ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Bachelor Thesis BSc^2 Economics & Econometrics

Where do we go now - Courier relocation in the Meal Delivery Routing Problem

Teona Banu
504450

Supervisor: Bart van Rossum B. T. C
Second assessor: Dr. Riley Badenbroek

July 2022

Contents

1 Introduction

All around the World, meal delivery platforms are becoming increasingly popular, with the market for meal delivery services having tripled since 2017 (Ahuja et al., 2022). It all began with home deliveries offered by restaurants selling party or comfort food, such as pizza (Statista, 2021) and Chinese food (Grubtech, 2021), as well as catering services which have long been used to provide lunches to workers and which trace their history to Indian dabbawalas (Thomke, 2014). By 2019, the scope and popularity of meal deliveries had already been increasing, as people had been, especially in urban centres, cooking less and opting for ready-made foods more. Some of these early adopters of meal delivery platforms had been driven by convenience, while others had limited access to high-quality, affordable ingredients or meals (Ahuja et al., 2022). However, the popularity of these services skyrocketed during the COVID-19 pandemic, when restaurants were forced to close their seating areas, and people were forced to socially distance, isolate or quarantine at home (Statista, 2021). Thus, restaurants saw an opportunity to remain in business by keeping their kitchens open for deliveries while customers needed food and groceries delivered to their doorstep when they could not leave their homes (Ahuja et al., 2022). However, it was still infeasible for many restaurants to hire their own dedicated delivery staff.

This was the perfect climate for online restaurant aggregators to emerge. These platforms took on the task of matching customer orders from restaurants to gig worker couriers, managing the distribution of human resources, and streamlining the delivery process. And so, the fast and accessible meal delivery services offered by these platforms proved to be indispensable to people suffering from short-term sickness or long-term disabilities, which make it difficult to leave their homes. Moreover, people at risk of complications from catching a cold or the flu are to this day forced to isolate themselves at home as the risk of infection with COVID-19 is at an all-time high. Finally, most people are unlikely to part with the convenience of having restaurant food delivered to their doorstep, which has become an entrenched part of our day-to-day lives. That is why even as restaurants worldwide are reopening their indoor seating areas and people are no longer required to distance themselves socially, the popularity of food ordering and delivery platforms is only set to increase to a projected market volume of $474.30 billion and 2.691 million users by 2026 (Statista, 2021). Some restaurants are even choosing not to reopen seating areas at all. "Dark kitchens" are restaurants operating solely through delivery platforms. These establishments are not only becoming more popular but are often prominently featured by meal delivery platforms (Ahuja et al., 2022), a testament to the ubiquity of meal delivery services and the massive growth of meal delivery platforms which have already changed the way we think about restaurant-cooked food.

This massive growth translates to ever-increasing demand, which requires an increased number of available couriers. However, since couriers should be guaranteed a minimum hourly wage, increasing the number of couriers can have diminishing or even negative returns. This is just one of the ways in which, despite this massive growth, meal delivery platforms often see low or even negative profits. More generally, the main challenges faced by meal delivery platforms deal with reconciling the clashing interests of the various stakeholders: customers, restaurants, couriers, and investors. Customers expect fresh food and short delivery times. On top of satisfying customer expectations, restaurants expect to have high service levels, all of which increase their popularity and result in higher revenues. Couriers are given the freedom of an independent contractor, which results in arbitrary log-on and log-off times and the ability to accept or reject any order. At the same time, they expect adequate compensation for their labour. Finally, investors expect high profitability of the food delivery platforms and a high market share, which hinge on satisfying customer expectations and maximising service levels while minimising costs.

The rapidly increasing demand represents great potential, but with so many overlapping factors comes the requirement of simulation and optimisation technologies that are more efficient than ever. To solve this, The Meal Delivery Routing Problem (MDRP) was formally introduced by Reyes et al. (2018). They provide an optimisation-based algorithm which takes orders as they come, generates routes, and then assigns these routes to couriers. Their approach achieved high-quality solutions using data sets that reflected the 2018 market conditions, but nowadays, the lack of sufficiently expressive and sufficiently accurate optimisation technologies is keeping the profits of meal delivery platforms from reflecting the size of the market (Ahuja et al., 2022), all while these platforms are failing to provide adequate wages for couriers. This has lead to courier strikes all over the world (Cornwell, 2022) (Aghayev, 2022) (Shoshiashvili, 2021) (France-Presse, 2022) (Stone, 2022).

The recent courier strikes point to one of the problems that optimisation algorithms employed by meal delivery platforms are yet to address and are far from solving. Couriers are the central link of the meal delivery operation, making it crucial to develop proper modelling algorithms to predict and optimise courier behaviour. This technology can be used to generate pay schemes that provide fair compensation for couriers and incentivise behaviours that lead to higher profits. Understanding the courier-related elements of the MDRP will lead to more efficient outcomes, which can improve the economic situation of both couriers and the meal delivery platform. Throughout their paper, Reyes et al. (2018) make a series of simplifying assumptions about courier behaviour. Consequently, they propose further insight into how courier autonomy shapes the meal delivery process as a worthwhile research direction. Thus, this paper sets out to expand upon the MDRP model introduced by Reyes et al. (2018) in two directions. First, I consider autonomous relocation while idle, allowing for the phenomenon of "courier drainage" to manifest its effects. Second, I implement a centralised relocation system meant to replace autonomous relocation.

The results confirm that autonomous relocation results in longer delivery times, with the effect being more pronounced when couriers relocate to neighbourhoods they perceive as more profitable and less pronounced when couriers relocate to the centre of their nearest neighbourhood. Centralised relocation proves to be dependent on parameter tuning but manages to achieve better results than the base model when properly tuned.

The further development of MDRP-related technologies, to which this paper contributes, fulfils the practical application of ensuring a better experience for customers, better work conditions for couriers, and increased profits for meal delivery platforms and restaurants. Furthermore, these algorithms and models can provide a basis for other similar problems, contributing to the existing scientific literature on the MDRP and, more generally, on PDP and VRP.

The rest of the paper is structured as follows: Section 2 presents an overview of relevant existing literature; The problem description is given in section 3, including the original MDRP and the two additional settings that will be investigated; Section 4 presents the methodology; The instances used and the results of the various experiments performed are presented in Section 5; Finally, Section 6 contains a discussion of the results and concluding remarks.

2 Literature Review

Prior to 2020, literature on the MDRP is sparse and tends to consider general settings. Subsection 2.1 presents some of the relevant literature that laid the groundwork for future papers. More recently, papers have started to consider alternative settings or prioritise alternative performance metrics. These will be detailed in section 2.2. Van Dun et al. (2020) adopt a customer-centric approach. Tao et al. (2020) consider various types of couriers and their respective features. Chen et al. (2022) use both vehicles and drones as couriers but also look at the possibility of intelligent order rejection. Section 2.3 presents papers that delve deeper into intelligent order rejection and repositioning. In addition to Chen et al. (2022), Ulmer and Thomas (2020) and Jahanshahi et al. (2021) also consider intelligent order rejection or repositioning in various settings, with the former looking at next-day goods delivery, and the latter looking at meal deliveries. Finally, Zou et al. (2021), and Quintero-Rojas (2020) have addressed the ways in which courier behaviour shapes the MDRP by means of simulators with more realistic courier properties and features which will be summarised in section 2.4.

2.1 The Roots of the MDRP

After formally introducing the MDRP, Reyes et al. (2018) provide an optimisation-based algorithm which uses a myopic rolling horizon approach by first solving the vehicle routing problem and then assigning the routes to couriers. To test the performance of their model, they use sets of artificially created instances with realistic properties. They use a coordinate system with a reference point and consider euclidean distances between couriers, restaurants and customers. Their results show that their approach manages to achieve high-quality solutions despite not capturing future order information. Furthermore, they highlight the critical importance of capacity scheduling and the minor impact of meal preparation time.

Yildiz and Savelsbergh (2019) consider a similar problem, the main difference being that they use a clairvoyant approach by assuming perfect information about the order-arrival. Using a combination of column and row generation, their main objective is minimising cost and click-to-door times. They perform extensive numerical experiments using real-world data which show the importance of courier schedule planning, order bundling and demand management. Furthermore, they show that a hierarchical optimisation approach efficiently guarantees Pareto optimal solutions and that minimum pay guarantees for couriers are feasible within their model description.

Steever et al. (2019) define the Virtual Food Court Delivery Problem (VFCDP), similar to the MDRP but allowing orders to contain items from multiple restaurants at a time. They use a mixed-integer linear programming formulation in a simulation environment and evaluate their model based on speed, freshness, and courier travel distance. An auction-based heuristic is used to incorporate future demand in the decision process. Their results, which consider artificially created instances based on probability distributions, show that look-ahead policies outperform purely myopic solutions.

This paper will provide an extension to the work of Reyes et al. (2018) by introducing additional features aimed at improving the MDRP's expressivity with regard to courier behaviour and dynamics. Moreover, I will be introducing a look-ahead element insofar as autonomous courier decisions reflect their experience and expectations of future order offers.

2.2 Alternative settings

Van Dun et al. (2020) theorise that speed and freshness, described as food temperature, are the most important factors in the meal delivery process. They dictate customer satisfaction, influencing customer loyalty and resulting in recurring orders and a higher market share control. Their model is customisable, with interchangeable short and long-term options. Using real-world data from a German platform, they show that a significant increase in long-term, customer-centric factors is achievable at the cost of only a slight reduction in short-term performance.

Tao et al. (2020) describe Online to Offline (O2O) services provided by platforms which allow users to place orders online and receive products/services offline. Their model considers three types of couriers with different characteristics and explores the challenges for the management of order assignment, routing, and courier incentive schemes posed by this variation in courier types. In this endeavour, they look at traditional in-house drivers as well as part-time and full-time crowd-sourced drivers. They find that the maximum allowed number of orders per trip and the detour proportion are critical for obtaining efficient results.

Chen et al. (2022) consider an assignment problem in which orders are assigned to either drones, which are fast but require charging and maintenance, or vehicles, which have more capacity but are prone to get stuck in traffic. They use Deep Q-learning and Reinforcement Learning approaches to solve fleet management policies for same-day deliveries of dynamic orders. They include the option of rejecting an order. The results of their policy manage to outperform benchmark policies.

Since Van Dun et al. (2020) showed that customer-centricity and profits are interrelated, this paper will investigate the way courier autonomy influences profits. The MDRP problem and this paper will only be considering crowd-sourced drivers who are free to choose their work hours and duration. Still, the findings from Tao et al. (2020) can provide valuable insights into the critical effects of bundle sizes. Likewise, this paper will diverge from the work of Chen et al. (2022) by only considering human couriers driving vehicles.

2.3 Intelligent Rejection & Repositioning

Ulmer and Thomas (2020) investigate the Capacitated Customer Acceptance Problem with Stochastic Requests (CAPSR). This problem considers next-day goods delivery instead of immediate meal deliveries, and moreover, it assumes a single courier with a very large capacity that services an entire area. The aim is to maximise expected revenue by accepting or rejecting requests. Within their model, accepted requests generate revenue and must be routed; however, they consume driver time and vehicle capacity. Their computational experiments show that Meso-Parametric Value Function Approximation outperforms benchmarks for the CAPSR.

Jahanshahi et al. (2021) study meal delivery services over a day with dynamic customer requests and a given set of couriers. They use a Markov Decision Process (MDP) for the courier assignment task and Deep Reinforcement Learning as the solution approach. They use both artificial and real-world data to compare various policies. They focus, among other goals, on investigating the effect of intelligent order rejection and repositioning of the couriers. They highlight the importance of incorporating geographical locations of restaurants, customers, and depots to improve service quality. Furthermore, they provide insights into the courier assignment process and the optimal number of couriers for different order frequencies.

This paper will strictly consider meal delivery services, as opposed to Ulmer and Thomas (2020)'s goods delivery.

2.4 Behaviour Simulators

Zou et al. (2021) created a simulator that combines the behaviours of customers, merchants, dispatchers and couriers. Their multi-agent model enables the evaluation of alternative order delivery strategies. They compare a Nearer Merchant Assignment, which always chooses the courier closest to the merchant, with a TSP-based Assignment, which appoints the courier whose total cost has the least increment after the additional order. Their results, which incorporate real-life road networks and real-life order data, show that the TSP-based strategy outperforms the Nearer Merchant Assignment strategy in terms of efficiency. Furthermore, they find that a larger load capacity of couriers improves completion rate but does not affect completion time.

Quintero-Rojas (2020) creates a discrete events simulator which is intended to represent the components of a meal delivery operation by representing how actors make decisions or take action. Their use a modular model where the results of different interchangeable policy sets can be tested using real-life instances. The novel courier-related characteristics introduced are the ability for couriers to move freely while idling, the introduction of vehicles that alter the movement mechanics, and the option to reject orders proportionately to distance or courier propensity to accept an order.

In light of the existing literature, this paper will extend the work of Reyes et al. (2018) through two new settings, allowing for courier relocation either based on autonomous decisions or on centralised decisions made by the meal delivery platforms. Under autonomous relocation, "courier drainage" takes place, resulting in longer delivery times, while centralised relocation shows potential to improve on the solutions of the base model.

3 Problem Description

The Meal Delivery Routing Problem (MDRP) is a type of dynamic pick-up and delivery problem. It describes the way a meal delivery platform functions, with couriers working to pick up orders from restaurants and deliver them to customers. The list of restaurants and their coordinates are known, as well as the list of couriers, including their coordinates and work hours. Under the rolling horizon algorithm used by Reyes et al. (2018), items from the list of orders, including placement time and customer coordinates, are revealed throughout the day as they are placed. The default algorithm used to solve the problem first creates bundles of orders from each restaurant and then assigns these bundles to couriers to be delivered to customers in a predetermined order. Travel times are computed using Euclidean distances and assume an invariant courier speed. Service times are considered, both at pick-up and delivery and are assumed to be four minutes long. Under the base model, couriers are assumed to take no autonomous action. Prepositioning is implemented through a commitment strategy which can result in instructions to move to a restaurant and wait there for bundles to be assigned. The default model maximises the throughput of orders, with a penalty for freshness loss. Click-to-door time, the difference between the drop-off time and the placement time is the performance measure used to select parameters such as those relating to the weight of the freshness loss penalty, assignment horizon or optimisation interval. Ready-to-pickup time, the difference between order pick-up time and ready time, as well as cost per order and orders per bundle, are other performance measures that are being computed and reported. The cost represents courier compensation and is computed as 15 per order delivered with a minimum hourly rate of 10.

Autonomous Relocation (AR) represents the couriers deciding to move towards areas perceived as more profitable. In practice, this leads to the phenomenon of "courier drainage", which characterises a state where certain areas become over-saturated with couriers while less profitable areas see an under-supply of couriers and, consequently, customers experience a poorer service. Because of this empirical observation, I expect performance measures to degrade under the AR setting, specifically through higher Click-to-Door and Ready-to-Pickup times. In order to simulate the effects of AR, the restaurants are split into clusters using the K-means clustering technique. This will simulate neighbourhoods or different areas of the city. Moreover, couriers that have finished an assignment will pick a cluster based on proximity and profitability (measured in the number of orders), and they will head to the centre of the cluster. This will simulate couriers heading towards the point that minimises their distance to (nearby) profitable restaurants, which is meant to increase their chances of getting an assignment as soon as possible. For this setting, I am experimenting with different numbers of clusters as well as different weights for profitability and proximity.

Centralised Relocation (CR) represents the meal-delivery platform making the relocation decision for the couriers. This is meant to replace autonomous relocation with couriers being expected to follow the instructions as opposed to freely relocating. Since this setting will represent a centralised decision for all couriers, it is not expected to lead to courier drainage and, on the contrary, is expected to improve performance measurements. A number of couriers proportional to the number of expected orders from each restaurant will be sent to the respective restaurants. In theory, the courier-restaurant pairings are aimed to minimise the total pick-up time in the next period. To that end, the optimiser will minimise the time required to arrive at restaurants while accounting for the average preparation time of each restaurant, with restaurants that see longer preparation times being assigned to couriers that are further away since there is no point in ensuring a quick arrival if no meals are expected to be ready by then, while restaurants with fast preparation times are to be assigned to couriers who are able to arrive as soon as possible after meals are expected to be ready.

4 Methodology

Subsection 1 presents the methodology used to replicate the default MDRP solution algorithm as described by Reyes et al. (2018); this includes the bundling procedure, priority grouping, linear assignment model and commitment strategy. Subsection 2 describes the autonomous relocation setting, including the clustering method and relocation algorithm. Subsection 3 presents the centralised relocation setting and the courier to restaurant relocation matching model.

4.1 Replication

In order to analyse the outcomes of the two relocation settings, these must be compared to the base model presented in the MDRP. I am using the default MDRP settings, with bundling procedure 1 and the first linear assignment model. I include the priority scheme and the two-stage additive commitment strategy. The algorithm solves a matching problem every f minutes, with f being the optimisation interval. Algorithm 3 from Appendix A presents the main elements used to replicate one full day of work, as described in this section. First, a list of available couriers is generated. This represents couriers who are currently working and who are expected to become available within the next Δ_2 minutes. The target bundle size is computed by dividing the number of undelivered orders that will be ready within the next Δ_1 by the number of available couriers. The list of undelivered orders that will be ready within the next Δ_U minutes is then used to generate bundles. This is done using a parallel-insertion procedure by inserting each order in bundles such that the increase in route cost is minimised. The route cost considers the travel time as well as the service delay generated by waiting for the order to be ready. The number of bundles created is up to, whichever is larger, the target bundle size or the number of couriers who are already waiting at the restaurant as a result of prepositioning. These bundles are then classified into three priority groups. If a bundle contains an order whose target drop-off time is impossible to achieve, it is given the highest priority, followed by bundles containing orders that can no longer be picked up at their ready time, and finally, bundles that do not fall into the previous categories. The linear assignment model then matches the bundles to couriers, with higher-order bundles being matched first. Each courier can only be assigned one bundle, and all bundles must be matched to either a real or a 'dummy' courier. The objective function to maximise contains a throughput value resulting by dividing the number of orders in a route by the time required to execute the assignment. There is a penalty for pick-up delays. These assignments are then subjected to the second stage of the commitment strategy. If an order has been ready for more than x minutes or if a bundle can be picked up by the next assignment period, a final commitment is made for the courier assigned to the respective bundle. Otherwise, if the courier can at least reach the restaurant by the next assignment period, a partial commitment is made, and the courier is instructed to move to the restaurant and wait for the assigned bundle, which can potentially be updated to include more orders. If none of these conditions can be met, the assignment is ignored. Once a final commitment is made, the courier moves to the restaurant, picks up the order, and proceeds to deliver each component of the bundle in the order generated by the bundling procedure. The courier ends the assignment period at the location of the last customer on the route. Under the base model, couriers wait at this location until they are given a new assignment.

4.2 Autonomous Relocation

The first setting considered is autonomous relocation (AR). Under this setting, I am assuming that restaurants form neighbourhoods or clusters. Due to previous experience living and working as couriers in the city, the couriers are assumed to be aware of the neighbourhoods, their centre, and the percentage of total orders coming from each neighbourhood. Since compensation for an order is higher than the minimum guaranteed compensation, when idle, couriers are assumed to want to receive a new assignment as soon as possible. To this end, after finishing an assignment, they are assumed to pick a preferred neighbourhood and move towards it. Couriers are assumed to prefer neighbourhoods that have a higher percentage of orders, as they are more likely to be assigned an order the closer they are to as many restaurants that see as many orders as possible. They are also assumed to prefer clusters that are closer to them in order to minimise the cost of relocation for which they are not compensated.

First, for each instance set, the restaurants are grouped into clusters using the K-means clustering technique. This is a partitional clustering approach where each cluster has a centre point, and each point is assigned to the cluster with the closest centre point. The basic algorithm is presented in algorithm 1. The clusters are formed by assigning restaurants to the *closest* centroid. "Closeness" is measured by Euclidean distance in order to be consistent with the MDRP. The number K must be specified at the beginning. I am using the sum of squared errors (SSE), expressed in equation **??**, to evaluate different clusterings and choose the K that minimises the SSE. In principle, the SSE is lower for higher K. However, a good clustering with a smaller K can have a lower SSE than a poor clustering with a higher K. That being said, couriers cannot be expected to have a mathematically precise knowledge of the neighbourhoods in their city. Therefore, the Results section will show how the outcomes change for different choices of K.

Algorithm 1 K-means Clustering

```
1: for k ← 1 to K do
2:     mean_k ← restaurant_{kR/K}
3: repeat
4:     Form K clusters by assigning all points to the closest centroid
5:     Recompute centroids
6: until The centroids don't change
```

Algorithm 2 shows how couriers choose clusters and move towards them. The travel time from the courier's initial position to the centre of a cluster is $t(courier, cluster)$. The point along the way from courier d to the centre of a cluster k that can be reached by assignment time *time+f* is $p_{d,k}$. It is computed using equation **??** and represented graphically in figure **??**. At each assignment period, as per the replication section, couriers might receive instructions to move to restaurants or pick up and deliver orders (or bundles). The default MDRP model assumes that at the time of these assignments, the time it will take a courier to complete the assignment is known with perfect precision. Thus, at the time of an assignment, it is known which couriers will finish an assignment before the next assignment period. Under the AR setting, these couriers are then assumed to move towards the *best* cluster. The criteria used for choosing the best cluster are proximity (closeness) and profitability (number of orders). In this regard, I experiment with varying the weight α for profitability vs proximity. Thus, couriers move towards the centre of their chosen cluster until they either arrive in the centre, at which point they stop as there is no spot where they're more likely to have orders assigned, or until they are assigned an order, at which point they stop in the spot they are when they receive the new assignment, accept the order, and head to the respective restaurant. Note that the *bestCluster* decision is taken independently by each courier. This makes it likely, especially if couriers tend to value profitability more than proximity, for couriers to converge in profitable clusters and over-saturate them.

Algorithm 2 Autonomous Relocation

for $d \leftarrow 1$ **to** D **do**
    if **then**$e_d \leq time + f$
        $bestCluster \leftarrow \arg\max_{k \in \overline{1,K}} (\frac{\alpha}{\%orders_k} + (1-\alpha) * t(courier_d, centre_k))$
        Compute point $p_{d,k}$
        $courier_d location \leftarrow p_{d,k}$
        $e_d \leftarrow e_d + t(courier_d, p_{d,r})$

4.3 Centralised Relocation

The second setting considered is centralised relocation (CR). Under this setting, the meal delivery platform is assumed to know the average number of orders per day that each restaurant sees. Moreover, the platform knows the average preparation time for each restaurant. The goal of the centralised relocation is to minimise the pick-up time of the orders assigned in the next assignment period. The pick-up time of an order is determined by, whichever is longer, the preparation time left after the order is assigned, and the travel time to the restaurant of the courier matched to the order. To achieve this, the platform matches idle couriers to restaurants and instructs them to start moving toward them. As opposed to the prepositioning through partial commitment feature included in the original MDRP, couriers are *likely, but not guaranteed* to be assigned orders from the restaurant to which they are matched.

To solve the matching problem, the sets D_{CR} and R_{CR} are used, representing the set of couriers that have finished their current assignment before the next assignment period, excluding couriers on partial commitments who are already at restaurants and the set of restaurants considered for courier relocation respectively. The set of restaurants is generated as the set of restaurants expected to have a number of orders that is above a certain threshold Δ_{CR}. For each restaurant r, p_r is the percentage of orders expected by the next period out of the total orders expected by the next period, and e_r is the average preparation time. For each pair (d, r) of courier d and restaurant r, $t(p_{dr}, r)$ is the travel time left for courier d until arrival at restaurant r, assuming that c is instructed to head to restaurant r after completing their previous assignment. Using these sets and parameters, the following assignment model generates assignments x_{dr} which take value 1 if courier d is instructed to head to restaurant r after completing their previous assignment, and value 0 otherwise.

$$
\begin{aligned}
min \quad & \sum_{d \in D_{CR}} \sum_{r \in R} max(e_r - \frac{f}{2}, t(p_{dr}, r) + \frac{service}{2}) x_{dr} && && \text{(1a)} \\
\text{s.t.} \quad & \sum_{r \in R} x_{dr} \leq 1, && \forall d \in D_{CR}, && \text{(1b)} \\
& \sum_{d \in D_{CR}} x_{dr} = p_r |D_{CR}|, && \forall r \in R, && \text{(1c)} \\
& x_{dr} \in \mathbb{B}, && \forall d \in D_{CR}, r \in R && \text{(1d)}
\end{aligned}
$$

The pick-up time of an order is the maximum between the ready time and the arrival time of the courier at the restaurant, plus half the service time. If within an assignment period, order placement time is assumed to be uniformly distributed, the average preparation time left for an order is the average preparation time at that restaurant minus half the assignment period. Thus, the earliest expected pickup time for courier d assigned to restaurant r is $max(e_r - \frac{f}{2}, t(p_{dr}, r) + \frac{service}{2})$. The objective function 1a minimises the total sum of time left from assignment period $time + f$ until the expected pick-up of a potential order from restaurant r by courier d. Constraint 1b ensures that each courier is not instructed to move to more than one restaurant at a time. Constraint 1d forces the number of couriers assigned to each restaurant to be proportional to the number of orders expected at each restaurant. With more couriers being instructed to move towards restaurants that see more orders per day. After each courier is matched to a restaurant, the couriers move towards the restaurant. Just like in the AR setting, the courier stops when they reach the destination or when they receive an assignment, at which point they accept it and move towards the restaurant newly assigned, which is likely but not guaranteed to be the one they were initially instructed to move towards.

5 Results

The instances used are described in subsection 1. The replication of the original results obtained by Reyes et al. (2018) is presented in subsection 2. Subsections 3 and 4 present various experiments performed under the autonomous relocation and centralised relocation settings, respectively, including variations in the number of clusters K, variations in α, effects of the $Delta_{CR}$ parameter and a method for predicting the best $DeltaCR$ for any given instance.

5.1 Instances

The data sets used were provided by the authors of The MDRP, who made them publicly available on GitHub and are the same data sets used in their original paper. There are 240 sets of instances in total, ten for each combination of original and reduced numbers of orders, preparation times, and travel times, as well as historical or optimised courier shifts. I will be using the ten instances with optimised courier schedules and with the original numbers of orders, preparation times, and travel times. Each instance consists of a list of orders, restaurants, and couriers. The instances also include parameters relating to courier speed, service times, and target click-to-door time, which are the same across used instances. I am furthermore grouping the restaurants into clusters using a K-means approach. Under base settings, the optimisation interval is 5 minutes, the assignment horizon Δ_U, as well as Δ_1 and Δ_2, are 10 minutes, the speed is 320 m/s, and the target click-to-door time is 40 minutes. The penalty for service delay β in the bundling procedure is 6, and the penalty for freshness loss θ is 0.003.

5.2 Replication

The base settings of the algorithm are an optimisation interval of $f = 5$ minutes and an assignment horizon of $\Delta_u = 10$ minutes. The following Tables 1,2,3, and 4 summarise the results obtained by changing the optimisation interval, the assignment horizon, and running the algorithm without bundling allowed and without the two-stage commitment strategy respectively.

Table 1 summarises the results of running the base algorithm with optimisation times of 2 minutes vs the default setting of 5 min. The results confirm that the default optimisation interval leads to a higher cost per bundle and more orders per bundle; however, they show that for the considered instances, a shorter optimisation time leads to lower click-to-door and ready-to-pickup times, as well as a slightly smaller percentage of undelivered orders although this difference is very small at 0.02 percentage points. In their paper, Reyes et al. (2018) describe how for instances with higher urgency or higher-order to couriers ratios, the advantage of a 5-minute optimisation interval "evaporates" with the 2-minute interval achieving lower percentages of undelivered orders and lower click-to-door times. Since the number of orders in the instances used has not been reduced, it is likely that they lie higher on the orders-to-couriers scale compared to the full set of instances used by Reyes et al. (2018).

Table 1: Differences in performance of algorithm with more frequent optimizations (2 min) vs default (5 min)

	Optimisation interval				Paired differences	
	5 minutes		2 minutes		(f=2) - (f=5)	
	mean	std	mean	std	mean	std
% undelivered	0.28	0.35	0.26	0.29	-0.02	-0.07
CtoD mean	37.39	2.27	35.65	2.16	-1.74	-0.11
RtoP mean	5.16	0.90	4.67	0.97	-0.48	0.07
cost per order	17.81	1.57	17.67	1.53	-0.14	-0.03
orders per bundle	1.20	0.05	1.12	0.03	-0.08	-0.02

Table 2 summarises the performance difference of the base model when orders are considered for assignment if they are ready within 20 minutes vs the default algorithm, where the assignment horizon is 10 minutes. The longer assignment horizon results in fewer orders undelivered and smaller costs per order, similarly to the results obtained by Reyes et al. (2018). However, the smaller assignment horizon results in higher click-to-door and ready-to-pickup times as well as more orders per bundle.

Table 2: Differences in performance of algorithm with longer assignment horizon (20 min) vs default (10 min)

	Assignment horison				Paired differences	
	10 minutes		20 minutes		$(\Delta_U = 10)$ - $(\Delta_U = 20)$	
	mean	std	mean	std	mean	std
% undelivered	0.28	0.35	0.22	0.27	-0.06	-0.08
CtoD mean	37.39	2.27	35.18	2.06	-2.21	-0.22
RtoP mean	5.16	0.90	3.38	0.86	-1.78	-0.04
cost per order	17.81	1.57	17.65	1.54	-0.16	-0.02
orders per bundle	1.20	0.05	1.17	0.04	-0.03	-0.01

Table 3 summarises the performance difference between using single-stage lazy commitment vs the default algorithm, which uses two-stage additive commitment, for the base model. The results are similar to those obtained by Reyes et al. (2018), with single-stage commitment resulting in a higher ready-to-pickup mean time but lower cost per order and fewer orders per bundle; however, in contrast to their results, the percentage of undelivered orders, as well as click-to-door times, are higher when using a two-stage commitment strategy.

Table 3: Differences in performance of algorithm with longer assignment horizon (20 min) vs default (10 min)

	Commitment strategy				Paired differences	
	Two-stage		Single-stage		(single - two-stage)	
	mean	std	mean	std	mean	std
% undelivered	0.28	0.35	0.25	0.32	-0.03	-0.03
CtoD mean	37.39	2.27	36.31	2.07	-1.08	-0.20
RtoP mean	5.16	0.90	5.18	0.84	0.02	-0.06
cost per order	17.81	1.57	17.75	1.56	-0.06	-0.01
orders per bundle	1.20	0.05	1.13	0.04	-0.06	-0.01

Table 4 summarises the performance difference between setting the maximum bundle size to 1 vs the default setting of effectively allowing bundles. The results mirror those obtained by Reyes et al. (2018) in the way click-to-door times and costs per order decrease when bundling is not possible. However, the ready-to-pickup mean time is also slightly shorter without bundling. However, all of these advantages pale as the results confirm that allowing bundles is critical for the system, with the percentage of undelivered orders being four times higher under the forbidden bundling model.

Table 4: Differences in performance of algorithm with longer assignment horizon (20 min) vs default (10 min)

	Bundling				Paired differences	
	Yes		No		(No - Yes)	
	mean	std	mean	std	mean	std
% undelivered	0.28	0.35	1.07	0.77	0.79	0.42
CtoD mean	37.39	2.27	34.21	1.71	-3.18	-0.57
RtoP mean	5.16	0.90	5.06	1.01	-0.09	0.11
cost per order	17.81	1.57	17.53	1.51	-0.28	-0.06
orders per bundle	1.20	0.05	1.00	0.00	-0.20	-0.05

5.3 Autonomous Relocation

Under the autonomous relocation setting, each instance has its restaurants split into clusters that are meant to represent different areas and neighbourhoods in a city. This is done using a K-means clustering method, with K being the number of clusters to be generated. Figure 1 shows the results of variations in the number of clusters K. Since each instance has a different number K of clusters, aggregate results of the same K across instances cannot be interpreted. As such, instance 8 is chosen as its performance is closest to the average performance across instances and, as such, is considered to be the most representative. The best K for this instance, chosen as the K where the SSE stops decreasing, is 12. I am taking $\alpha = 0.9$ as this value tends to yield the best results. All four performance measures tend to increase as K increases, and, with the exception of undelivered orders, the values for K = 12 are close to the overall trend. K has no effect on the number of orders per bundle. All future results are generated using K = 12 for instance 8, and $K_i = \arg\min_{k \in \mathbb{N}} SSE_i(k)$ in general for all instances i.

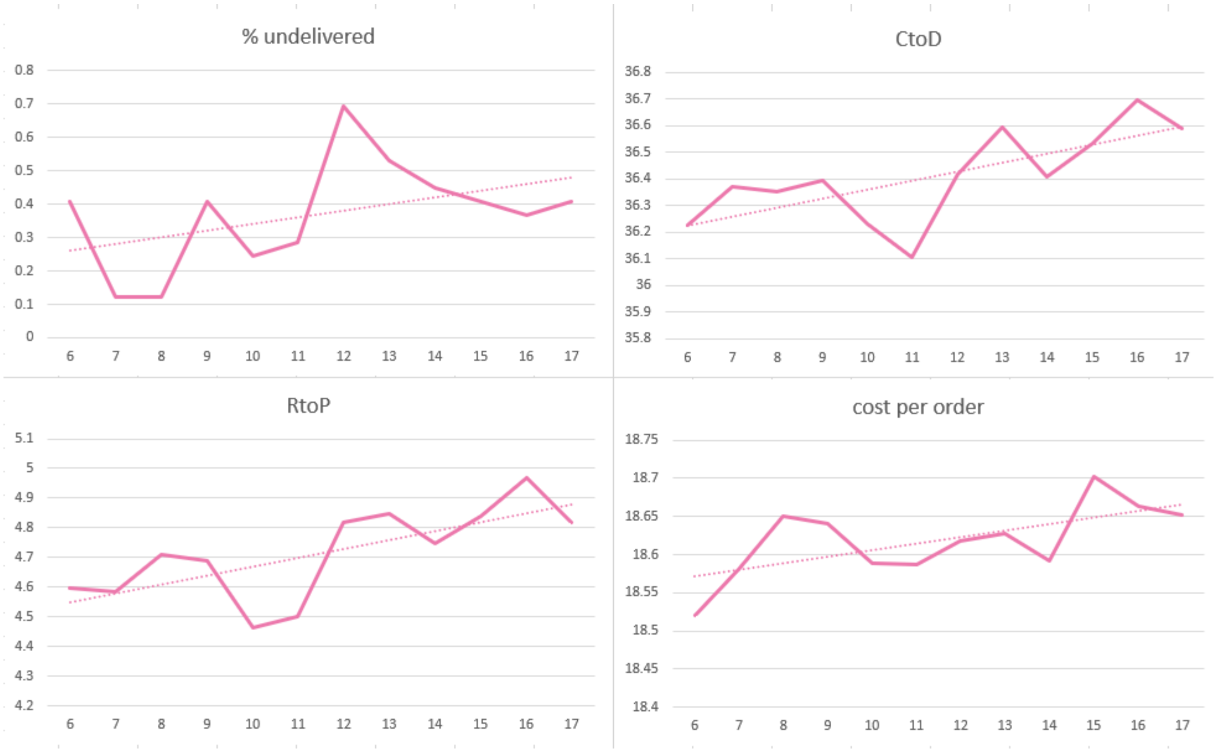

Figure 1: Results of the AR setting for variation in K for instance 8

Table 5 presents the mean results obtained by varying the alpha variable of the AR model. Values for $\alpha \in [0.1, 0.8]$ do not vary and are thus aggregated under a single column. Alpha represents, in this context, an ordinal attribute. Smaller alpha corresponds to a higher priority being placed on how close a cluster is, while higher alpha corresponds to prioritising the profitability of a cluster. The extremes of only proximity or only profitability being considered result in the lowest percentages of undelivered orders. Considering only cluster profitability results in the highest click-to-door and ready-to-pickup mean times, with the "moderate" values resulting in lower times. However, profitability-driven relocation also results in the lowest cost per order. Alpha has almost no effect on bundle size. Overall, $\alpha = 0.9$ seems to yield the most consistently good results, while $\alpha \in [0.1, 0.8]$ yields some of the highest but also some of the lowest performance results. That being said, as proximity and profitability cannot be compared in absolute terms, future analysis will compare the extreme cases where only one goal is being considered by couriers.

Table 5: Results of varying alpha

	0	0.1-0.8	0.9	1
% undelivered	0.24	0.34	0.26	0.24
CtoD mean	37.87	37.64	37.72	38.14
RtoP mean	5.50	5.35	5.40	5.80
cost per order	17.81	17.84	17.81	17.72
orders per bundle	1.21	1.20	1.20	1.21

5.4 Centralised Relocation

For the centralised relocation setting, the parameter Δ_{CR} must be first set to values between 0 and 1. This parameter decides, for each instance, how many restaurants are being considered in the repositioning matching problem. For Δ_{CR} very small, only the restaurants with the most (expected) orders are being considered. Conversely, $\Delta_{CR} = 1$ leads to all restaurants being considered. This parameter is likely to vary across instances. Thus, for each instance, the parameter that minimises the mean click-to-door time is chosen. Table 6 shows the results of the centralised repositioning algorithm for each instance, including the Δ_{CR} that is used with each instance. It is noteworthy that the Δ_{CR} varies for each instance from 0.06 to 0.95 with a mean of 0.315.

Table 6: Results of centralised repositioning

instance	Δ_{CR}	% undelivered	CtoD	RtoP	cost per order	orders per bundle
0	0.15	0	34.77	3.79	17.99	1.17
1	0.25	0	34.79	5.14	16.39	1.14
2	0.10	0	35.00	3.78	19.77	1.13
3	0.08	0	35.52	3.70	19.03	1.17
4	0.05	0	38.79	4.86	16.33	1.22
5	0.60	0.11	41.87	5.80	16.16	1.30
6	0.06	0.90	35.92	5.54	16.62	1.19
7	0.95	0.59	39.04	5.99	16.43	1.25
8	0.30	0.37	37.26	5.42	18.62	1.22
9	0.60	1.15	39.17	6.07	20.57	1.18

The well-tuned centralised repositioning is next being compared to the base algorithm results as well as the results of running the centralised repositioning with the same Δ_{CR} for each instance. I am taking $\Delta_{CR} = 0.315$ for the untuned centralised repositioning problem, as this is the mean value of the tuned Δ_{CR} across instances. The results are summarised in table 7. They show that both tuned and untuned CR have no effect on the number of orders per bundle or the percentage of undelivered orders. Moreover, untuned CR has a very small effect on the cost per order. However, the most striking result is the fact that, while untuned CR performs even more poorly than the base algorithm in terms of click-to-door and ready-to-pickup times, the properly tuned CR achieves better results than the base algorithm. This shows that for a model with centralised repositioning, tuning is crucial but has the potential to significantly improve performance.

Table 7: Summary statistics of tuned and untuned CR compared to the base algorithm

	Base		Untuned CR		Tuned CR	
	mean	std	mean	std	mean	std
% undelivered	0.28	0.35	0.28	0.35	0.28	0.36
CtoD mean	37.39	2.27	37.44	2.23	37.27	2.35
RtoP mean	5.16	0.90	5.19	0.85	5.06	0.96
cost per order	17.81	1.57	17.80	1.55	17.79	1.53
orders per bundle	1.20	0.05	1.20	0.05	1.20	0.05

Properly tuning the centralised relocation setting for each instance is paramount to achieving better results than the base setting - without proper tuning, centralised relocation can perform even worse than the base algorithm. In reality, a trial-and-error tuning process can take many periods and can thus be very costly. As such, it is worthwhile investigating what drives the "perfect" ΔCR and how to obtain it for an instance without having to go through the trial-and-error process. Table 8 shows the results of regressing the best Δ_{CR} for each instance over the average number of orders per restaurant, the number of orders-to-courier, as well as the number of restaurants-to-couriers. The results show that Δ_{CR} is expected to increase, on average, with 0.35 for each additional order per restaurant and with 1.83 for each additional courier per restaurant, and is expected to decrease, on average, with 0.29 with each additional order per courier. All of these results are statistically significant at a 3% confidence level.

Table 8: Results of regressing the best Δ_{CR} for each instance over various instance characteristics

	Coefficient	Estimate	Standard Error	t-statistic	p-value
Constant	β_0	-1.47	0.52	-2.84	0.029
orders/restaurants	β_1	0.35	0.08	4.35	0.005
orders/couriers	β_2	-0.39	0.11	-3.55	0.012
couriers/restaurants	β_3	1.83	0.64	2.86	0.029

5.5 Comparison

Table 9 shows the summary statistics of the results obtained under different model settings. Both of the autonomous relocation models result in a higher percentage of orders delivered, although this difference is very small. In spite of this, the medial value for the delivery rate is much better for the base and CR settings, with the AR setting ending up with 3 and 4 times more orders undelivered for the proximity and profitability-driven relocation, respectively. Thus, the base and CR models achieve 100% delivery rate in half of the instances, while autonomous relocation results in fewer instances with perfect delivery rates and instead evens out the delivery rates across instances. This might be acceptable for proximity-driven relocation; however, profitability-driven relocation results in 1.3% undelivered orders in its worst-case scenario - almost three times higher than what would be considered a "successful" run. Overall, this shows that autonomous relocation tends to hurt delivery rates. Furthermore, it shows the effect of courier drainage on autonomous relocation - when couriers all congregate into areas perceived to be more profitable; these areas become oversaturated, while less profitable areas end up deserted and, depending on the layout of the city, a lot of orders may end up impossible to deliver.

Click-to-door times under both AR settings are longer than either the base setting or the CR, with the CR achieving the shortest times and profitability-driven AR resulting in the longest time. The exact same trend holds for the minimum, median, and maximum values, showing that this trend is consistent across instances. Once again, this shows the effects of courier drainage. Even when a courier is assigned to an order in a less profitable area, the fact that all couriers are effectively moving away from these areas in their idle time causes the travel distance to those areas to be much longer than it would be otherwise. Even autonomous relocation to the nearest areas does little to mitigate this effect. A centrally planned relocation method is the only way to improve click-to-door times.

The exact same trend holds for ready-to-pickup times with even more intensity. Consistent with the explanation offered for the longer click-to-door times, the issue lies with the travel time to the restaurant; the effect is more pronounced in the ready-to-pickup time, which is comprised in a larger percentage of travel time to the restaurant. Thus, ready-to-pickup times under the profitability-driven autonomous relocation are much larger than base and CR settings and are followed closely by the other AR setting. The inefficiency of the profitability-driven AR is most visible in the worst-case scenario - the maximum value over instances - which is almost 20% longer than under the CR setting. Once again, centralised relocation manages to predict the amount of couriers needed at each restaurant and manages to assign couriers to restaurants where they are actually likely to be assigned to orders, thus gaining a time advantage compared to the base model.

The cost per order is highest under the base algorithm and the proximity-driven relocation, with centralised relocation achieving a slightly lower cost and profitability-driven relocation resulting in the lowest cost. The same holds, more or less, for the minimum, median, and maximum values across instances. Thus, while profitability-driven relocation might live up to its name and increase profitability through lower costs per order, centralised relocation achieves similar results while avoiding the pitfalls of the AR setting in terms of delivery rates and times.

The number of orders per bundle is slightly higher for autonomous relocation systems, especially for profitability-driven relocation. A possible explanation is that under autonomous relocation, it is easier for some couriers, who are already closest to the cluster centre, to be assigned multiple orders, while other couriers are not assigned any orders. However, these couriers still have to be paid the minimum hourly wage, on top of the per order wages going to the couriers who do get assignments. *This shows that the phenomenon of courier drainage can affect the costs and, therefore, the profits of meal delivery platforms.*

When comparing the two extreme cases of autonomous relocation, it becomes obvious that relocation to the nearest bundle is preferable. Not only it results in a smaller increase in click-to-door and ready-to-pickup times, as well as better delivery rates, but it also results in less deviation across instances and, therefore, more predictability. The only exception being the cost per order measure which is not only the lowest but also has less variance for the profitability-driven relocation. That being said, centralised repositioning achieves the lowest time measures out of all settings, as well as the second-lowest cost, while maintaining the same percentage of delivered orders as the base algorithm. Overall, centralised repositioning manages to significantly improve on the base model and provides a much better alternative to allowing couriers to relocate freely. Its advantage lies in being able to decide simultaneously for all couriers at once, taking into account what every other courier is going to do - something that even the best trained, most experienced couriers cannot do.

Table 9: Comparison of results obtained using each of the three algorithm settings

		MIN	MED	MAX	AVG	STD
% undelivered	Base	0	0.06	0.92	0.28	0.35
	AR $\alpha = 0$	0	0.23	0.60	0.24	0.21
	AR $\alpha = 0$	0	0.18	1.30	0.24	0.37
	CR	0	0.06	0.92	0.28	0.36
CtoD	Base	34.82	36.65	41.87	37.39	2.27
	AR $\alpha = 0$	35.45	37.24	42.73	37.87	2.04
	AR $\alpha = 0$	35.52	37.40	42.87	38.14	2.30
	CR	34.77	36.59	41.87	37.27	2.35
RtoP	Base	3.71	5.32	6.64	5.16	0.90
	AR $\alpha = 0$	3.80	5.68	6.96	5.50	1.01
	AR $\alpha = 0$	3.82	6.12	9.22	5.80	1.66
	CR	3.70	5.28	6.60	5.06	0.96
cost per order	Base	16.16	17.34	20.59	17.81	1.57
	AR $\alpha = 0$	16.18	17.37	20.41	17.81	1.51
	AR $\alpha = 0$	16.16	17.29	20.10	17.72	1.44
	CR	16.16	17.30	20.50	17.79	1.53
orders per bundle	Base	1.13	1.18	1.30	1.20	0.05
	AR $\alpha = 0$	1.13	1.19	1.31	1.21	0.05
	AR $\alpha = 0$	1.14	1.19	1.32	1.21	0.05
	CR	1.13	1.18	1.30	1.20	0.05

6 Conclusion

At the beginning of this paper, I set to replicate the results of the MDRP obtained by Reyes et al. (2018) and to perform the same experiments on two new settings - one allowing for autonomous relocation and one allowing for centrally planned relocation. The results showed that for the given instances, a shorter optimisation interval as well as a longer assignment horizon can improve delivery rates, click-to-door times, ready-to-pickup times, and order costs, while also decreasing bundle sizes. Van Dun et al. (2020) show that improving customer satisfaction tends to drive up profits by ensuring not only more customers, but also that these customers place orders more often. Thus, it is likely that by decreasing optimisation intervals and increasing assignment horizons, these instances can not only benefit from lowers costs per order, but also higher revenues thanks to higher customer satisfaction. Moreover, bundling proves to be crucial for ensuring satisfactory delivery rates.

The results from experimenting with the AR and CR settings showed that a system with autonomous relocation tends to perform worse than one where couriers remain in place while idle, regardless whether couriers are moving to nearby centres of activity, or are all converging to one most profitable centre. That being said, the AR model where couriers prioritise distance to their repositioning destination performs better in terms of delivery rates as well as well as delivery times and tend to be more predictable than a profitability-driven repositioning system. This proves that when couriers are allowed to reposition while idle, the more they tend to congregate in areas perceived as profitable, the poorer the system performs. However, even when couriers try to simply relocate to neighbourhood centres, the effect of courier drainage still holds, but on a smaller scale - instead of profitable neighbourhoods becoming oversaturated, the centres of neighbourhoods are oversaturated.

The CR system can be very responsive to parameter changes. This means that, when properly tuned, centralised repositioning assignments can significantly improve the performance of a meal delivery system, but without this tuning, they can hurt the performance of the system (although they are always an improvement on systems with autonomous relocation). That being said, in practice, the larger an instance is, the more costly it can be to generate centralised repositioning assignments, and the more costly it can be to do the market research required to achieve a well-tuned centralised repositioning algorithm. Moreover, this paper maintains the assumption that all travel and service times are known beforehand and do not vary. Without this assumption, centralised repositioning assignments become more difficult to generate. As such, for large platforms as well as platforms with fewer resources, centralised repositioning may be infeasible. In this case, in light of the results comparing the performance of the various settings, if centralised repositioning is not an option, the platform might effect improvement in service quality by either instructing couriers to remain at their last position when finishing an assignment or, at least, to prioritise proximity over profitability in relocation destinations, and to head for local neighbourhoods close to their initial position, rather than set on longer trips to the city centre. For example, a courier finishing an order in the south of Rotterdam is advised to move to the Zuidplein area, over Rotterdam Centrum.

As the market size for meal deliveries is set to increase, the continued development of MDRP-related technologies is more important than ever, particularly with respect to courier behaviour modelling. Thus, a worthwhile direction for further research can be into other areas of courier autonomy, such as order rejection as well as more elaborate courier payment schemes. Without liveable and fair wages, couriers are likely to keep striking, and their numbers will dwindle as they will move toward jobs that are more profitable for them. Thus, it is paramount that meal delivery platforms are not only able to make a profit but keep that profit even while employing a fair pay scheme. Therefore, further research into courier pay schemes and their impact on other qualitative and economic performance measures is necessary.

References

Aghayev, I. (2022). Wolt and bolt couriers go on strike in azerbaijan.

Ahuja, K., Chandra, V., Lord, V., and Peens, C. (2022). Ordering in: The rapid evolution of food delivery.

Chen, X., Ulmer, M. W., and Thomas, B. W. (2022). Deep q-learning for same-day delivery with vehicles and drones. *European Journal of Operational Research*, 298(3):939–952.

Cornwell, A. (2022). Foreign food delivery drivers stage rare strike in dubai.

France-Presse, A. (2022). Gig economy: Egypt's delivery riders face rough ride.

Grubtech (2021). How did food delivery start? history of food delivery.

Jahanshahi, H., Bozanta, A., Cevik, M., Kavuk, E. M., Tosun, A., Sonuc, S. B., Kosucu, B., and Başar, A. (2021). A deep reinforcement learning approach for the meal delivery problem. *Knowledge-Based Systems*, 243.

Quintero-Rojas, S. (2020). Computational framework for solving the meal delivery routing problem.

Reyes, D., Erera, A., Savelsbergh, M., Sahasrabudhe, S., and O'Neil, R. J. (2018). The meal delivery routing problem.

Shoshiashvili, T. (2021). Wolt and bolt couriers go on strike in azerbaijan.

Statista (2021). Online food delivery - worldwide: Statista market forecast.

Steever, Z., Karwan, M., and Murray, C. (2019). Dynamic courier routing for a food delivery service. *Computers & Operations Research*, 107:173–188.

Stone, E. (2022). Stuart and just eat courier drivers are still on strike after almost 100 days over pay and conditions.

Tao, J., Dai, H., Jiang, H., and Chen, W. (2020). Dispatch optimisation in o2o on-demand service with crowd-sourced and in-house drivers. *International Journal of Production Research*, 59(20):6054–6068.

Thomke, S. (2014). Mumbai's models of service excellence.

Ulmer, M. W. and Thomas, B. W. (2020). Meso-parametric value function approximation for dynamic customer acceptances in delivery routing. *European Journal of Operational Research*, 285(1):183–195.

Van Dun, C., Fehrer, T., Kratsch, W., and Wolf, N. (2020). Customers like it hot and fast – incorporating customer effects into the meal delivery process.

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

Zou, G., Gao, M., Tang, J., and Yilmaz, L. (2021). Simulation of online food ordering delivery strategies using multi-agent system models. *Journal of Simulation*, page 1–15.

7 Appendix A

Base algorithm used to solve the MDRP.

Algorithm 3 Replication

while $time < 1440$ **do**
    $D \leftarrow \{c \in C | e_c \leq time$ **and** $l_c > time$ **and** $e_d > time + \Delta_2\}$
    Initialise the three priority lists
    $Z \leftarrow \dfrac{|\{o \in O | undelivered \textbf{ and } e_o \leq time + \Delta_1\}|}{|D|}$ or Z=1 if D is empty
    for all restaurants r **do**
        $U_r \leftarrow \{o \in O | undelivered$ **and** $e_o \leq time + \Delta_U$ **and** $a_o < time\}$
        $k_r \leftarrow$ number of couriers at r
        $S_r \leftarrow$ Set of bundles generated using **Procedure 1**
        Bundles from S are allocated to priority groups according to the **priority scheme**
    for all priority groups **do**
        Generate assignments using the default **linear assignment model**
        Update assignments using the **commitment strategy**
        for all courier-bundle pairs (d, s) **do**
            if d has a partial commitment to s **then**
                $e_d \leftarrow \max(time, e_d) + t(d, r)$
                Update d location to r location
            if d has a full commitment to s **then**
                $arrivalTime \leftarrow \max(time, e_d) + t(d, r)$
                $pickupTime \leftarrow \max(arrivalTime + \dfrac{serviceTime}{2}, \arg\max_{o \in s}(e_o))$
                $deliveryTime \leftarrow pickupTime + \dfrac{serviceTime}{2}$
                Update d location to r location
                for all orders o in s **do**
                    $deliveryTime \leftarrow deliveryTime + t(d, o) + \dfrac{serviceTime}{2}$
                    Update d location to o location
                    $CtD \leftarrow CtD + deliveryTime - a_o$
                    $RtP \leftarrow RtP + pickupTime - e_o$
                    Update order o delivery status and total number of delivered orders
                    $deliverytime \leftarrow deliveryTime + \dfrac{serviceTime}{2}$
                $e_d \leftarrow deliveryTime$
        Update D by removing couriers with partial/full commitments assigned
    $time \leftarrow time + f$