Optimization of Orders Assignment to Couriers for On-demand Delivery

Tomás Rocha
Instituto Superior Técnico
tomasrocha@tecnico.ulisboa.pt

The on-demand economy has shaped the habits of consumers, who now expect faster deliveries. Instant deliveries gained popularity for carrying meals to urban areas, and the spectrum of products available to order has broadened. The platforms that provide these services rely on crowdsourced couriers, who use their personal vehicles, resulting in a heterogeneous fleet. Companies experience intense competition to retain both customers and couriers. Hence, it is vital to develop superior optimization models integrating multiple types of vehicles, capable of producing assignments in real-time to meet customers and couriers' expectations. An optimization model is developed to solve the assignment problem with vehicle restrictions using the Jonker-Vogenant and Branch-and-Cut algorithms. The mathematical model is inserted into a dynamic framework that continuously solves it, while controlling the arrivals of orders, couriers' shifts and performing position updates. The model also contemplates dynamic traffic congestion and regional speed limits for different types of vehicles. Besides a myopic assignment approach, the implementation of policies is investigated – extended assignment policy and bicycle policy – in order to improve performance along different metrics. Based on real-world instances, the proposed model with the extended assignment policy achieves a decrease in total delivery time of approximately 4.5% and an increase of 9.6% in balanced courier utilization when compared with the real assignment, thus improving the solution from the customer and couriers' perspectives. The bicycle policy achieves an increase in balanced courier utilization of 3p.p., at the expense of a 0.3% increase in delivery time compared to the baseline model.

Keywords: on-demand, instant delivery, courier assignment, dynamic assignment problem, heterogeneous fleets, mixed-integer programming.

I. Introduction

The demand for fast delivery options has surged in recent years. The technological developments that followed the advent of the internet and the mass adoption of smartphones had a profound impact on the consumers' behavior and expectations. The same technologies connected businesses and customers, frequently cutting the middleman and allowing for the coordination of self-employed workers. Specially in urban areas, there has been a push for faster deliveries whose highest expression is the instant delivery (ID) that takes 45 minutes or less (Dablanc et al. 2017). Recently, the lockdown response to the Covid-19 pandemic made more people open to try IDs especially for meals, resulting, for some countries, in a seven time increase in sales (Ahuja et al. 2021) compared to 2018 levels. With the end of restrictions, the growth has slowed down, however the habit stuck and now a broader audience regularly uses these services. The exponential growth IDs experienced in the last decade was sustained largely on the low investment required due to crowdsourced workers bringing their own vehicles. The other side of this coin is that couriers can choose schedules, reject requests and decide where to wait. The fleet is heterogeneous with vehicles with different carrying capacity, speed, range, susceptibility to congestion and restrictions to circulation.

The literature concerning IDs is scarce and directed to the meal delivery niche. Most works do not contemplate heterogeneous fleets, or do so incompletely, and do not consider the dynamic nature of couriers' schedules or congestion. However, all these factors influence the real-life problem and must be pondered to ensure the best assignment decisions are made.

Given that the industry has not yet consolidated and competition for customers, partners and couriers is intense, having assignments that bring value to all parts involved is of vital importance to the short to long-term success of businesses. It is important to build models that generate results in real-time for large instances of data, but also to study how various policies affect the often-conflicting objectives. The success in building better models has the potential to reduce costs to both the company and customers and ensure that the workforce is retained and motivated.

The present paper is propelled by a practical case study of an ID platform operating in London Metropolitan Area. The problem studied concerns the assignment of orders to couriers subject to vehicle restrictions, that is affected dynamically over the course of a day. Therefore, developing an optimization model to answer the aforementioned problem is the motivation behind this research.

II. Literature Review

IDs are remarkably distinct from conventional distribution, meaning that traditional delivery problems cannot be applied. On the other hand, being a new field of study means that there is a lack of uniformity, evidenced by the absence of a clearly defined general purpose ID problem. For the most part, IDs have been studied in the form of meal delivery problems. These problems capture many real-world aspects, but usually let other factors such as order and fleet heterogeneity unexplored.

Chen et al. (2022) studied the problem of IDs where couriers depart and return to centralized depots. The formulation incorporates the effects of street networks and traffic by multiplying the Euclidean distance by a constant. The model also defines a cut-off time after which no changes to the delivery are allowed and has the objective of maximizing customers served using Deep Q-Networks.

Li et al. (2022) proposes the use of transfer stations for meal delivery to expand the range of restaurants. These stations are placed based on the time-weighted distance between customer and restaurant nodes to convert long-range tours into multiple smaller trips that are assigned to regional couriers that can pre-position themselves on the transfer stations for minimal delay. First, regional orders are generated. The long-distance orders are added to build new test instances. Then, long-distance orders are grouped by a Density Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm and the routing is re-optimized for each region, including normal and split paths. The objective function minimizes travel time while also penalizing lateness if the customer time windows are not respected. The author uses an Adaptive Large Neighborhood Search (ALNS) heuristic to solve the model.

Liao et al. (2020) developed a meal delivery model that allows for the pooling of orders into a single job. Each courier has a capacity from one to five boxes that limits the order combination. Of the reviewed literature, this article is the only that considers multiple objectives – average customer satisfaction (minimize delivery time with lateness penalties and earliness bonuses), total carbon footprint and the scheduling equalization utilization rate of couriers. The author uses k-means clustering paired with Principal Component Analysis (PCA) to cluster similar orders in the same trip. In a subsequent phase, the initial assignment and routing obtained via a Nondominated Sorting Genetic Algorithm II (NSGA-II) is optimized using ALNS.

Liu (2019) studies the use of drones, instead of couriers, for meal delivery. Despite the use of autonomous vehicles, the author incorporates heterogeneous fleets since drones have different speeds and capacities limited by weight. Besides, the author considers two types of meal – hot and cold – that cannot be transported together. The weight of the load affects not only the speed but also battery consumption. Requests arrive dynamically and are assigned random preparation times, customer time windows and weights. The objective is to minimize travel time in order to evade battery swaps and the model is solved using a progressive dispatch algorithm.

Reyes et al. (2018) published and named the first Meal Delivery Routing Problem (MDRP) and implemented a strategy that pursues an ideal bundle size. This target can be set beforehand or can be adjusted dynamically based on the number of couriers and orders. The orders are then pooled together aiming at the target. The algorithm allows for more orders to be inserted above the maximum threshold of a bundle only if it improves the overall route efficiency. The model minimizes total delivery time while penalizing lateness and is solved using an exact approach, to act as a reference, and a rolling-horizon approach using an ALNS heuristic and compares both.

Steever et al. (2019) address the MDRP and study two policies – split, meaning that more than one courier is involved in pickup and delivery, and non-split, where one courier must visit all the locations and deliver the consolidated order. The non-split option is modeled as a constraint that can be relaxed for the split scenario. The author concluded that a split strategy is effective in ensuring freshness but also increases the operational cost when compared to the non-split policy. The model dynamically updates travel time to simulate traffic and weather conditions and allows for courier diversions while the courier drives to a pickup. An objective function is used that maximizes earliness while penalizing lateness and compares it with two other functions – minimizing the time since an order is ready to pickup until it is delivered or minimizing the total travel time. The model is solved using an auction-based heuristic to cope with dynamic orders and the necessity of re-solving the problem with every new entry. Whenever a request arrives, a sub-problem for each active courier, considering the already attributed orders plus the new one, is solved and the objective evaluated. Then, in the myopic approach, the courier with the maximum bid is assigned to the customer. A proactive variant is also proposed that not only considers the bid value, but also future looking metrics of equity – distance between the nearest courier and all restaurants – and dispersion – scattering of couriers over the grid. The results from the heuristic are compared with those of an exact method that does not consider the future looking measures.

Tu et al. (2020) develop a delivery model that incorporates dynamic orders and courier arrivals, courier exits and travel times to simulate traffic. The model includes pre-dispatching constraints that prevent couriers from being concentrated in the same region, by instead serving requests from more distant outlets. The authors calculate cost based on travelled distance and attribute a penalty cost for tardiness. This cost is minimized and solved using a hybrid metaheuristic based on ALNS, to ensure diversity in the solutions, and it is balanced with tabu search (TS) to intensify and improve assignments and routes.

Ulmer et al. (2021) addressed the variability of orders by considering stochastic preparation times. The author applied a policy to direct couriers to the nearest empty restaurant after delivering all assigned orders. The restaurant is only viewed as empty if no other idle courier is stationed there, ensuring that couriers are distributed across multiple restaurants if orders are scarce. The author suggests that other relocation policies should be studied in future works.

Table 1 compares the most relevant features of this work with similar papers in literature. Thus, the model developed and presented in this article makes four primary contributions to the literature regarding on-demand deliveries (ODD):

1) The inclusion of vehicle restrictions that are imposed based on availability of parking space in urban areas and on the size of orders.
2) Considering fully heterogeneous fleets, with different types of vehicles, each with a unique speed, susceptibility to congestion and capacity.
3) The inclusion of dynamic order arrivals, courier arrivals and exits, as well as traffic congestion.
4) The incorporation of the layout of street networks (circuity and speed limits) that differ by region and affect travel time.

***Table 1**. Comparison between this work and literature papers.*

Reference	Heterogeneous Fleet	Crowdsourced Couriers	Traffic	Vehicle Restrictions	Dynamic Requests
Percentage of papers %	*63%*	*38%*	*25%*	*0%*	*86%*
Chen et al. (2022)	✓				✓
Li et al. (2022)					✓
Liao et al. (2020)	✓				
Liu (2019)	✓				✓
Reyes et al. (2018)		✓			✓
Steever et al. (2019)	✓	✓	✓		✓
Tu et al. (2020)	✓	✓	✓		✓
Ulmer et al. (2021)					✓
This work	✓	✓	✓	✓	✓

III. METHODOLOGY

A. *Problem Description*

The ID problem consists in finding the optimal assignment of orders to couriers, while respecting a set of constraints. Uncertainty is associated to both couriers that can log in and out of the platform and orders that are instantaneously put forward by customers and none of this information is known beforehand. Couriers start their day at one location and have an associated mode of transport, which for the case study is limited to bicycle, car and scooter, and cannot be changed. Each type of vehicle has a different speed. Naturally motorized vehicles are quicker, however can be slowed down by traffic that varies regionally and dynamically throughout the day. The platform automatically limits the type of vehicle that can be assigned to an order based on the size of the order, the availability of parking spaces, limitations to the circulation, among other factors. Besides, each order has a pickup and delivery point that must be visited by the same courier. Assigned couriers must travel to a given location to pick the items to deliver. After delivering an order, the courier can either stay at the delivery location waiting for a new assignment or proactively move to an area with higher demand.

B. *Dynamic Framework*

Modelling IDs requires the incorporation of dynamic elements, namely the arrival of orders, changes in the fleet of couriers and the city traffic throughout the day so as to achieve a sincere representation of the real-world. To capture the dynamic aspects, Pillac et al. (2013) outline various strategies, among them periodic re-optimization, which serves as the backbone for the presented framework.

Figure 1 illustrates a simplified flowchart where some actions were condensed and represented as processes. The first step is to import data containing information about couriers, orders and parameters as well as a timeline that initially lists the entries and exits of couriers and the arrival of orders. Courier data includes a unique identifier for each courier, the type of vehicle and the location at which the courier starts the shift. Likewise, order data includes an identifier, information of the types of vehicles that can serve the request and the associated pickup and delivery locations, as well as the submission time. The parameters include the speed of each type of vehicle by region, the traffic coefficients, the regional circuity factors (ratios of network and Euclidean distance that are incorporated to increase the precision of distance estimate) and the service times at the pickup and delivery locations. Having imported the data, additional empty lists are created to hold the couriers available at the moment, the orders that have been submitted and have not yet been picked and the current assignments. Two databases are created to store the final records of assignments and failed assignments to aid with the result collection. Afterwards, the model is initialized by setting the starting and end time and step.

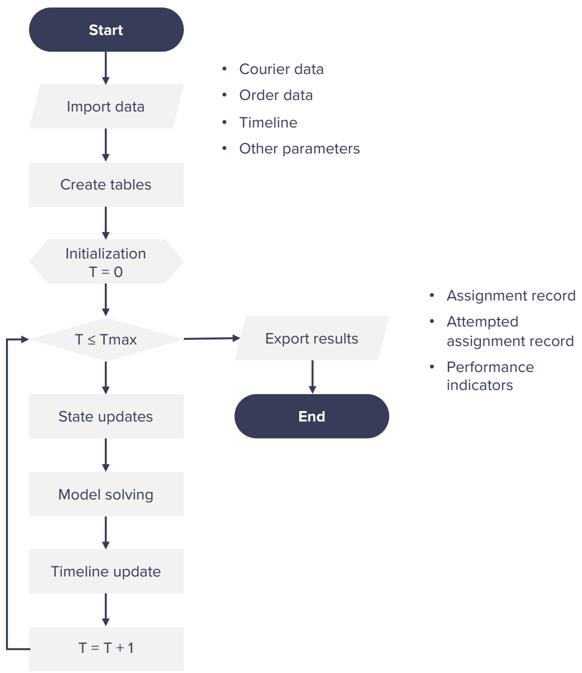

***Figure 1.** Dynamic problem workflow.*

Represented as a single process in the flowchart, the state update phase condenses a series of tasks that must take place before solving the model, including updating the lists of active couriers and unserved requests, the position of couriers in-transit and the arrival times of couriers.

For the first step, the timeline is filtered for current time, then a series of routines are triggered depending on the events that take place during that instant. Arrivals of couriers or orders imply that the correspondent identifiers are added to the available couriers and unserved orders respectively. Courier exits are applied immediately if the courier is idle or, if the courier is occupied with a job, are postponed until the order is delivered and only then is the courier removed from the list. Since it is assumed that couriers and orders can be reassigned up until the picking, both remain in the available lists. This strategy aims at preventing premature assignments by increasing the window of opportunity, allowing for new couriers or orders to arrive and provide a better overall outcome, while ensuring that the courier is on the way. The strategy has a downside which is the increase in the running time of the model, however, since the variation is small, it is a trade-off worthy of making. When the courier arrives at the picking location, both the order and the courier are removed from the available lists and the match is made permanent by removing it from the assignment database and placing it in the permanent record. After the order has been carried out, the courier re-enters the available courier database. Having completed all the updates, the timeline is cleaned of all events reported in the temporary assignment table ("pickup start order", "pickup start courier" and "delivery end") that were not made permanent before re-solving the problem, ensuring that only the best matches remain in the timeline.

After updating the lists of available couriers and orders, the second step is to update the courier position. Despite the assignments only becoming permanent after picking, couriers start to drive towards the location. Since the couriers are kept in the available database, if a new order drops closer to the courier or a new courier enters the system, this results in having to find a new optimum which requires knowing the current position of the courier. This is done in the second step by calculating the coordinates of an intermediate point in the line between the courier initial location and the pickup location that varies linearly as a function of time, *e.g.*, if a trip takes 10 minutes, at 5 minutes time, the courier will be exactly halfway between the two points. With the updated list of couriers, orders and locations, the distances between points and the consequent times are computed.

After state updates the AP Mixed-Integer Linear Programming (MILP) model can be solved using one of two solution methods. Solving the iteration of the model results in the creation of a new temporary assignment database. For every pair of courier and order, three events are created and added to the timeline ("pickup start order", "pickup start courier" and "delivery end"). Following this stage, the step is added to current time and the algorithm will have another iteration if the end time has not been reached. When the end of the running horizon is reached, a table with the final assignments for the running horizon is printed, as well as a report with Key Performance Indicators (KPIs).

C. Mathematical Formulation

Sets

- K Set of available couriers
- O Set of unserved orders
- V Set of vehicle types
- R Set of regions

Parameters

- l_k^K Current location of courier k
- l_o^P Pickup location of order o
- l_o^D Drop-off location of order o
- ω_{vk}^K 1 if courier's k vehicle is of type v, and 0 otherwise
- ω_{vo}^O 1 if order o can be assigned to a courier with a vehicle of type v, and 0 otherwise
- T_o^s Submission time of order o
- T_{ko}^A Assignment time of order o to courier k
- T_{ko}^P Picking start time of order o by courier k
- T_{ko}^D Delivery end time of order o by courier k
- d_{ko}^P Distance between courier k and the pickup location of order o
- d_o^D Distance between the pickup and drop-off locations of order o
- c_r Circuity factor of region r
- s_{vr} Speed of vehicle of type v for region r without traffic
- μ_v Congestion factor of vehicle of type v
- t_{ko}^P Travel time for courier k to get to the pickup point of order o from its current location
- t_{ko}^D Travel time for courier k to get to the drop-off point of order o from the pickup point of order o
- τ^W Waiting time for the acceptance of an order
- τ^P Pickup time
- τ^D Drop-off time

Variables

- x_{ko} Binary variable equal to 1 if courier k is assigned to order o, and 0 otherwise.

Objective Function

Considering the notation, the AP is formulated as a minimization problem. Objective function 3.1, instead of cost, minimizes the sum of total delivery times, from moment of placement to delivery.

$$\min \quad \sum_{k\in K}\sum_{o\in O}(T_{ko}^D - T_o^s)\cdot x_{ko} \tag{3.1}$$

Constraints

Constraints 3.2 ensure that all requests are served. Constraints 3.3 restrict the maximum number of tasks a courier can perform at each time to one. Constraints 3.4 enforce the vehicle restrictions, ensuring that only couriers driving admissible vehicles to each order can be assigned to it. Constraints 3.5 ensure the domain of the decision variable is respected.

$$\sum_{k\in K} x_{ko} = 1 \qquad o \in O \tag{3.2}$$

$$\sum_{o\in O} x_{ko} \leq 1 \qquad k \in K \tag{3.3}$$

$$\sum_{v\in V} \omega_{vk}^K \omega_{vo}^O \geq x_{ko} \qquad k \in K, o \in O \tag{3.4}$$

$$x_{ko} \in \{0,1\} \qquad k \in K, o \in O \tag{3.5}$$

Auxiliary Computations

Equation 3.6 gives the distance between two points in a sphere, where E is Earth's radius and all latitudes (φ) and longitudes (λ) are in radians. The formula is used to determine the distance between couriers and pickups $d_{ko}^P = d(l_k^K, l_o^P)$ and the distance between each order's pickup and delivery locations $d_o^D = d(l_o^P, l_o^D)$. Every location parameter corresponds to a pair of coordinates $l = (\varphi, \lambda)$.

$$d = E \cdot 2 \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (3.6)$$

Equation 3.6 returns the shortest distance between two points, which in urban environments is seldom the case due to infrastructure or natural features constraining the direction of movement. This is handled by multiplying a regional circuity factor to the distance. Durations are given by equation 3.7 by dividing the real distance, that is equal to the linear distance multiplied by a circuity factor c_r, by the speed of the courier with a vehicle of type v in region r, while incorporating the slowing effect of traffic for the current time on motorized vehicles.

$$t_{ko} = \frac{d_{ko} \cdot c_r}{s_{v,r}}(1 + \mu_v) \quad (3.7)$$

Equations 3.8 and 3.9 compute the time at which the courier starts the pickup and the time at which the order is delivered to the customer, respectively.

$$T_{ko}^P = T_{ko}^A + t_{ko}^P \quad (3.8)$$

$$T_{ko}^D = T_{ko}^P + \tau^P + t_{ko}^D + \tau^D \quad (3.9)$$

D. Solution Approach

The dynamic framework continuously calls an algorithm to solve the updated assignment problems. Two exact methods are tested, the Jonker-Volgenant (JV) algorithm (Jonker and Volgenant 1987) that solves the assignment problem and requires the previous incorporation of the vehicle restrictions into the cost matrix, and a Branch-and-Cut (BC) algorithm (Dijkstra 1959) included in a mathematical optimization solver – Gurobi, that solves the mathematical model with vehicle restrictions.

IV. EXPERIMENTS AND RESULTS

A. Baseline Analysis

The simplest way of matching couriers and orders is to do so as soon as new orders arrive and not changing the assignment even if in a later instant a different match is better. This myopic approach is labeled as baseline to distinguish it from the alternative policies and is useful to serve as a benchmark. The assignment algorithm (JV or BC) is invoked every minute of a day, starting at 5am (minute 0) and ending at 12am (minute 1140), and solved to minimize the total delivery time at every instant. These results are compared with the assignments made by the delivery platform in real life. The list with the courier-order pairs does not necessarily correspond to the optimal assignment obtained by the company's Hungarian Algorithm (HA) (Kuhn 1955) because couriers can reject a job and the final assignment might not be the ideal one. However, for the purpose of comparing the developed approach with the real results, the assignments presented by the company in this list are used. For each courier and order pair, delivery time is computed using the same parameters of speed, service time, congestion and circuity factors used in the developed approach. Each courier has an initial location when entering the system and it is assumed that the first order of that courier starts from that point. If a courier is assigned to more than one order, the starting location is not its initial point, but rather the delivery location of the previous order.

Table 2 resumes relevant KPIs for the two algorithms and for the real assignment. In terms of the total objective, there is a difference, albeit small, between the JV and BC algorithms. This might seem paradoxical since both approaches are exact, however, the discrepancy is due to the way both algorithms assign orders to a heterogeneous fleet in a tie. The JV algorithm starts assigning from right to left side of the cost matrix. By contrast, the BC algorithm selects the first optimal solution found, which might not coincide with the exact pair selected by the JV approach. This does not present any difference at the instant it happens but may start a chain reaction that has consequences in later instants, resulting in a difference of around 0.5% or 14 seconds per order.

Table 2. *Baseline policy and real assignment KPI comparison.*

KPI [unit]	JV	BC	Real
Total Delivery Time [min]	47 599	47 844	48 002
Requests Served [%]			
Bicycle	*31%*	*32%*	*20%*
Car	*31%*	*31%*	*24%*
Scooter	*38%*	*37%*	*55%*
Average Delivery Time [min]			
Bicycle	*38*	*38*	*35*
Car	*43*	*44*	*47*
Scooter	*50*	*51*	*48*
Average Courier Utilization [jobs/courier]			
Bicycle	*7.5*	*7.7*	*4.9*
Car	*4.9*	*4.9*	*4.2*
Scooter	*4.8*	*4.7*	*6.9*
Variation Courier Utilization [jobs/courier]	3.3	3.4	3.5

The total delivery time for the real assignment and the model with the two algorithms is close. Since the model solves all 1 062 requests, the average delivery time is 44 min 49 s for the JV approach (best) and 45 min 12 s for the real assignment (worst) which represents just an improvement of less than 1%. This difference is negligible and can be attributed in part due to these not being the optimal assignments, but instead the accepted ones, and due to the fact that the real matches of couriers and orders did not incorporate congestion and the regional variability of roads. Both algorithms assign, approximately, one third of the requests to couriers with each type of vehicle. By contrast, the real assignment produced by the company attributes more than half of the matches to scooters, a fourth to cars and a fifth to bicycles. The results of the mathematical model, with either approach, might seem more even but, when juxtaposed to the average fleet composition (22% bicycles, 35% cars, 43% scooters), it becomes clear that both approaches result in proportionally more bicycles being matched at the expense of cars and especially scooters, which is likely the result of considering the same congestion factors and regional speeds for motorized vehicles. The real model, on the other hand, favors scooters at the expense mostly of cars.

A challenge that ODD platforms with crowdsourced workers face concerns the difficulty in hiring and retaining couriers, in part due to the existence of competing firms. If the workload of a courier is low, the courier might leave the platform or use it in parallel with other platforms. On the other hand, if the workload is too high, the courier can be overworked and start to reject orders. For these reasons and from a long-term perspective, it is preferable to have a balanced use of the courier fleet, which is measured by the courier utilization indicators. The average number of jobs assigned to bicycle couriers is greater for the model, while the scooter utilization is higher for the real assignment. The standard deviation in courier utilization is also greater for the real assignment, meaning that the discrepancies between couriers are greater. The model solved with the JV approach constitutes an improvement of 4.5% in this metric, and with the BC of 1.5% in balanced courier utilization.

The delivery time can be analyzed not only in total or grouped by vehicle type, but also by period of the day and region. Figure 2 shows the average delivery time from a periodic perspective. The morning period goes from 5am to 9am, the day period from 9am to 2pm, evening from 2pm to 7pm and night from 7pm to midnight. The morning period represents the lowest delivery time because, during this period, congestion is lower and the ratio of couriers to orders is high. Traffic also explains the values for evening and night periods, but fails at justifying the spike, for both real and proposed assignments, during the day period. Instead, this increase is attributed to the ratio of couriers and orders. The period from 9am to 10am is the only time when more orders arrive than couriers are available in the system, furthermore, an overwhelming majority of orders have vehicle restrictions during this period. Of these, 49 orders must be served by scooters between 9am and 10pm when there are only 35 couriers with scooters in the system, which helps explain the higher delivery time for this period.

The real assignment enables pooling, which means that orders can be combined to be delivered by the same courier. This means that when there is a lack of couriers with scooters, instead of waiting for more to arrive or finish the deliveries, the real model can attribute many orders to a single courier, which helps to explain the lower delivery time for the day period.

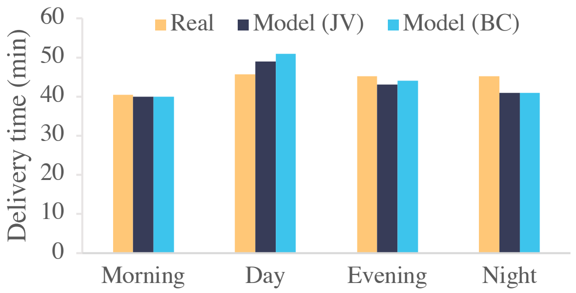

***Figure 2**. Delivery time by period, in minutes.*

Regionally, there are disparities in the average delivery time, which can range from 35 minutes to more than 80 minutes. For most regions, delivery time tends to be closer to the average and the more atypical values are registered in regions where there are fewer requests. Figure 3 represents the map of London, where the colors represent, for each region, the ratio between delivery time of the approach solved by the JV algorithm and the real assignment. If the ratio is higher, it means that the real assignment results in a shorter delivery time and is represented in red. A lower ratio means that the JV algorithm has the shortest delivery time and the correspondent borough is represented in blue. Light grey boroughs have no orders associated, while boroughs in white have an identical delivery time.

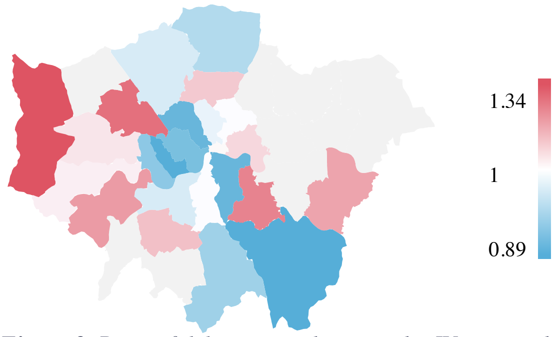

***Figure 3**. Ratio of delivery time between the JV approach and real assignment.*

In the inner part of the city, the JV algorithm approach results in faster assignments, which is the region where most requests originate from. In the outer part of the city, results are mixed and, in the west, the real assignment produces faster deliveries. The cause for these results is likely due to the excess of orders for scooters that take surge in peripheric regions, the starting location of couriers with different vehicles and variability that is more noticeable in regions with less orders.

B. *Policy Analysis*

Besides the baseline assignment approach, the implementation of two policies is investigated – extended assignment policy (1) and bicycle policy (2).

1) *Assignment until Pickup Policy*

The baseline policy considers that immediately after an assignment, the assigned order is permanently removed from the list of orders waiting for assignment and the assigned courier becomes unavailable until finishing this job. This means that, if minutes later a new courier arrives that could improve the overall objective, such option is never contemplated. To prevent this issue, the Assignment until Pickup (AuP) policy is incorporated that allows couriers and orders to remain available for assignment up until the courier reaches the pickup point. This implies that if in the future a courier logs in or finishes a delivery task and is faster to reach the pickup point, the assignment can be changed. The policy also requires the current location of the pre-assigned courier to be updated until the assignment is made permanent or nullified. Figure 4 shows a timeline with all the steps of a delivery in the model. When an order is assigned, there is a waiting time accounting for the time couriers take

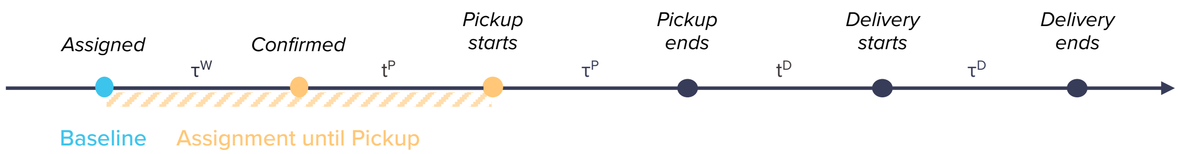

Figure 4. *Delivery timeline with assignment windows for different policies.*

to accept the job (τ^W) and, afterwards, the courier starts to travel towards the pickup location, reaching it t^Pminutes after the confirmation.

The framework with the AuP policy is run using the JV and BC algorithms, considering as input the same instance used for the baseline. Table 3 compares the baseline and AuP policies in terms of total objective and other KPIs. The new policy increased the running time of the model in six and eight times for the JV and BC approaches, respectively, however the model can be used in real-world applications for large instances. This is due to the excess of state updates that this policy adds and the fact that each assignment becomes larger with more couriers and orders.

Table 3. *Baseline and AuP policy KPI comparison.*

KPI [unit]	Baseline		AuP	
	JV	BC	JV	BC
Total Delivery Time [min]	47599	47844	45882	45912
Requests Served [%]				
Bicycle	*31%*	*32%*	*32%*	*31%*
Car	*31%*	*31%*	*30%*	*31%*
Scooter	*38%*	*37%*	*39%*	*38%*
Delivery Time [min]				
Bicycle	*38*	*38*	*37*	*37*
Car	*43*	*44*	*43*	*43*
Scooter	*50*	*51*	*48*	*48*
Courier Utilization [jobs/courier]				
Bicycle	*7.5*	*7.7*	*7.6*	*7.6*
Car	*4.9*	*4.9*	*4.7*	*4.8*
Scooter	*4.8*	*4.7*	*4.9*	*4.8*
Variation Courier Utilization [jobs/courier]	3.3	3.4	3.1	3.1
Time on Permanent Assignments [min]				
Bicycle	*290*	*295*	*285*	*283*
Car	*215*	*219*	*201*	*204*
Scooter	*246*	*240*	*231*	*230*
Time on Non-permanent Assignments [min]				
Bicycle	-	-	*3*	*3*
Car	-	-	*4*	*4*
Scooter	-	-	*6*	*6*

In terms of the total objective, there is a reduction of 1717 minutes for the JV algorithm and 2032 minutes for the BC algorithm when compared to the baseline or, on average, a reduction of 1 minute and 37 seconds and 1 minute and 49 seconds, respectively. The policy results in the objectives for the two approaches becoming more similar since, with more couriers and orders in an assignment decision, the probability of a tie is smaller.

The policy has little to no effect on the percentage of orders attributed to each type of vehicle, yet still affects courier utilization. When compared with the baseline, the AuP policy results in changes in the couriers with no orders and in the couriers with more orders, thus affecting the disparities between couriers. While the baseline policy results in three and two couriers not being assigned to any order for the JV and BC algorithms respectively, for the AuP both algorithms result in only one courier with no assignments. Moreover, the maximum number of orders a scooter courier performs is also reduced. The average courier utilization remains practically the same, with modest increases for scooters and decreases for cars. The standard deviation in courier utilization is reduced to 3.1 jobs per courier, which means that, despite not being the aim of the policy, the distribution of work by the couriers is more balanced.

In terms of delivery time by vehicle type and when compared with the baseline, the AuP policy registers an overall reduction and greater similarity between the JV and BC algorithms. The same can be said for the delivery time for each period and by region. The boroughs of Barnet and Hackney register, on average, longer delivery times with the new policy, although only worse by approximately one minute. The rest of the regions either reduce the delivery time or maintain it. The boroughs of Hillingdon and Richmond upon Thames register the biggest improvements, of approximately 20%. The drastic improvement might be linked to these regions having less couriers that are moved in the early periods to areas with more requests, which would have left the region without the means to respond promptly to new orders. With long assignment windows, if a request arrives in an area without couriers, a distant courier is mobilized and starts driving in that direction. However, if a courier starts or finishes delivering an order and is closer to the pickup, a new match is established and the courier that was coming to aid can remain in the same region and serve local requests.

The AuP policy also raises concerns about the time couriers spend on assignments that fail to materialize. Couriers spend less time delivering using this policy, which is part due to trips becoming more optimized and delivered faster, but also due to the increase in balanced courier utilization. The time spent on non-permanent assignments is on average small (4 minutes and 42 seconds) when compared to the daily delivery activities (234 minutes), equivalent to less than 2% of active time. The time spent delivering by each courier is on average greater on the baseline than the sum of the time spent on permanent and non-permanent assignments combined for the AuP policy. This means that the policy apart from being positive in terms of the customer-oriented objective (delivery time), might also provide benefits for couriers (balanced utilization).

For the entire running period, there is a total of 156 unrealized assignments. Assuming that each one is equivalent to an order, which might not be the case since one order can be re-assigned many times, this implies that 15% of orders or less are re-assigned at some point. Scooters are the most common transport among temporary assignments, making up 48%, followed up by cars with 33% and bicycles with 19%. Congestion makes bicycles a more appealing vehicle for delivery, which explains why they are unassigned less, while scooters make up the bulk of the fleet, making them more likely to get unassigned. Figure 5 represents the number of failed assignments by period and differentiated by type of vehicle for the JV algorithm. The morning registers only seven failed assignments due to the ratio of couriers to orders being high. During the day, this number increases drastically, peaking in the evening and then decreasing at night. The proportion of vehicles assigned to orders is consistent to the fleet composition, with more cars in the morning and bicycles in the evening.

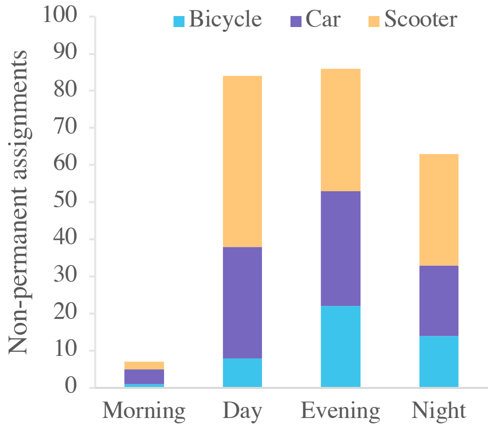

Figure 5. *Number of non-permanent assignments per vehicle type, by period, for the JV algorithm.*

The assignment time comprises the period between an order being submitted and the time at which the order is assigned permanently to a courier. For the baseline model, this period is zero if there are enough couriers that respect the vehicle restrictions, or longer otherwise. In practice, due to the shortage of couriers during some periods, the average assignment time is 35 seconds for the JV and 37 seconds for the BC algorithms, respectively. For the AuP policy, this value averages 3 minutes and 3 seconds for both algorithms. This means that, by taking on average two and half minutes longer on the assignment, the policy is able to reduce the total delivery time, which already includes the assignment time, by 1 minute and 37 seconds for the JV algorithm and by 1 minute and 49 seconds for the BC algorithm, *i.e.*, equivalent to a 3.6% and 4.2% reduction, respectively. Figure 6 represents the average assignment time by period for the JV and BC algorithms. There is a clear peak in the morning period for scooters, that coincides with the appearance of scooter-restricted orders in the region of Brent. Cars are the transport that takes the longest to assign during the day, evening and night periods, which might be partially influenced by congestion. However, since scooters take less time to be matched, other factors such as the initial location of both transports might account for the difference.

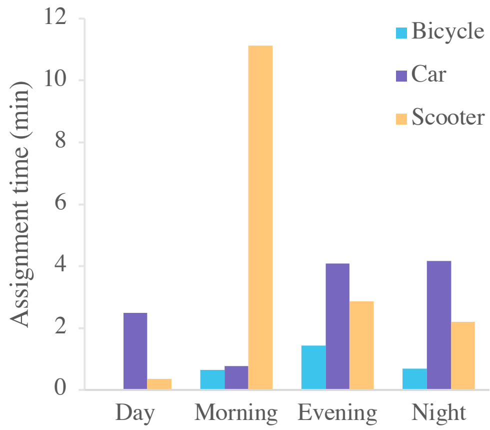

Figure 6. *Average assignment time by period per vehicle type for the JV algorithm, in minutes.*

Regionally, the values for average assignment time of orders are not correlated with the sheer number of non-permanent assignments, meaning that some regions can have many re-assignments that result in orders having a small assignment time or vice-versa, *e.g.*, if an order is re-assigned three times in three minutes, the total assignment time is just three minutes. By contrast, an order can be re-assigned just one time after five minutes of the first assignment and this would result in an assignment time of 5 minutes. The region of Brent is one of a few cases where there are many re-assignments that take on average 20 minutes.

It can be concluded that the policy improves the main objective of reducing delivery time, alleviates the problem of the high number of orders with a restriction in the region of Brent and improves the distribution of work by couriers.

2) Bicycle Policy

For environmental concerns or in anticipation of future legislation limiting the circulation of motorized vehicles, it is interesting to study how can more bicycles be employed and what are the effects in the total objective and other KPIs. The policy consists in giving priority to couriers with bicycles for orders whose total distance, from courier to pickup location plus pickup to delivery location, is less than a fixed value. This is done by reducing the objective value for couriers with bicycles for nearby orders before solving the model. The policy is applied over the baseline approach using the same parameters, but also applied over a set of modified parameters that reduce the congestion of scooters and reduce bicycle speed.

Table 4 shows the values of the KPIs for the baseline and bicycle policy, applied only to orders whose travelled distance is lower than or equal to three kilometers, using the two algorithms. The two policies barely differ in terms of running time, however, the objective values increase around 0.3%, which is expected since the policy introduces an inefficiency by not assigning the order to the best match and instead favoring bicycle couriers.

***Table 4**. Baseline and bicycle policy KPI comparison.*

KPI [unit]	Baseline		Bicycle (d^{max}=3)	
	JV	BC	JV	BC
Total Delivery Time [min]	47599	47844	48146	48077
Requests Served [%]				
Bicycle	*31%*	*32%*	*34%*	*32%*
Car	*31%*	*31%*	*30%*	*31%*
Scooter	*38%*	*37%*	*36%*	*37%*
Delivery Time [min]				
Bicycle	*38*	*38*	*38*	*38*
Car	*43*	*44*	*44*	*44*
Scooter	*50*	*51*	*52*	*52*
Courier Utilization [jobs]				
Bicycle	*7.5*	*7.7*	*6.9*	*7.8*
Car	*4.9*	*4.9*	*4.6*	*4.9*
Scooter	*4.8*	*4.7*	*5.3*	*4.6*
Variation Courier Utilization [jobs]	3.3	3.4	3.4	3.3

Contrary to the baseline results, where the JV algorithm performed better, for the bicycle policy it is the BC algorithm that is faster. Due to the difference being less than one percent, it is not possible to discern if it is caused by the policy or by small variations in the assignments.

Since the policy is only applied to couriers and orders whose total travel distance does not exceed a fixed value, it is important to study how this parameter influences the KPIs. Figure 7 plots the average objective value and the share of requests served by each vehicle type. The graph shows that for a distance above 4.5 kilometers there is a steep increase in delivery time, which is due to a higher number of orders above that distance on one side, and to the fact that misallocating couriers for small distances does not change delivery time much, while misallocating couriers that are further away means that these couriers have to travel longer distances, which lengthens the deliveries. For this reason, the policy should be set only for small distances, which means that the increments in bicycle utilization are also lower. The policy might still be worthy of considering for small distances because it does not significantly influence the delivery time and occupies bicycle couriers with closer orders, leaving the distant orders for motorized vehicles that are not as worn-out by long trips.

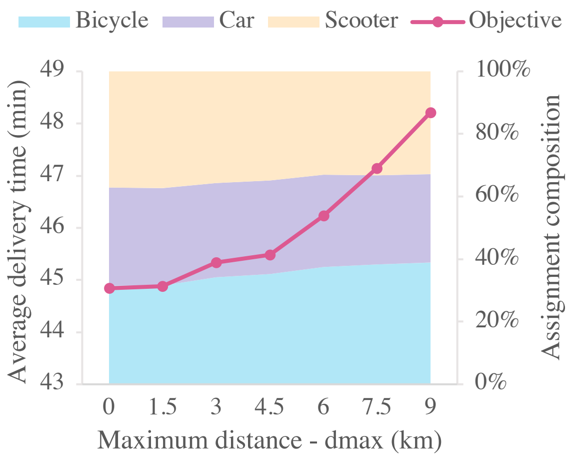

***Figure 7**. Average delivery time and assignment composition for variations in the maximum distance to apply the bicycle policy.*

Speed and congestion parameters heavily influence the model. Furthermore, scooters are less affected by congestion and bicycles are unlikely to be totally immune. Additional experiments are conducted using the same instances applied to previous policies with changes in speed and congestion parameters. The speed of bicycles is reduced to not surpass the speed without congestion of motorized vehicles. Scooter and bicycle congestion parameters are reduced to 60% and 40%, respectively, to that of cars.

The results are analogous to the previous analysis, with an increase in the maximum distance to apply the policy, leading to slower deliveries. At the same time, the proportion of orders performed by bicycles and its utilization increases at the expense of other modes of transportation. What differs relatively to the previous analysis is the magnitude of the changes. Figure 8 shows the average delivery time and the proportion of orders delivered by each vehicle. Compared to Figure 7, the increase in the utilization of bicycles is proportionally higher. The difference in the number of orders served by bicycles rises 12p.p. between the scenario without this policy (distance zero) and the scenario where the policy is applied for a maximum distance of up to nine kilometers. This happens because, with the modified input, motorized transports become proportionally more competitive than bicycles. Using the original input, the model heavily favored bicycles, which meant that the policy gave preference to an already favored mode of transportation. The figure also evidences the fact that, applied to distances greater than 4.5 kilometers, the policy leads to an increase in delivery time that can be detrimental.

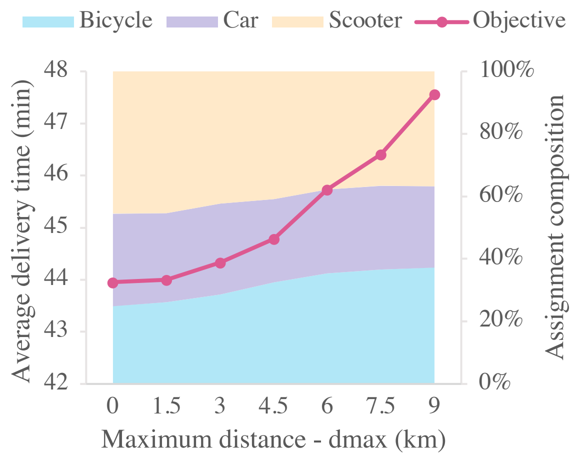

***Figure 8**. Average delivery time and assignment composition for variations in distance to apply the policy for the modified data.*

The results of the policy are mixed. On the one hand, a forced increase in bicycle courier utilization naturally leads to higher delivery times, which impairs the performance of the company at the eyes of the consumer. On the other hand, the increased courier satisfaction or a decrease in direct emissions might still be worth to pursue. An intermediary policy, where bicycles are given preference for distances of three kilometers or less, can be beneficial by not only increasing the usage of bicycles, but also by ensuring that couriers with bicycles deliver closer orders and are not tired due to excessive pedaling.

V. CONCLUSIONS AND FUTURE RESEARCH

IDs are characterized by intense competition not only in gaining customers, but also in hiring and retaining couriers. Hence, the development of an optimization model that, in real-time, assigns couriers to orders for large sets of data is essential to ensure that both customer and courier objectives are satisfied. These results can be further improved by the implementation of assignment policies.

A myopic baseline approach is used, which constitutes an improvement in the total delivery time objective of 0.8% using the JV algorithm and 0.3% using the BC algorithm when compared with the real assignment, and an increase of 4.5% and 1.4%, respectively, in balanced courier utilization. These results are further improved by applying a policy that enables re-assignments until the courier arrives at the pickup point. This policy achieves a reduction in total delivery time of 4.4% and accomplishes an increase in balanced utilization of 9.6% for the two algorithms when compared to the real assignment, while still being fast enough to be used in real applications. On the other hand, this policy has the inconvenient of sometimes making couriers start to work on an order only for the job to be assigned to another courier; however, on average, this rarely happens since, in the model, couriers take 2 minutes to accept an order and the average assignment time with the policy is only 2 minutes and 25 seconds, making the trade-off worth considering. Additionally, a policy that prioritizes bicycle utilization for short distances is also studied. Applied only to orders that require travelling for 3 minutes or less, the total delivery time increases by 0.3% and 0.2% for the JV and BC algorithms, respectively, while the courier utilization rises by 14p.p. compared to the real assignment. Even with this inefficiency, the policy might still be worth applying if a company has the objective of reducing direct carbon emissions. As evidenced by the sensitivity analyses, the model is heavily influenced by pickup time and the relative differences in speed and congestion between vehicle types.

It is suggested that more attention should be devoted to the research of multiple objectives (from the perspectives of customers, couriers and partner restaurants or stores), the inclusion of stochastic courier rejections, order priority, heterogeneous requests (by differentiating size, type of product and source as either a restaurant or store with different processing/pickup times), different dynamic approaches and the addition of drones as a new vehicle type to work along regular couriers. The study of additional policies might provide valuable insights for the application by delivery platforms. Reposition policies that incentivize couriers to move to areas with higher demand of orders or lower supply of couriers might provide great benefits. This can also be adopted to move couriers with motorized vehicles to less congested areas. Additionally, a policy that evaluates the number of vehicles of each type in a fleet and manages the assignment based on preventing future shortages of vehicles might also yield positive results.

VI. REFERENCES

Ahuja K, Chandra V, Lord V, Peens C (2021) *Ordering in: The rapid evolution of food delivery* (McKinsey & Company).

Chen J fang, Wang L, Wang S, Wang X, Ren H (2022) An effective matching algorithm with adaptive tie-breaking strategy for online food delivery problem. *Complex Intell. Syst.* 8(1):107–128.

Dablanc L, Morganti E, Arvidsson N, Woxenius J, Browne M, Saidi N (2017) The rise of on-demand 'Instant Deliveries' in European cities. *Supply Chain Forum Int. J.* 18(4):203–217.

Dijkstra EW (1959) A note on two problems in connexion with graphs. *Numer. Math.* 1(1):269–271.

Jonker R, Volgenant A (1987) A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing* 38(4):325–340.

Kuhn HW (1955) The Hungarian method for the assignment problem. *Nav. Res. Logist. Q.* 2(1–2):83–97.

Li J, Yang S, Pan W, Xu Z, Wei B (2022) Meal delivery routing optimization with order allocation strategy based on transfer stations for instant logistics services. *IET Intell. Transp. Syst.*:itr2.12206.

Liao W, Zhang L, Wei Z (2020) Multi-objective green meal delivery routing problem based on a two-stage solution strategy. *J. Clean. Prod.* 258:120627.

Liu Y (2019) An optimization-driven dynamic vehicle routing algorithm for on-demand meal delivery using drones. *Comput. Oper. Res.* 111:1–20.

Pillac V, Gendreau M, Guéret C, Medaglia AL (2013) A review of dynamic vehicle routing problems. *Eur. J. Oper. Res.* 225(1):1–11.

Reyes D, Erera A, Savelsbergh M, Sahasrabudhe S, O'Neil R (2018) The Meal Delivery Routing Problem. :70.

Steever Z, Karwan M, Murray C (2019) Dynamic courier routing for a food delivery service. *Comput. Oper. Res.* 107:173–188.

Tu W, Zhao T, Zhou B, Jiang J, Xia J, Li Q (2020) OCD: Online Crowdsourced Delivery for On-Demand Food. *IEEE Internet Things J.* 7(8):6842–6854.

Ulmer MW, Thomas BW, Campbell AM, Woyak N (2021) The Restaurant Meal Delivery Problem: Dynamic Pickup and Delivery with Deadlines and Random Ready Times. *Transp. Sci.* 55(1):75–100.