), 614–624.

[48] Vidal, T., Crainic, T. G., Gendreau, M., and Prins, C. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time-windows. *Computers & operations research 40*, 1 (2013), 475–489.

[49] Wang, W., and Jiang, L. Two-stage solution for meal delivery routing optimization on time-sensitive customer satisfaction. *Journal of Advanced Transportation 2022* (2022).

[50] Wang, Z. Delivering meals for multiple suppliers: Exclusive or sharing logistics service. *Transportation Research Part E: Logistics and Transportation Review 118* (2018), 496–512.

[51] Wang, Z., and Lin, W.-H. Incorporating travel time uncertainty into the design of service regions for delivery/pickup problems with time windows. *Expert Systems with Applications 72* (2017), 207–220.

[52] Wen, M., Cordeau, J.-F., Laporte, G., and Larsen, J. The dynamic multi-period vehicle routing problem. *Computers & Operations Research 37*, 9 (2010), 1615–1623.

[53] Xu, Q., Zhang, Q., Liu, J., and Luo, B. Efficient synthetical clustering validity indexes for hierarchical clustering. *Expert Systems with Applications 151* (2020), 113367.

[54] Xue, G., Wang, Z., and Wang, G. Optimization of rider scheduling for a food delivery service in o2o business. *Journal of Advanced Transportation 2021* (2021).

[55] Yildiz, B., and Savelsbergh, M. Provably high-quality solutions for the meal delivery routing problem. *Transportation Science 53*, 5 (2019), 1372–1388.

[56] Zhang, Z., Qin, H., and Li, Y. Multi-objective optimization for the vehicle routing problem with outsourcing and profit balancing. *IEEE Transactions on Intelligent Transportation Systems 21*, 5 (2019), 1987–2001.

[57] Zhou, L., Zhen, L., Baldacci, R., Boschetti, M., Dai, Y., and Lim, A. A heuristic algorithm for solving a large-scale real-world territory design problem. *Omega 103* (2021), 102442.

[58] Zhu, S., Hu, X., Huang, K., and Yuan, Y. Optimization of product category allocation in multiple warehouses to minimize splitting of online supermarket customer orders. *European journal of operational research 290*, 2 (2021), 556–571.