UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

UN PROBLEMA DE RUTEO DE VEHÍCULOS EN UN MODELO DE ECONOMÍA COLABORATIVA

TESIS PARA OPTAR AL GRADO DE MAGISTER EN GESTIÓN DE OPERACIONES

PABLO ANTONIO AZURDUY SALINAS

PROFESOR GUÍA:
MARCEL GOIC FIGUEROA

PROFESOR CO-GUÍA:
CRISTIÁN CORTÉS CARILLO

COMISIÓN:
NICOLAS ARAMAYO BENVENUTTO

SANTIAGO DE CHILE
2022

RESUMEN DE LA MEMORIA PARA OPTAR AL TITULO DE: Magister en Gestión de Operaciones
POR: Pablo Antonio Azurduy Salinas
FECHA: 2022
PROFESOR GUÍA: Marcel Goic

UN PROBLEMA DE RUTEO DE VEHÍCULOS EN UN MODELO DE ECONOMÍA COLABORATIVA

En los últimos años, el desarrollo de las tecnologías de la información ha provocado importantes cambios en el diseño de los sistemas logísticos que las empresas utilizan para distribuir los productos a sus clientes. Entre los cambios más relevantes se encuentra la aparición de plataformas digitales que conectan a las empresas que necesitan entregar sus productos con conductores que pueden abordar el transporte de "última milla". Uno de los atributos centrales de estas plataformas es la falta de relación contractual con los potenciales conductores y, por tanto, cuando una plataforma ofrece una posible ruta, los conductores aceptarán sólo aquellas rutas que les resulten más atractivas. Esta lógica requiere modificar los enfoques tradicionales para resolver el problema de enrutamiento de vehículos (VRP) resultante para generar soluciones que no sólo proporcionen una reducción de los costes de transporte, sino que también generen rutas atractivas para los conductores.

En este trabajo, describimos el problema de enrutamiento de vehículos con conductores no contractuales (VRPNCD) y proponemos un esquema de solución que permite aprender de las preferencias de los conductores e incorporarlas para generar soluciones que cumplan los objetivos del negocio. Aplicando nuestro enfoque de solución a un problema real de una plataforma logística digital (Wareclouds), encontramos que podemos reducir cerca del 15,6% del tiempo de aceptación en comparación con la solución actual implementada. Esta reducción en tiempos está directamente relacionada con los costos de la empresa subastadora e indirectamente podría mejorar también el costo final del despacho.

La integración de las preferencias de los conductores en el modelo de ruteo viene acompañada de un nuevo conjunto de retos que abren una nueva familia de problemas en los servicios logísticos de "última milla". Esta nueva familia de modelos de despacho intercambia flexibilidad de la demanda con la internalización de los costos de heterogeneidad de los conductores traducidos en los precios de remate.

A Vehicle Routing Problem for Marketplaces in the Shared Economy

Pablo Azurduy (pablo.azurduy@ug.uchile.cl)
Marcel Goic (mgoic@uchile.cl)
Cristián E. Cortés (ccortes@ing.uchile.cl)
Nicolás Aramayo (nicolas@wareclouds.cl)

Abril 2022

Abstract

In recent years, the development of information technologies has led to important changes in the design of the logistics systems that companies use to distribute products to their customers. Among the most relevant changes is the emergence of digital platforms that connect firms that need to deliver their products with drivers who can address the transportation of the last mile. One of the central attributes of these platforms is the lack of contractual relationship with potential drivers and therefore when a platform offers a possible route, drivers will accept only those routes they find more attractive. This logic requires modifying traditional approaches to solve the resulting vehicle routing problem (VRP) to generate solutions that not only provide reduced transportation costs but also generate attractive routes for drivers. In this paper, we describe the vehicle routing problem with non-contractual drivers (VRPNCD) and propose a solution scheme that allows learning from drivers' preferences and incorporating them to generate solutions that meet business objectives. By applying our solution approach to a real problem of a digital logistics platform (Wareclouds), we find that we can reduce near 15.6% of the acceptance time compared to the current implemented solution. The integration of the driver preferences in the routing model comes with a new set of challenges that open a new family of problems in the last-mile logistic services. This new family of routing models exchanges demand flexibility with the internalization of driver heterogeneity costs translated into auction prices of the marketplace.

Keywords: VRP, Online-Platforms, crowdsourcing

Table of Content

1. Introduction

In recent years, the exponential growth of e-commerce has pushed for new and innovative ways of storing and delivering products. The increasing complexity to satisfy all the requirements of the operations in the last-mile has brought a large variety of innovative approaches. The introduction of new marketplace platforms for package delivery such as Ziticity[1], Amazon Flex[2], Wareclouds[3] among others have provided novel solutions for the increasing interest of consumers to get products faster and at an accessible cost. To provide an efficient operation, these platforms leverage an important feature of the shared economy: the use of the flexible capacity provided by a pool of drivers who can complete deliveries on demand. In this setting, firms could have not only more flexibility to accommodate variations in demand but also a lean cost structure that leads to superior business performance.

In this new paradigm, the digital platforms rely on marketplaces of drivers who provide a crowdsourced fleet. The role of marketplaces is to match routes with available drivers to provide a solution that allows a promptly delivery of all products under an operational scheme that is attractive to the drivers. A major difference between this assignment to the traditional centralized vehicle routing problem (hereinafter denoted as VRP) is that drivers can accept or reject the proposed routes. This condition can induce solutions that could radically differ from those generated by a central planner with contractual drivers. For instance, the variability in length of the proposed routes can become important. In a traditional VRP, the optimal solution might include some long routes in distance but with only a few delivery points. However, in a non-contractual setting, those routes could not be accepted by any driver because they might be unattractive from the drivers' perspective leading to an infeasible assignment to work in practice.

In general, we pose that these new last-mile marketplaces give origin to a new family of routing problems that introduces new sources of uncertainty and require considering different constraints that do not directly fit with the traditional VRP methodologies (Fatehi & Wagner, 2021). First, capacity is no longer fixed, and it not only depends on external factors but also on endogenous decisions of the marketplace. This capacity is determined by the willingness of drivers to accept the proposed routes. To make these decisions, drivers evaluate the attractiveness of the route considering their own preferences. These preferences could depend on many factors such as the total distance, the proximity to home location, or if the route considers more congested areas of the city.

Second, there is more uncertainty about the available capacity. In this two-sided form of operation, the main sources of uncertainty come from the imperfect knowledge about whether drivers would be willing to accept a proposed route. This aspect introduces a conceptual difference with respect to traditional VRPs that depends on the availability of transportation resources and the technical components of the system such as; the capacity of each vehicle or the maximum number of hours that drivers could work in a day. It is worth noting that driver preferences about routes can be heterogeneous and they change dynamically over time. While previous literature has proposed stochastic programming approaches to deal with variation in capacity (Noorizadegan & Chen, 2018), the capacity depends on the proposed routing solution and therefore the platform should

[1] https://ziticity.com/

[2] https://flex.amazon.com/

[3] https://www.wareclouds.com/en/

learn about the features that make a solution more attractive to drivers. Fortunately, advances in information technologies and in estimation methods allow for rich learning about agents (Goic & Olivares, 2019).

Third, the platform has no control over the actual execution of a route and while it could propose an optimal sequence of delivery points, the drivers could optimize themselves based on their own requirements. For instance, some drivers might deliver some packages in the morning and the rest a few hours later in the afternoon. As a consequence, the evaluation of the attractiveness of a given route cannot longer be solely decided based on the shortest path. Certainly, drivers assign value to the length of the routes but providing an exact TSP is no longer a first-order concern.

In this research, we address the problem of a platform who receives requests for delivery of products from multiple vendors, to offer them to a set of drivers who could either accept or reject a given request. To solve this routing problem, we propose a multi-stage framework in which the platform learns about driver preferences and uses that information to propose routes that are attractive to them while satisfying other business constraints. In this framework, we use a data-driven approach to estimate the driver preferences, and we use those preferences to build routes using a Mixed Integer Linear Optimization model. To complete the cycle, we solve a matching problem for the route driver assignment, based on historical matching acceptance-rejection data, which allows us to evaluate the expected performance of each proposed solution.

We illustrate the proposed framework to solve the operational problem of Wareclouds, a Chilean two-sided platform that provides a last-mile solution for some local ecommerce that manages the dispatch via a crowdsourced driver marketplace. We model and optimize using the actual Wareclouds dispatch instances for 25 days. We compare the solutions and estimate improvements in the routing solutions. Our estimates provide an average reduction in the acceptance time from about 15% less than the current solution. We estimate that there is a direct link between the acceptance time and the auction price of each route, therefore this time saving can be reflected also in a reduction of the auction prices for the platform. To the best of our knowledge, this is the first paper in providing a comprehensive methodology to solve the VRP problem with non-contractual drivers, which has become a mainstream logistical arrangement in recent business models. The solution to this problem requires additional considerations to deal with the imperfectly observed drivers' preferences.

The rest of the article is organized as follows. In Section 2, we revise the relevant literature. In Section 3, we present the conceptual framework whereas in Section 4 we describe de technical details we use to address each component of the framework. Section 5 is devoted to presenting the result and Section 6 elaborates on some relevant extensions and sensibility analysis. We close in Section 7 with a discussion about the results and some ideas for future research.

2. Literature Review

Our paper is related to the two main research streams. First, we have the dense literature on Vehicle Routing, where we focus on those developments that deal with routing for last-mile parcels delivery. The second stream is associated with nascent literature on routing in sharing-economy platforms.

Traditional vehicle routing problem (VRP) publications (Dessouky, Ordóñez, & Sungur, 2008) studied a capacitated vehicle routing problem (CVRP) in the context of demand uncertainty.

(Gounaris, Wiesemann, & Floudas, 2013) similarly studied a CVRP but refined the demand uncertainty estimation. (Liu, He, & Max Shen, 2021) studied a last-mile delivery service with time travel estimations and the impact of the order assignment on the network time, they use a robust optimization approach to solve this problem. The traditional VRP research has included uncertainty in diverse forms, however, the literature on shared economy uncertainties, such as the drivers' preferences or supply uncertainty has not been applied to the problem until very recently.

VRP in the context of the sharing economy has been studied in some recent publications, (Fatehi & Wagner, 2021) presents a similar problem to the one studied in this research; they model this problem using robust optimization, time windows, and queuing. Additionally, the work of (Qi, Li, Liu, & Shen, 2018) was a pioneer in the crowdsourcing last-mile problem. They study the ride-sharing industry and the adoption of the last-mile logistic firms to the crowdsourcing model, establishing many of the tradeoffs of this new business model.

In a more recent publication, (Zhen, Baldacci, Tan, Wang, & Lyu, 2022) studied a mixed delivery platform with dedicated vehicles and occasional crowdsourced drivers, which provides a modern view of the optimization problem behind the crowdsourced drivers' marketplace. Similarly, In the context of the ridesharing and parcels marketplace, another type of mixed market (Li, Krushinsky, Reijers, & Van Woensel, 2014) presented a theoretical formulation of an optimization problem of a shared network of parcels and ridesharing. Most of this literature, VRP in a shared economy context, is focused on the routing problem and its own specific constraints in a sharing-economy marketplace, our research however doesn't optimize based only on the constraints of the problem and the uncertainty of the drivers' supply, but rather to design the routes considering the drivers' preferences when grouping the parcels into routes.

There is some research related to VRP and drivers' preferences heterogeneity (Srivatsa Srinivas & Gajanand, 2016) studied the driver heterogeneity in the routing preferences and their impact on the network cost in routing in a VRP, a more recent publication (Guo, Yang, Hu, Jensen, & Chen, 2020) also studied the routing preferences heterogeneity of the drivers and use it to optimize the network routing. Both publications study the drivers' heterogeneity when they decide their own routing, however, we studied the preferences of the drivers before solving the routing and assignment.

A closer problem when routing and preferences are part of the same problem is a publication from (Karels, Veelenturf, & Van Woensel, 2020) that studied a collaborative auction mechanism between carriers in an auction model, this publication describes a similar problem but when carriers design the routes as a result of a parcel-drop auction. This mechanism is a different approach to the one that we describe in this publication when the auction is based on routes rather than parcels, however, the underlying problem is very similar.

3. Solving for the VRP with crowdsourced drivers (VRPNCD).

3.1 Problem Setting

We consider the case of a two-sided platform that receives orders from several vendors and needs to find drivers to conduct the corresponding deliveries. Our framework captures what we believe are the key components of this type of platform, but the empirical application is tailored to accommodate the business situation of Wareclouds. Wareclouds is a last-mile logistic firm that

serves small and mid-sized companies by providing them with storage and last-mile dispatch performed by non-contractual drivers. In each city they serve, the company has a series of warehouses and a list of drivers who connect to a digital marketplace to auction each route every day. Then, the platform offers routes to drivers, but as they do not have a contractual obligation with the platform, they could decide whether to accept or reject the delivery of a given route. Once a route is accepted, the driver is responsible for picking-up all products from each warehouse and delivering the goods requested to final customers. These drivers, who the company internally calls *clouders*, are not forced to deliver in any given order as far as they complete the assigned task during the day. Furthermore, they have no obligation about their availability for delivery implying that in the models we consider the drivers' supply as unknown.

Similar to other platforms, the company receives a list of requirements of products that must be dispatched the next day. They handled the list to all warehouses so they can prepare orders to be collected the next day. Before dispatching, in the early morning, the platform defines the routes to be offered to the potential drivers. As the final customers might be clustered together, there are important savings in offering drivers sets of products that must be delivered. However, several considerations should be taken into consideration when defining these routes.

- As there is no obligation for drivers to be available, the platform should have an estimation of the number of drivers who will be available and willing to accept the proposed routes.
- In practice, there is a minimum and a maximum length for the proposed routes. The minimum is justified because short routes would not be economically viable for drivers. The maximum is justified because extremely long routes would be infeasible to be dispatched within the same day.
- Unlike the traditional centralized planning where the time and distance of the routes captures all the relevant transportation costs, in this case, drivers could find that delivering to a different part of the city could be either more or less attractive. For instance, they could find some areas unsafe, or on the contrary, they could prefer to deliver products around their residences.

Once the routes are defined, the platform posts these routes to a subset of drivers. The platform decides the level of exposure of the routes to drivers based on geographical considerations and the historical compliance of previous orders. These criteria allow the platform to prioritize drivers in cases of excess supply. In these scenarios, the platform prioritizes drivers with longer tenure offering them routes that span areas in which they have often delivered products in the past.

After observing the proposed routes, the drivers decide whether to accept each route or not and upon observing acceptances, the platform can run a few additional rounds with new proposals until every route is accepted by some driver. While most of the routes are accepted in less than an hour, if some routes are not being assigned the platform can exert additional effort to convert drivers. For instance, they could engage in direct communications with selected drivers. In some exceptional cases, the platform can manually modify the proposed routes by either splitting them into two shorter routes in case they appear to be too long or merging short routes to create a longer one that could become more attractive to drivers.

The decision of drivers to accept or not a given route depends on a potentially large number of factors including the expected income, the duration of the route, the estimated time/location of the end of the proposed route, the perceived safety of the neighborhood that must be visited, the expected congestion and their own availability of time to complete the whole sequence to name a

few. It is worth noting that drivers' preferences can be highly heterogeneous. While some routes might be attractive to some drivers, the same routes can be not attractive at all to others even for the same price.

Once a driver accepts a route, s/he is not required to follow any specific order to complete the route. The platform only observes the time at which the packages are delivered and therefore the driver is entitled to use the route s/he prefers. In this regard, drivers could even split the route to do a fraction in the morning and the remaining fraction in the afternoon. The only requirement is to complete the route before the end of the day.

To empirically validate our methodology, we use the data of a complete month of operation of Wareclouds in Santiago, Chile. In this city, the company has 6 warehouses during the whole evaluation period. Table 1 reports descriptive statistics of demand for the demand in those days, and the proposed routes implemented by the company. At the time of the analysis, Wareclouds decide routes purely based on geographic segmentation and we will use it as a benchmark for our proposed solution.

	Mean	Min	Max
Number of Drops	314.9	193	503
Number of routes	19.4	15	27
Number of nodes by route	19.9	2	50
Number of pickups by route	3.8	1	7

Table 1. Descriptive Statistics of the daily instances used to calibrate the preferences and design routes.

To illustrate the spatial distribution of the problem, Figure 1 displays the location of warehouses and drops for a representative instance. Although most of the demand is concentrated in the more densely populated center of the city, there are a significant fraction of packages that must be delivered in more peripherical areas. The sparsity of the location of the drop points provides a preliminary indication of the value of grouping multiple products to be delivered in a single route.

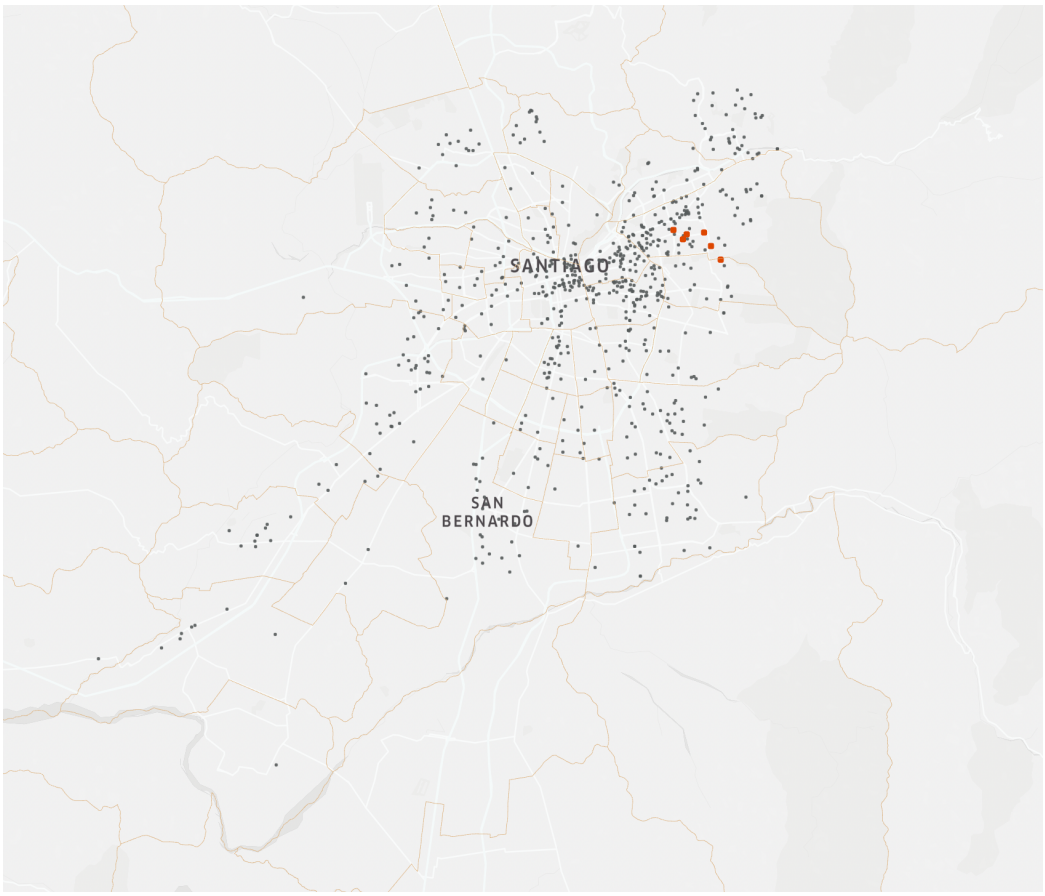


Figure 1. Map of a typical instance in the dataset, on black the drop points on red the warehouses

3.2 Conceptual Framework

Our research focuses on how to design and optimize the routes considering the preferences of the drivers. We aim to provide a workable methodology that considers the key elements of the problem. Among them, we consider learning about how drivers evaluate if a route is worth enough to be accepted, the provision of routes that are consistent with driver preferences, and the evaluation of how those routes perform in terms of business objectives. Unlike the traditional centralized planning that mostly focuses on cost reductions, our methodology should also lead to a better acceptance rate from the drivers and therefore fewer negotiation efforts from the company to achieve their service levels.

To address this problem, we propose a conceptual framework that considers its three main components. (i) a data-driven approach to learn from drivers' preferences, (ii) an optimization routine to generate routes that are consistent with drivers' preferences, and (iii) a matching model to determine the assignments of routes to drivers. These models interact with each other over time. In Figure 2 we provide a schematic representation of the framework along with the sequence in which they interact to provide a workable solution for the platform.

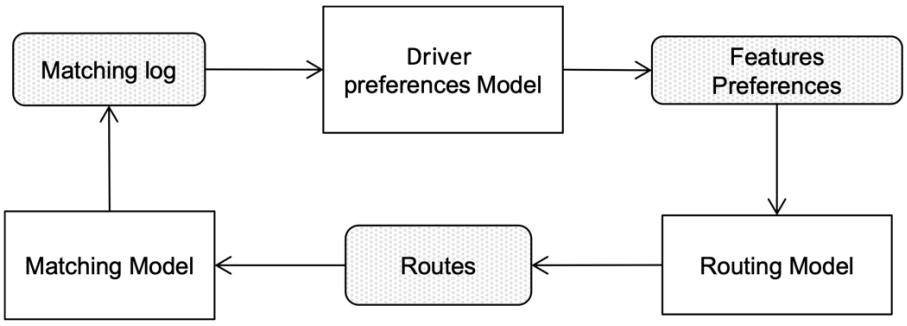


Figure 2. Conceptual Framework to Solve for the Routing Problem with non-contractual drivers.

The Driver preference model uses a matching data log that registers the historical acceptances of drivers. Using this data, we estimate a regression model to understand what are the relative weights that drivers' give to different features when deciding to accept a given route. In our case, we assume that more attractive routes are accepted earlier, and we regress acceptance time on a list of features characterizing the route. The output of this model is a vector indicating the relative importance of different features.

The vector of relative preference of features is instrumental to populate the objective function for the routing model. Using these parameters, we can estimate how desirable is each route providing a proxy for the auction cost of a particular routing solution in the marketplace. Considering the precision and solution times are important in practice, in this stage we consider both heuristic and exact methods to provide the set of optimal rules. Once optimized we have a set of routes to be matched in the next model.

The matching model takes the proposed routes and the set of active drivers' to allocate drivers' to routes. Using a probabilistic acceptance model, we can assess the likelihood that drivers' have to accept the routes, which is the final goal of the platform. This matching runs iteratively until all the routes are accepted and allow the evaluation of different mechanisms to incentivize the acceptance of less attractive routes. After the matching, we have a new log that can be used to update the training of the preference model and to use the new parameters for the next day.

This theoretical framework separates the prediction problem, "Drivers' preference Model", from both optimization problems; "Matching Model" and "Routing Model". More modern approaches, such as "predict then optimize" from (Bertsimas & Kallus, 2020) or "smart predict and optimize" (Grigas & Elmachtoub, 2022) argue that a unified approach presents high improvements in performance, depending on the misspecification of the prediction model. We used a separated approach here for the simplicity of this approach and some limitations in the matching data, but this opens an opportunity for a future extension of this framework.

The details of the models that we use to solve all the three components of the methodology are discussed next.

4. Models

4.1 Driver Preferences Model

The learning model to elicit preference parameters depends on the available data. We model the driver's preferences using actual data provided by Wareclouds. In this case, we observe a basic description of the route including the number and location of pickups and the number and location of delivery points. In general, and based on this basic information, we can derive several features, $ft(r) \in F$, that characterize drivers' preferences. For instance, we can compute the length of the route, the expected time to deliver, and the geographic dispersion of the delivery points among others. In addition, we observe whether drivers accepted the route and the elapsed time for acceptance. To estimate preferences, we assume that more attractive routes $r \in R$ are accepted earlier and we use the time to accept $t_a(r)$ as the main dependent variable of the regression analysis.

$$t_a(r) = \sum_{ft} \beta_{ft} * ft(r) \; + \; \epsilon \tag{1}$$

In these preference models, we consider several features $ft(r) \in F$ that might affect the attractiveness of a route. There are a large number of features that can influence the likelihood of a driver accepting a route. However, given the availability of data, in the empirical application, we restrict our attention to the characteristics of the route itself. Although we do not consider driver characteristics or the intensity of the interaction with the app, these and other features can be easily incorporated into the methodology in a future application.

Before presenting the complete list of features considered in our application, let us introduce some notation. Let $Y_{r,i}$ be a binary variable that takes the value 1 if the node i belongs to the route r and 0 otherwise. Next, if we consider a city tessellation G (details in the appendix Annexed A. City Tessellation (Geos/Polygons)), then $A(g)$ represents the total area of a polygon (or "geo") $g \in G$. Similarly, $I(g,r)$ is a binary variable taking the value 1 if at least one drop of the route r belongs to the polygon g. $W = \{w_1..w_{nW}\}$ it's the set of all warehouses. $D = \{d_1..d_{nD}\}$ the set of drops. Finally, dr_i is the linear distance between a drop i and its corresponding warehouse. With these definitions, the full list of features is displayed in Table 2.

Feature	Description	Formal Definition
Num. of pickups	Number of pickups (warehouses) along the route r	$np(r) = \sum_{i \in W} Y_{r,i}$
Num. of drops	Number of drops nodes along the route r	$nd(r) = \sum_{i \in D} Y_{r,i}$
Dist. all to depot	The sum of all distances dr_i (km) from each drop i to their warehouse	$dr(r) = \sum_{i} dr_i * Y_{r,i}$
Cover area	The sum of all cover area $A(g)$ due to a given city tessellation $g \in G$	$a(r) = \sum_{g} A(g) * I(g,r)$

Inter geo distance	The sum of all distances $d(g_1, g_2)$ (km) between the centroids of all polygons where the route has a node Z_{k,g_1,g_2}. This is a route concentration/dispersion measure	$dg(r) = \sum_{g1,g2} Z_{k,g_1,g_2} * d(g_1, g_2)$
Includes polygon g	Binary indicator if the route has a node in the polygon $g \in G$	$I(g,r)$
Number of polygons	Total number of polygons where the route has one pickup or drop	$ng(r) = \sum_{g} I(g,r)$
Route length	Total length in km of a route based on a linear approximation described on 0	$total\ distance(r)$

Table 2. Features Driver Preferences Model

Using these features, we fit a linear model using the acceptance time $t_a(r)$, in minutes as the dependent variable. From the results of this regression model, we get an estimate of the preference parameters vector β_{ft} representing the relative importance of each component in driving acceptance of routes. These estimates will be used in the routing model to create more routes that are more likely to be accepted in a short time.

It is worth noting that, except for the total length of the route, all the remaining features presented in Table 1 can be estimated without solving the TSP explicitly, and therefore, to impute the attractiveness of a given route there is no need to solve the exact VRP. To define routes, we can search for partitions of the list of drops that maximize their attractiveness based on driver preferences.

Not having to solve the exact VRP brings large reductions in computational complexity allowing us to solve real-sized problems with hundreds of drops in a matter of minutes. However, an important downside of not solving the TSP of each route is that we do not have access to the optimal order of the nodes nor the length of the route. We believe the length of the route is a relevant feature that drivers' use to decide to accept a proposed route. Nevertheless, literature on subjective evaluation of distances suggests that drivers do not necessarily evaluate on precise estimates of distances but on approximated constructs (Montello, 1997; Li, Kang, and Ba, 2020). Thus, to include the evaluation that drivers might have about distances without incurring in the prohibitive computational cost of the exact VRP, in this research we use an approximation based on Daganzo, (2005). This approximation is explained in the following sub-section and allows us to estimate the length of each possible route without solving the TSP.

4.2 Daganzo TSP approximation

To estimate the total distance of a route without solving the TSP, we adapt the approach proposed by Daganzo (2005) who addresses a similar approximation problem. In his work, Daganzo demonstrates that the length of a route *r* can be approximated using some aggregated statistics of the problem. The approximation equation is presented in Equation (2).

$$Total\ Distance\ \approx \frac{2E(r)}{C} * N + k\sqrt{|R|N} \qquad (2)$$

Here, *C* represents the total number of drops (or stops) and *N* the total number of customers on the route. In our application, each drop is associated with a single customer and therefore we assume that *C*=*N*. In this equation $E(r)$ represents the expected value of all the distances from the drops to the warehouse and, $|R|$ represents the area covered by the route and k is a constant that varies in the range of (0.82, 0.57) depending on the type of distance used (e.g. L1, Euclidian).

To implement this idea in our setting, we use a regression model to find the relative weights of the distances and areas to properly represent the actual length of a route. The regression equation is presented in Equation (3) and is calibrated using the distances derived from the Christofides' Algorithm (1976)[4] to solve the underlying TSP's.

$$Total\ Distance\ (\mathrm{r}) \approx \gamma_1 * \sum_{\mathrm{i} \in \mathrm{D} \cup \mathrm{W}} dr_i * Y_{i,r} + \gamma_2 * \sum_{\mathrm{g} \in \mathrm{G}} A(g) * I(g,r) \qquad (3)$$

The first term includes the linear distances dr_i of every drop in the route to its corresponding warehouse. As they are multiplied by the dummies $Y_{i,r}$, the summation only includes the drops associated with that route. The second term is associated with the geographical dispersion of the drops and includes the areas $A(g)$ of all polygons considered in the route. For instance, if all drops are located in the same polygon, the binary indicators $I(g,r)$ guarantee that only the area of that polygon is included. Under this specification, the calibration of the parameters γ_1 and γ_2 provides the best linear estimate of the length of the route. Results of this approximation are available in Appendix 9.2 and they confirm that this approach leads to a good approximation of the total length implied by the resolution of the TSP.

4.3 Routing Model

Following the previous discussion, our decision task consists of the definition of subsets of drops that define a route that can be offered to drivers. In principle, this problem could be viewed as a clustering problem where several locations should be grouped. However, traditional clustering techniques ignore some important features of our problem. For instance, to be operationally feasible the routes should consider a minimum and a maximum length. In addition, we are not only interested in grouping drops that are close geographically, but also in creating routes that are attractive to drivers. Thus, to address this problem we use a mixed-integer programming model as explained below.

To introduce some notation, let us define a set of features F, a set of routes R (clusters), a city tessellation[5] $g \in G$, a set of warehouses $W = \{w_1 .. w_{nW}\}$ for picking up the products, a set of drops $D = \{d_1 \dots d_{nD}\}$ where the demand must be satisfied, a set $DW = \{(d_i, w_j), \dots\}$ that contains all corresponding pairs (d_i, w_j) where w_j is the warehouse that stored the goods for the drop d_i. To simplify notation, we will say that the node i belongs to the polygon $g \in G$ $(i \in g)$ if the node i is located inside the polygon $g \in G$. Then, we define the following decision variables:

$$Y_{r,i} = (\mathrm{bin})\ 1 \text{ if node } i \text{ to route } r, 0 \text{ otherwise}$$
$$Z_{r,g_1,g_2} = (bin)\ 1 \text{ if polygon } g_1 \text{ and polygon } g_2 \text{ had nodes belonging to route } r$$

[4] In our computational Christofides' implementation, we used Networkx (Schult, Hagberg, & Swart, 2008).
[5] For details about city tessellation review appendix 0

$I(g,r) = (bin)1$ if at least one node in route r belongs to polygon g
$Y\_sd\_cod_r = (bin)\ 1\ if\ the\ route\ has\ at\ least\ one\ node\ 0\ otherwise$
$nd(r) = (\mathbb{R}_0^+)$ number of drops in the route r
$np(r) = (\mathbb{R}_0^+)$ number of pickups in the route r
$ng(r) = (\mathbb{R}_0^+)$ number of polygons in the route r
$dg(r) = (\mathbb{R}_0^+)$ sum of all distances between polygons in the route r
$dr(r) = (\mathbb{R}_0^+)$ sum of all distance (km) from each drop to the warehouse
$a(r) = (\mathbb{R}_0^+)$ sum of all covering area (all polygons covered area)

Thus, the optimization problem can be expressed as.

$$\min \sum_{r \in R, \forall ft \in F} \beta_{ft} * ft(r) \qquad (4.1)$$

Features codification

$$nd(r) = \sum_{i \in D} Y_{r,i} \quad \forall r \in R \qquad \text{(cod ft\_size\_drops)} \qquad (4.2)$$

$$np(r) = \sum_{i \in P} Y_{r,i} \quad \forall r \in R \qquad \text{(cod ft\_size\_pickups)} \qquad (4.3)$$

$$M * I(g,r) \geq \sum_{i \in D : i \in g} Y_{r,i} \quad \forall r \in R \qquad \text{(cod ft\_has\_geo\_min)} \qquad (4.4)$$

$$I(g,r) \leq \sum_{i \in D : i \in g} Y_{r,i} \quad \forall r \in R \qquad \text{(cod ft\_has\_geo\_max)} \qquad (4.5)$$

$$ng(r) = \sum_{g \in G} I(g,r) \quad \forall r \in R \qquad \text{(cod ft\_size\_geo)} \qquad (4.6)$$

$$a(r) = \sum_{g \in G} I(g,r) * A(g) \quad \forall r \in R \qquad \text{(cod ft\_cover\_area)} \qquad (4.7)$$

$$dr(r) = \sum_{i \in D} Y_{r,i} * dr_i \quad \forall r \in R \qquad \text{(cod ft\_cover\_area)} \qquad (4.8)$$

$Z_{r,g1,g2} \leq I(g_1,r) \quad \forall r \in R, \forall g1 \in G$ (cod intergeo g1) (4.9)

$Z_{r,g1,g2} \leq I(g_2,r) \quad \forall r \in R, \forall g2 \in G$ (cod intergeo g2) (4.10)

$Z_{r,g1,g2} \geq I(g_1,r) + I(g_2,r) - 1 \quad \forall r \in R$ (cod intergeo max) (4.11)

$nd(r) \leq M * Y\_sd\_cod_r * min\_size\_routes \ \forall r \in R$ (size drops min cod) (4.12)

Model Constraints

$$\sum_{r \in R} Y_{r,i} = 1 \quad \forall i \in D \qquad \text{(demand fulfillment)} \qquad (4.13)$$

$$Y_{r,i} \leq Y_{r,j} \quad \forall i \in D, \forall j \in W: (i,j) \in DW \qquad \text{(drop} - \text{warehouse pairs)} \qquad (4.14)$$

$$\sum_{r \in R} Y_{r,j} = 0 \quad \forall j \in W: (i,j) \notin DW \quad \forall i \in D \qquad \text{(unused warehouses)} \qquad (4.15)$$

$$nd(r) \geq Y\_sd\_cod_r * min\_size\_routes \quad \forall r \in R \qquad \text{(size min drops)} \qquad (4.16)$$

$$nd(r) \leq max\_size\_routes \quad \forall k \in R \qquad \text{(size max drops)} \qquad (4.17)$$

The first set of constraints (4.2 - 4.12) in this model corresponds to linear feature codifications. For instance, (4.2) defines the number of drops and (4.3) the number of pickups respectively as a function of the decision variable $Y_{r,i}$.

The second group of constraints (4.13 – 4.17) corresponds to business constraints, (4.13 – 4.15) configure basic solution codifications. (4.16, 4.17) add a minimum and a maximum number of drops per route. Certainly, the model enables us to include a variety of operational constraints, but in this empirical analysis we focus on these two for the following reasons:

- The drivers' preference model is built using the routes that have been implemented by the platform. By restricting the decision space to routes that are similar to those offered historically, we reduce the potential forecasting errors.

- In practice, extremely short routes are not attractive to drivers because stopping by warehouses involves a fixed cost associated not only with the physical movement of products but also the coordination with the personnel at the depots. Similarly, extremely long routes are not accepted by drivers because they cannot be completed in a single day as is requested by the platform.

In theory, these considerations could be controlled in the objective function such that short and long routes are not selected because they are not attractive. We prefer this specification because it leads to better computational performance.

To solve this model, we use an exact approach via a MIP solver. In addition, we develop an ad-hoc heuristic to provide a faster solution that might be attractive for practical implementations. As we use this heuristic as a warm start for the solver, it also leads to faster and better-quality solutions in the exact approach. To complete the analysis, we compare the solution of these two approaches against those implemented by the company.

4.3.1 A Heuristic Approach

To solve the optimization problem described above, we considered a heuristic procedure that provides a fast approximation to the problem. The heuristic proceeds as follows:

1. We initialize the heuristic by generating feasible solutions through the division of the set of drops of a given instance in clusters/routes with sizes between min_size_route and

max_size_route. To find these clusters/routes, we use the constrained-k-means implementation of Levy-Kramer & Klaber (2021) based on the methodology described in Bradley (2000) this model finds clusters with sizes within certain bounds. We iterate over all possible values of k (number of clusters/routes) keeping the sizes of the clusters constrained with the same bounds, we finally keep the solution with the lowest cost.

2. For each cluster/route, we evaluate local changes by evaluating how they impact the total cost of each route. While the impact of some changes can be computed directly, others depend on the composition of the whole route and therefore we need to impute them. Using these cost estimates, we implement two sub-routines, a **transfer routine,** and **a swap routine.**

3. The **transfer routine** identifies costly drops from the longest routes and evaluates if transferring that node to the closer routes with available space would improve the solution. To illustrate the logic of the transfer routine, Figure 3 shows two routes, r_1(in gray) and r_2(in white). The numbers in each node represent its cost in the route. The figure illustrates a potential transfer of the last node of the white route (r_1), with cost 3, to the gray route (r_2). If the transfer induces a total cost reduction (of the new routing), we kept that node in the new route. Otherwise, we transfer it back. We iterate over all drops of the solution from the longest route to the shortest and then from the more expensive to the cheapest drop.

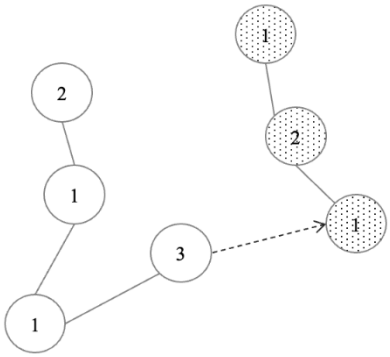


Figure 3. Illustration of the transfer routine.

4. The **swap routine** will trade the costly nodes from a route r_2 with the closest node to another route r_1 and verify if that swap of drops contributes to the cost reduction of the overall solution. We illustrate the swap routine in Figure 4 where we evaluate the potential interchange of a node with a cost of 2 in route r_2 (gray) to another with cost 1 in route r_1 (white). If the cost of this new solution is lower than the previous solution, we consolidate the trade. Otherwise, we do not and continue searching for alternative swaps.

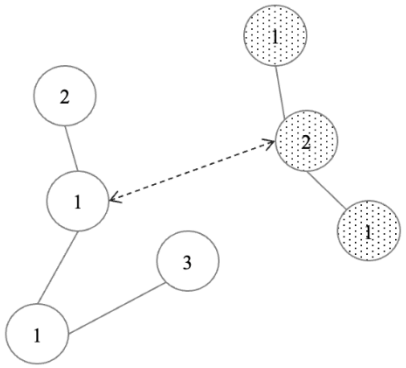


Figure 4. Illustration of the swap routine.

5. We execute the transfer and the swap routine in all routes once. After we complete both routines, we exit the heuristic having a feasible routing instance

4.4 The matching problem

Beyond the value of the objective function, to assess the operational performance of the solution, we consider a matching problem that allows us to simulate how drivers would accept the proposed routes. In this matching problem, we first compute the likelihood of each driver $c \in \Omega$ of accepting a certain route $r \in R$. We model this problem using features from both, the routes $\{ft(r)\}$, and the drivers $\{ft\_clouder(c)\}$. In addition, we consider a price p than can be adjusted to clear the market. We label this probability of acceptance as $Q(r, c, p)$

$$Q(r, c, p) = \mathbb{P}(c\ accept\ route\ r \mid \{ft(r)\}, \{ft\_clouder(c)\}, p) \quad (5)$$

To fit that model, we need historical data from past matchings, such as the offer price, and some Clouders features. Once we have that model fitted, using a classification model such as XGBoost, we can estimate the price of a match. This price, between a clouder and a route $p^*(c, r|\alpha)$, is defined as the minimum price where the probability of acceptance is equal to or higher than a certain threshold α. Given all prices of all pairs of clouders and routes, we can solve the min weight matching problem which provides us the assignments of routes to drivers. Let $X_{c,r}$ be a binary variable that takes the value 1 if the route r is assigned to the driver c. Then, this assignment can be determined by solving the following min-cost matching problem.

$$\min \sum X_{c,r} * p^*(c, r|\alpha)$$

$$st: \sum_{c \in \Omega} X_{c,r} = 1\ \forall r \in routes, \qquad X_{c,r} \in \{0,1\}$$

$$p^*(c, r|\alpha) = argmin_p \{Q(r, c, p)\ |Q(r, c, p) \geq \alpha\}$$

This problem can be easily solved by a bipartite m x n matching algorithm (Karp, 1980). To solve it computationally, we use the NetworkX Implementation (Schult, Hagberg, & Swart, 2008). In our empirical application, we do not observe individual acceptances and therefore we perform the matching using simulated data that mimics aggregated acceptance behavior. The details of these simulations are explained in **0 Annexed C. Estimating the** cost by drop

To implement the heuristic, we need to estimate the impact of having a node on the total cost of a particular route. These costs allow us to prioritize more costly nodes to be swapped or transferred to other routes in a greedy approach.

We base the estimation of this cost on the objective function formula, which estimates the cost of each route given certain features that describe that route (e.g, number of drops in the route, number of warehouses in the route). We group these features into two groups: **non-separable features** and **separable features.** A separable feature is a feature that can directly separate the contribution of each drop to the final cost. For example, for the feature that sums all distances from the drops to their warehouse $dr(r) = \sum_i dr_i * Y_{r,i}$, we can simply assume that the contribution of drop i is the distance from i to its warehouse dr_i. Similarly, for the number of drops $nd(r)$, each drop contributes one unit to the total.

A non-separable feature is a feature for which the contribution of each drop depends on other drops in the route. Consider for example the case of dummies $I(g, r)$, that account for the presence of the route on a drop in a given polygon *g*. Even if the node $i: geo(i) = g$ is removed, the corresponding cost could still be accrued by the route because there might be another node in the same polygon. To deal with this type of feature, we divide the cost of this feature between the drops that participate in that feature. In the example of the feature $I(g, r)$, we simply divide the cost by all the nodes in that feature. We operate with the rest of the features using the same approach.

Finally, we approximate the cost of each drop i (in a particular route r) by:

$$cost_r(i) = \sum_{ft\ is\ separable} \beta_{ft} * ft(i) + \sum_{ft\ non-separable} \frac{\beta_{ft}}{|\{i \in r: i\ contrib\ to\ ft\}|}$$

That formula provides us with an approximation that can be used in the heuristic for prioritization purposes.

5. Results

To present the results we start with the preferences model and then we show a detailed characterization of the routes derived from the different solution methods. We conclude with an evaluation of the performance of the proposed routes in terms of business performance.

5.1 Results from the Driver Preferences Model

Before estimating the model, we discarded a few outliers with disproportionally long acceptance times. These cases typically correspond to some packages that must be delivered to suburban areas and therefore require some spatial treatment. In fact, those cases are typically consolidated and delivered separately with a different compensation scheme for the drivers. The results of the Driver Preferences Model, using a standard linear regression and 515 observations, had an adjusted R-Squared of 0.148, and an AIC of 4326. The resulting coefficients are described in Table 3. We keep only some geo features, the ones that were more significant in a reduced model, otherwise, that will add 52 more variables.

	coef	P>|t|
$I(g = 3, r)$	0.1740	0.952
$I(g = 33, r)$	24.1430	0.134
$I(g = 4, r)$	3.5513	0.257
$I(g = 41, r)$	-17.8391	0.279
Inter geo distance	-0.0384	0.054
Num. of drops	-0.0573	0.620
Number of polygons	0.7512	0.379
Num. of pickups	0.7547	0.152
Route length	0.1253	0.008

Table 3. Regression results for drivers' preferences

According to the results of Table 3, only a few features are significantly different from zero. The smallest p-value is associated with the approximated length of the route, followed by the spatial concentration of the points captured by the distance between geos (polygons). Although we expect that with a longer history of acceptances more details about drivers' preferences could be revealed, these results provide preliminary evidence that previous acceptances can be informative about which routes might be more attractive. Furthermore, this is a useful exercise to illustrate how these preferences can be incorporated into a vehicle routing problem.

In terms of the direction of the effects, the large majority of the coefficients had the expected sign. For instance, the number of different geos (polygons) in the route and the approximated length according to Daganzo's approximations have all positive signs, implying that longest routes (in distance) are less preferred and therefore have longer acceptance times. Similarly, routes with more pickups are less likely to be accepted. While there is a monetary compensation for each additional pickup, this positive sign implies that drivers internalize that there is value in aggregating routes with only a few pickup points.

There are other features with a less intuitive interpretation that require further discussion. This is for example the case of the number of drops, where we find that "larger" routes (number of drops) are associated with lower acceptance times. We believe that an important reason why is that the payment formula is directly related to the number of drops, therefore, a larger route (in drops) means a higher price route. The negative sign for the distance between geos (polygons) considered in the route reinforces this observation, but this will not be intuitive when we also consider that this probably increases the route length (in distance). There is a strong correlation between those three variables (number of drops, inter geo distance, and route length), these effects are not easy to separate in a standard linear regression model therefore it's expected that some coefficients are not significant when we fit in a small number of samples without controlling for the price (for example).

5.2 Result of the VRP Problem

We have a set of 29 real instances of the problem. Each instance represents a day of operation of the platform in Santiago. Along with the location of the drop points and the corresponding pickup,

points for each instance we observe the manual solution implemented by the platform and the acceptance time of each of those routes. We run both, the heuristic and the optimization model over all the instances. As indicated in the description of the methodology we use the heuristic solution as the warm start for the optimization routine. The heuristic constrained k-means is solved using COIN-OR (taking on average 5 minutes), but the MILP of the full model is solved using Gurobi. In the numerical results, we impose a maximum execution time of 5:30 hours obtaining an average optimality gap of 4.78%.

In Figure 4 we display a comparison of the value of the objective function for the three solution approaches. In the right panel, we exhibit the detailed values for all the 29 instances we used in the analysis, whereas the left panel shows a boxplot summarizing those results. According to these figures, the heuristic solution results is, on average, *+3.66%* worse than the manual solution, but it outperforms it in *25%* of the scenarios. The MILP model solution results in a *-15.59%* improvement in the total cost in comparison with the manual solution currently implemented by the company.

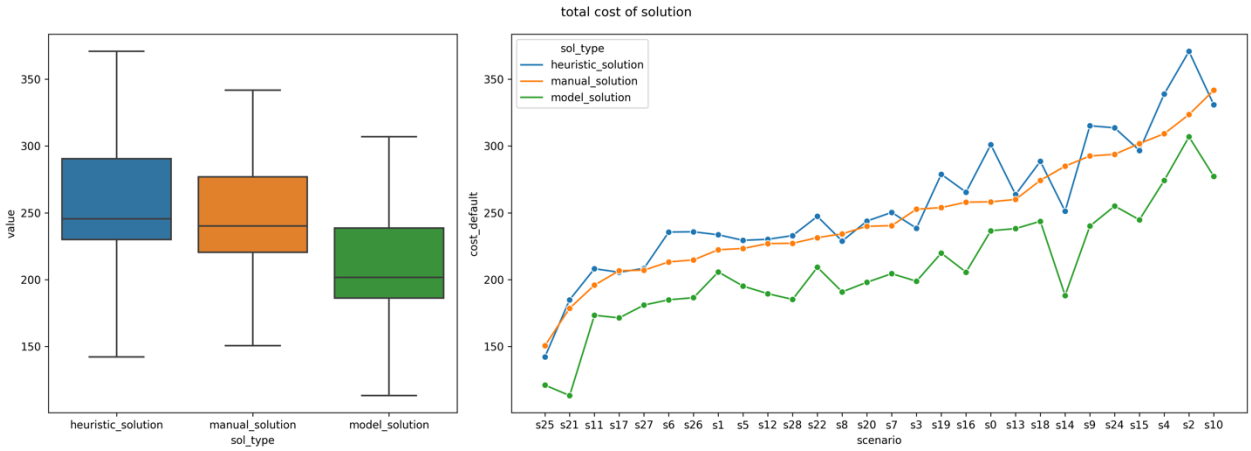


Figure 4. Cost of solutions by type of solution. Sorted by manual solution cost.

Overall, these results indicate that while the MILP consistently leads the best solution, the heuristic can provide quick results with relatively good performance. For these instances, the manual solution implemented by the company presents competitive performance. However, we would be cautious of the generalizability of this result. In fact, the manual solution is mostly based on aggregation at the county level which works well in this case because in our preference model we only identify weak effects for the majority of the proposed features. We expect that with a more sophisticated vector of preferences, the manual solution could more dramatically deviate from the optimal solution. As we discuss in the next subsection, the manual solution leads to route profiles that are significantly different from those implied by the optimal model.

5.3 Characterization of Proposed Routes

Here we study how the different methods lead to different route configurations and we compare them against those resulting from the manual solutions. We first compare how many routes are used by each model to cover the demand of each instance. Results are presented in Figure 5.

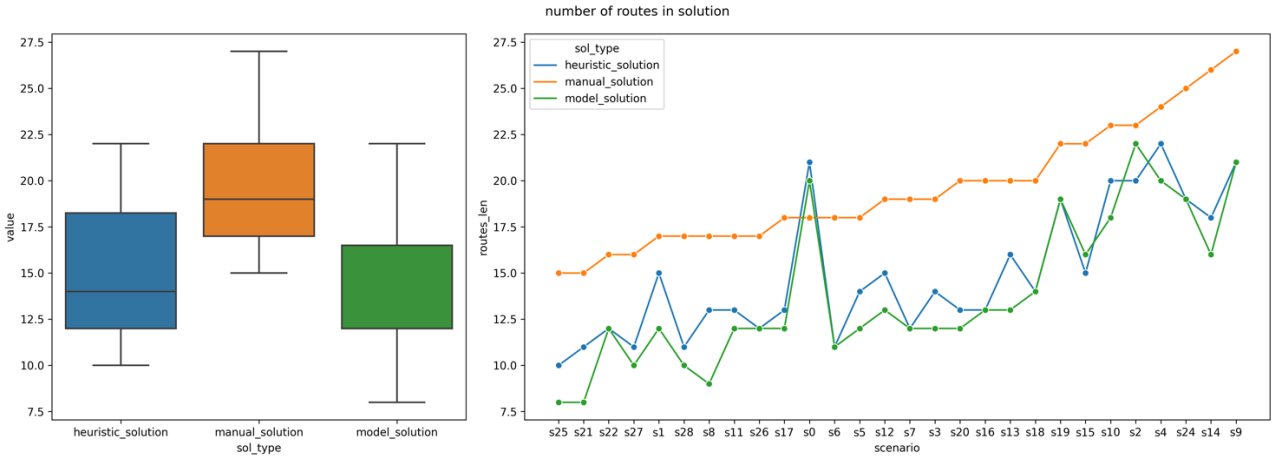


Figure 5. Number of Routes by type of solution

Results from Figure 5 indicate that the number of routes used by the full model and the heuristic is consistently smaller than those demanded by the manual solutions generated by the company. We believe this is because the model has more flexibility to search for spatial synergies to define routes, in comparison to the more static approach used by the firm that creates routes based on aggregations at the geo (polygon) level.

The number of routes is also useful to discuss an important difference between traditional VRPs and the operations in a shared economy. While in centralized planning, there is an important fixed cost of labor that motivates the platform to have a relatively constant number of routes, in the sharing economy there are more degrees of freedom to use an uneven number of drivers on different days. It is worth noting that, although the company mostly decides routes based on predefined regions, they can consider additional business considerations that we do not include in our model. For instance, they could have a sense bout the number of drivers that are not being assigned to any route, and based on that information they could decide to split the demand among a larger number of drivers to keep them engaged.

We now compare solutions in terms of the length of the routes, which we summarize in Figure 6. According to these results, the average number of drops per instance route is consistently larger for the heuristic and the MILP models. The MILP model is, on average, +10 drops larger, whereas the heuristic is on average +6 drops longer than the manual solution. This is expected because solutions with more routes are naturally associated with a smaller number of drops per route. Notice that the heuristic and the MILP models are already constrained to select routes in a range of values for the number of drops, which implicitly restricts the number of drops that can be assigned to a route. It is also interesting to note that, the optimization routines generate longer routes (in distance and nodes) despite having a positive coefficient for the driver preferences for longer routes (in distance) but a negative in longer routes (in nodes), this probably means that in the objective function the size of the route in drops weights higher than the distance length of the routes.

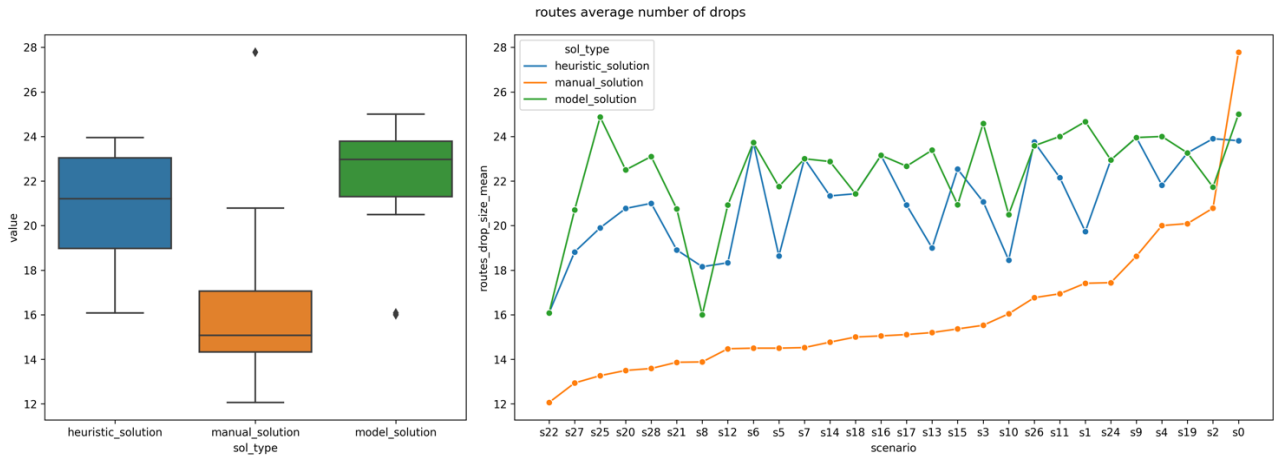


Figure 6. Average number of drops by route.

A third metric that we are interested in evaluating is the number of pickups by route. They are important in terms of both, the operational efficiency of the process and the attractiveness of the routes. Although most routes have only a few pickups, they are typically associated with longer waiting times and they require additional coordination with other agents.

Unlike the number of drops, here we observe that the heuristic solution departs from the MILP and consistently generates routes with more pickups. This result indicates that the heuristic could be improved by creating specialized refinements focusing on pickups. In fact, our implementation of transfers and swaps routines mostly focuses on the number of drops, and the number of pickups is only considered indirectly when evaluating the cost function. The difference with the solution implemented by the firm is also expected because they do explicitly consider this feature when manually designing routes.

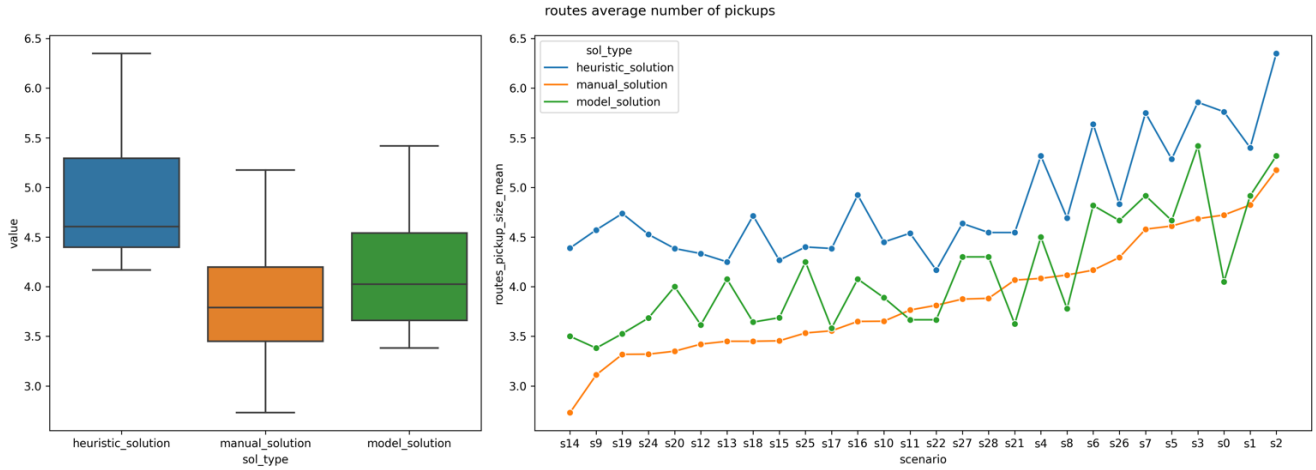


Figure 7: Average number of pickups by type of solution.

We complete this comparison with the total distance of the route. To compute this metric, we use the linear TSP approximation (Daganzo). Results are displayed in Figure 8. As the total distance is very closely related to the number of drops, the results presented here are similar to those reported in Figure 6. For instance, compared to the other two, the solution implemented by the firm have

consistently shorter distances. Interestingly, the distances that the heuristic generates routes that are only marginally longer than the current solutions.

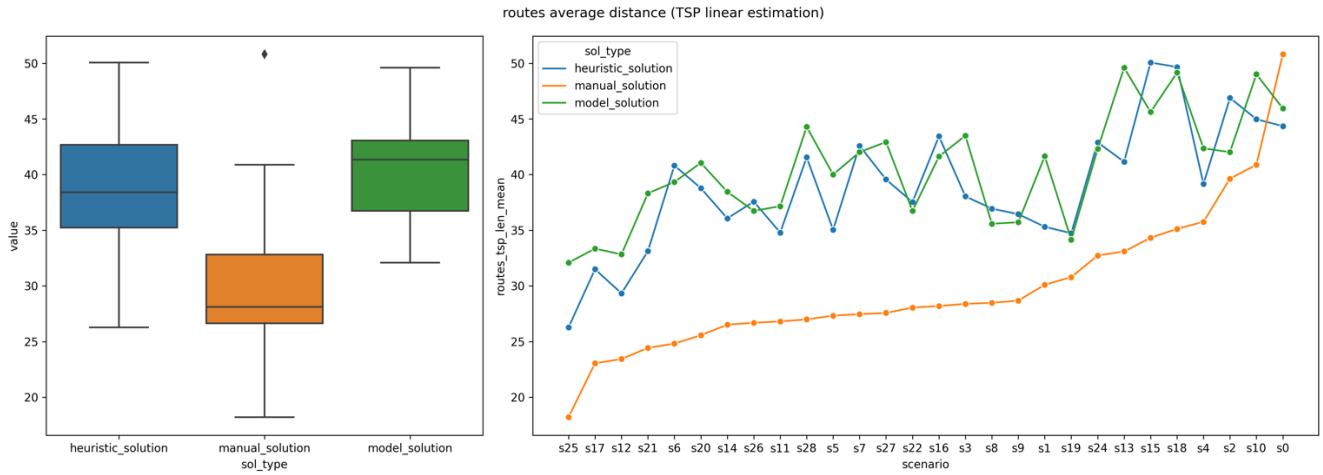


Figure 8: Routes average distance using TSP linear estimation.

To summarize, compared to the current solutions implemented by the firm, the proposed solutions generate a smaller number of routes that consider more drops and pickups. In the following subsection, we evaluate to which extent these new routes are associated with better acceptance times.

5.4 The gain of accounting for driver preferences.

To complete the analysis, we evaluate the impact of the proposed solution on business performance. A basic premise of our approach is that the speed at which drivers choose to accept to serve a route can be mapped into a finite number of features. In this section, we evaluate the impact on the performance of the solution by not properly accounting for the relationship between the features of the solution and the rate of acceptance.

As we pointed out in section 5.2 the MILP model leads to an average improvement in the cost function of *-15.59%* with respect to the manual solution. This translates into faster acceptances of nearly 50 min of saving on total by instance. Both routing methodologies have a high variation in acceptance times. In general, given that the objective function minimizes the sum, the reduction in the number of routes is the main reason behind these savings rather than a reduction on cost-per-route.

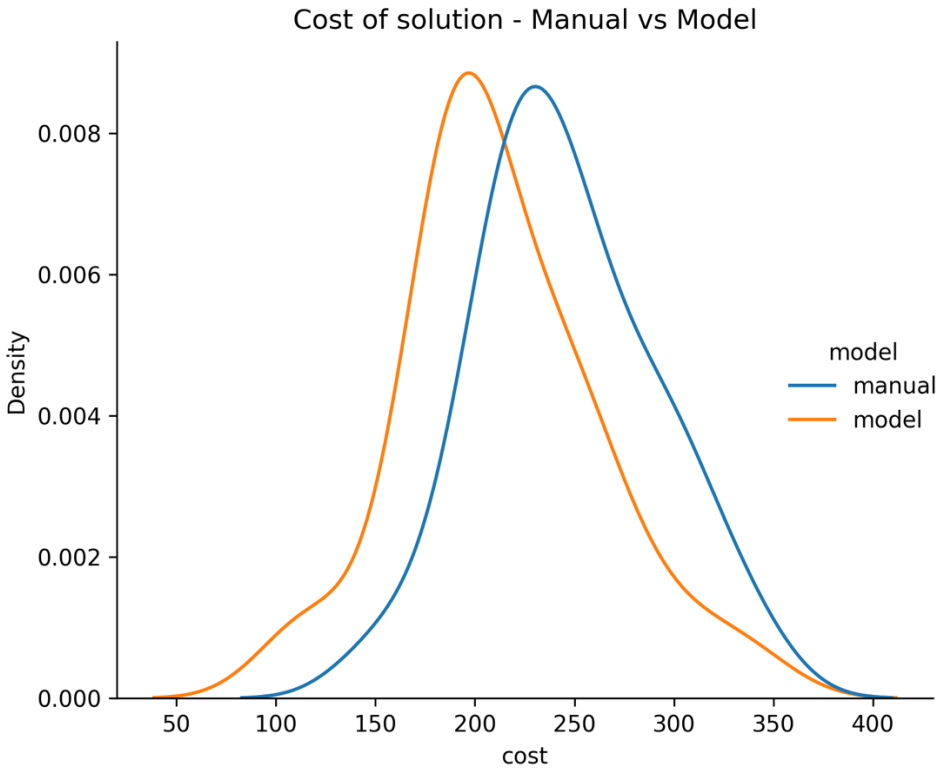


Figure 9: Objective function savings with the optimization model.

To gain perspective on the importance of these changes, we estimate how much the company would need to pay for those routes to achieve a similar reduction in acceptance time. Here we find that a decrease of 15% translates into a decrease in the auction price of a similar magnitude.

6. Discussion and Future Research

This investigation has a few major improvement opportunities that were not able to solve due to the technical capabilities of the data collection in the current marketplace, both are related and very important for future research and to make the results of this research more robust.

This dataset is a "partial matching data log" because we don't have access to what routes were offered to what clouders, what routes were accepted what routes were declined, how much time the offer was active until re-send, what were the contact medium, how the route was priced, how many routes were sent to a particular driver, what is the driver tenure, etc. The available data is way more aggregated and with several unknown variables.

In the driver preferences model, we couldn't archive good accuracy or reliability of the model, the reasons for this are well explained in section 5.1 but we attribute this to two major limitations, unobservable variables (such as price), and the lack of operative consistency during the data collection period. Both factors limit a lot of the possibilities to build a stronger model here and probably in future research this data should be re-collected using stronger experimental conditions and a wider number of variables.

The link between the preferences vector and the auction cost is not easy to argue, our model is based on the acceptance times of the marketplace, and we assume that a more desirable route (with a lower acceptance time) will be translated into a lower auction cost of a route, but we know that this auction cost is a condition by other external effects such as marketplace conditions, the matching algorithm, supply conditions, traffic conditions, payment formula, auction mechanics, etc. Again, to study this link we should have the "full matching data log". With this information, the study of that link would provide insightful information about the matching and marketplace conditions.

On the Routing model, we prioritize the overall cost of the solution rather than the cost-per-route minimization, this led to fewer and longer routes, which has direct effects on the supply side of the marketplace; the heterogeneity of the drivers might lead to losing drivers that are more interested in shorter routes, or providing fewer routes will reduce the overall interest to connect to the marketplace. This might be solved, in future research, by studying different objective functions, taking into consideration some heterogeneity variables in the expected auction cost.

Given that the routing model was built based on the preferences vector a natural extension will be to study the reliability of the solutions based on a robust optimization, considering multiples values of the preferences vector or their distributions, which also opens the opportunity to study further the heterogeneity of the drivers and his effect on the routing solutions.

In this first study, we integrate the heterogeneity of the drivers' preferences in a routing model. This problem has become more interesting with the irruption of shared-economy marketplaces to the last-mile delivery business. This is quite novel, even in mature shared economy marketplaces (Jin, 2021) (Lyft, 2018) (Uber, 2021) where the driver preferences are barely adopted. This is more relevant when the routing design is a previous step to the auction, given that the final cost will rely heavily on the driver's preferences. Building more heterogeneity-aware models will lead to a new family of problems and might become an important piece of the future of last-mile delivery logistics.

7. Bibliography

Bertsimas, D., & Kallus, N. (2020). From Predictive to Prescriptive Analytics. *Management Science 66*, 1025–1044.

Bradley, P. S. (2000). Constrained k-means clustering. *Microsoft Research, Redmond*, 0.

Christofides, N. (1976). Worst-case analysis of a new heuristic for the travelling salesman problem. *Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group*.

Daganzo, C. (2005). Chapter 4 One-to-Many Distribution. In C. F. Daganzo, *Logistics Systems Analysis* (pp. 93–102). Springer Publishing.

Dessouky, M., Ordóñez, F., & Sungur, I. (2008). A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 509-523.

Fatehi, S., & Wagner, M. (2021). Crowdsourcing Last-Mile Deliveries. *Manufacturing & Service Operations Management*.

Goic, M., & Olivares, M. (2019). Omnichannel Analitycs. In S. Gallino, & A. Moreno, *Operations in an Omnichannel World* (pp. 115-150). Cham: Springer.

Gounaris, C., Wiesemann, W., & Floudas, C. (2013). The Robust Capacitated Vehicle Routing Problem Under Demand Uncertainty. *Operations Research*, 677-693.

Grigas, P., & Elmachtoub, A. (2022). Smart "Predict, then Optimize". *Management Science*, 9-26.

Guo, C., Yang, B., Hu, J., Jensen, C., & Chen, L. (2020). Context-aware, preference-based vehicle routing. *The VLDB Journal*, 1149-1170.

Jin, H. (2021, 11 10). *Next-Generation Optimization for Dasher Dispatch at DoorDash*. Retrieved from DoorDash Engineering Blog: https://doordash.engineering/2020/02/28/next-generation-optimization-for-dasher-dispatch-at-doordash/

Karels, V., Veelenturf, L., & Van Woensel, T. (2020). An auction for collaborative vehicle routing: Models and algorithms. *EURO Journal on Transportation and Logistics*, 100009.

Karp, R. (1980). An algorithm to solve them ×n assignment problem in expected timeO(mn logn). *Networks*, 143-152.

Levy-Kramer, J., & Klaber, M. (2021, August 24). *k-means-constrained*. Retrieved from K-means constrained clustering implementation: https://github.com/joshlk/k-means-constrained

Li, B., Krushinsky, D., Reijers, H., & Van Woensel, T. (2014). The Share-a-Ride Problem: People and parcels sharing taxis. *European Journal of Operational Research*, 31-40.

Liu, S., He, L., & Max Shen, Z. (2021). On-Time Last-Mile Delivery: Order Assignment with Travel-Time Predictors. *Management Science*, 4095-4119.

Lyft. (2018, 06 10). *Matchmaking in Lyft Line — Part 3 - Lyft Engineering*. Retrieved from Medium: https://eng.lyft.com/matchmaking-in-lyft-line-part-3-d8f9497c0e51

Noorizadegan, M., & Chen, B. (2018). Vehicle routing with probabilistic capacity constraints. *European Journal of Operations Research*, 544-555.

Qi, W., Li, L., Liu, S., & Shen, Z. (2018). Shared Mobility for Last-Mile Delivery: Design, Operational Prescriptions, and Environmental Impact. *Manufacturing & Service Operations Management*, 737-751.

Schult, D., Hagberg, A., & Swart, P. (2008). Exploring Network Structure, Dynamics, and Function using NetworkX. *Proceedings of the 7th Python in Science conference (SciPy 2008)*, 11-15.

Srivatsa Srinivas, S., & Gajanand, M. (2016). Vehicle routing problem and driver behaviour: a review and framework for analysis. *Transport Reviews*, 590-611.
Uber. (2021, 05 03). *Freight Pricing with a Controlled Markov Decision Process*. Retrieved from Uber Engineering Blog: https://eng.uber.com/freight-markov/
Zhen, L., Baldacci, R., Tan, Z., Wang, S., & Lyu, J. (2022). Scheduling heterogeneous delivery tasks on a mixed logistics platform. *European Journal of Operational Research*, 680-698.
Ziticity. (2022, February 21). *Same-day delivery*. Retrieved from Ziticity: https://ziticity.com/

8. Annexes

Annexed A. City Tessellation (Geos/Polygons)

All the instances used are in the same city: Santiago CL, for this city there is a geopolitical tessellation that is very informative in terms of socioeconomic distribution, accessibility, population density, and other variables. There are 52 geo-zones called “comunas" that we use to model this problem. The caveat of using a geopolitical tessellation is that this is not extrapolated to other cities, the cover area varies vastly from the city-core geos to the city outskirts geos, probably a more scalable approach is to use a more standardized tessellation system such as S2[6] or H3[7].

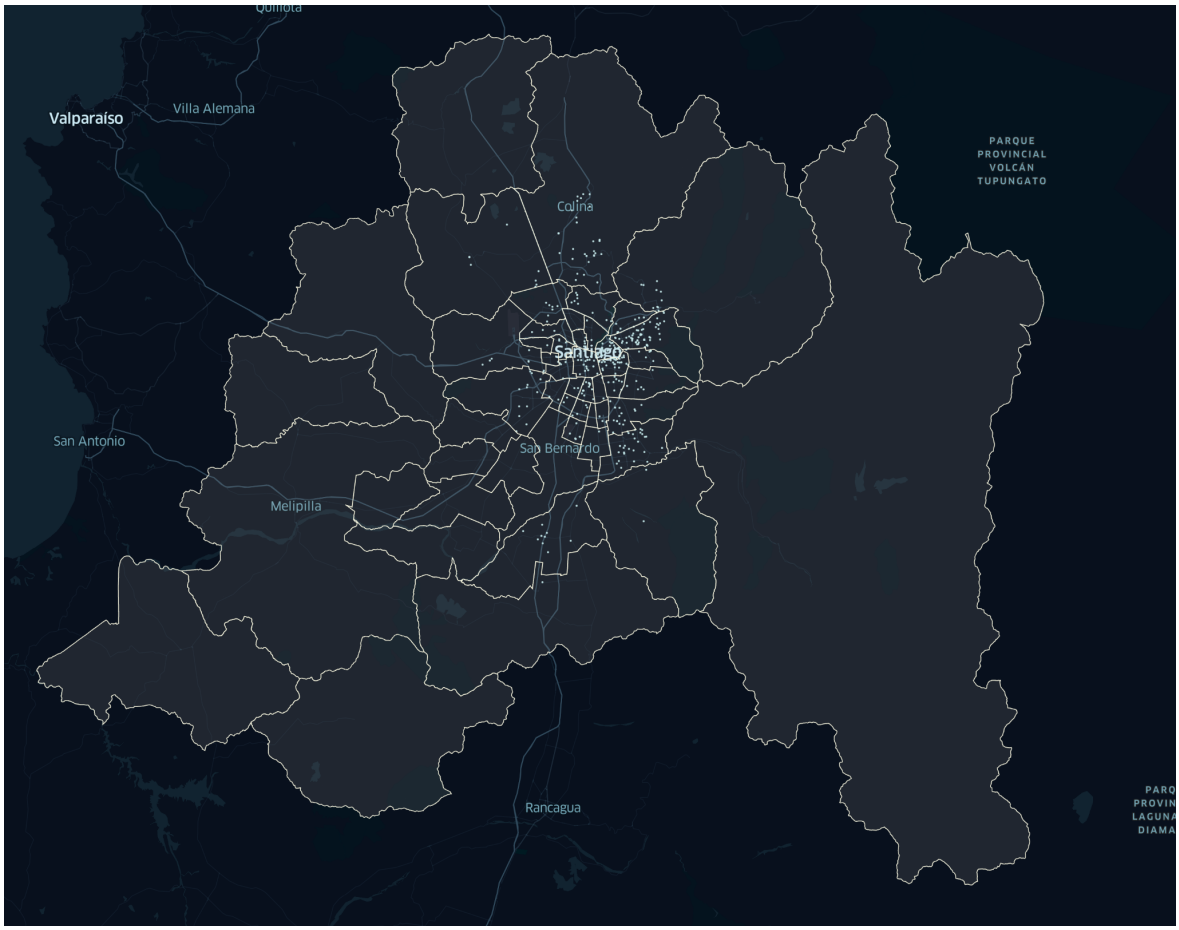


Figure 3. Santiago City Tessellation based on geopolitical zones

Annexed B. Daganzo TSP linear Approximation results

[6] *S2 Geometry*. (n.d.). S2 Geometry. Retrieved January 15, 2022, from http://s2geometry.io/

[7] *H3 geospatial indexing system*. (n.d.). H3 Geospatial Indexing System. Retrieved January 15, 2022, from https://h3geo.org/

Using the model described in 0 we estimate the parameters on the linear regression to estimate the distance of the TSP of the routes that we had in the dataset generated by Wareclouds. The results of this standard regression are the following, an Adjuster R-Squared of 0.919, and an AIC of 3957 with 515 samples. The coefficient values are in the following table.

	coef	P>|t|
cover area	0.0190	0.000
Sum linear distances to depot	0.1201	0.000

Table 4. Daganzo Coefficients estimation regression

We use the coefficients from the regression above to estimate the TSP distance in most of this paper

Annexed C. Estimating the cost by drop

To implement the heuristic, we need to estimate the impact of having a node on the total cost of a particular route. These costs allow us to prioritize more costly nodes to be swapped or transferred to other routes in a greedy approach.

We base the estimation of this cost on the objective function formula, which estimates the cost of each route given certain features that describe that route (e.g, number of drops in the route, number of warehouses in the route). We group these features into two groups: **non-separable features** and **separable features.** A separable feature is a feature that can directly separate the contribution of each drop to the final cost. For example, for the feature that sums all distances from the drops to their warehouse $dr(r) = \sum_i dr_i * Y_{r,i}$, we can simply assume that the contribution of drop i is the distance from i to its warehouse dr_i. Similarly, for the number of drops $nd(r)$, each drop contributes one unit to the total.

A non-separable feature is a feature for which the contribution of each drop depends on other drops in the route. Consider for example the case of dummies $I(g,r)$, that account for the presence of the route on a drop in a given polygon g. Even if the node $i: geo(i) = g$is removed, the corresponding cost could still be accrued by the route because there might be another node in the same polygon. To deal with this type of feature, we divide the cost of this feature between the drops that participate in that feature. In the example of the feature $I(g,r)$, we simply divide the cost by all the nodes in that feature. We operate with the rest of the features using the same approach.

Finally, we approximate the cost of each drop i (in a particular route r) by:

$$cost_r(i) = \sum_{ft\ is\ separable} \beta_{ft} * ft(i) + \sum_{ft\ non-separable} \frac{\beta_{ft}}{|\{i \in r: i\ contrib\ to\ ft\}|}$$

That formula provides us with an approximation that can be used in the heuristic for prioritization purposes.

Annexed D. Simulation dataset for the matching algorithm.

We simulated n clouders building a custom random utility function based on the market beta features solved on the driver preferences regression. We assume that the market preferences were a good approximation of the individual driver's preferences.

We assume that the clouder c has a utility function given by $\pi_c(r)$ which is modeled as a linear function where each coefficient ($w_{c,feat}$) has the same variance as the original β_{feat} in the market regression. Just to maintain some consistency we simulate each new parameter $w_{c,feat}$ with the gamma distribution keeping the sign of each parameter with its corresponding β_{feat} sign. We manually adjusted $w_{c,price}$ considering the actual range of price per node offered by Wareclouds and the normal acceptance rate that we had from the data, we tweak that value until the simulation log looks like the aggregated logs that we have from real instances.

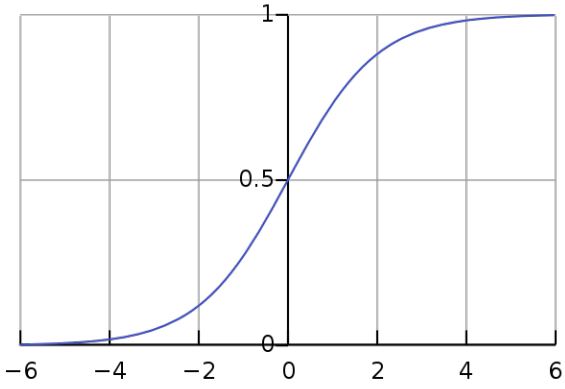
$$\pi_c(r) = \sum_{feat} w_{c,feat} * feat(r) + w_{c,price} * price(r)$$

In the clouders features, we only add the following variable, due to data limitations.

Feature Name	Description
origin distance	Distance between route centroid, and clouder origin polygon centroid in km

Table 5. Clouder features

We transform this utility function to a probability using a sigmoid function.



[8]

The sigmoid function goes from -6 to 6 in his domine; therefore, we normalize the utility function creating a "worst route" and a "best route" for each simulated clouder. The "worst route" $r_{worst}(c)$ will be the longest available route, with the lowest registered price by node, with nodes spread among all far polygons. The "best route" $r_{best}(c)$ will be the average number of nodes with all

[8] Logistic curve. (n.d.). [Graph]. Wikipedia. https://commons.wikimedia.org/wiki/File:Logistic-curve.svg

nodes only in one polygon close to the clouder origin polygon, and the highest pay by node declared by Wareclouds. We estimate then the utilities for each clouder using this synthetic route:

$$\pi_c\big(r_{worst}(c)\big) = \pi\_worst_c$$
$$\pi_c\big(r_{best}(c)\big) = \pi\_best_c$$

Using these extreme values, we normalize the utility function $\bar{\pi}_c(r)$ to be contained in the domine of $(-6,6)$ using that we can simulate a probability of acceptance using the sigmoid function on top of this normalized utility:

$$\mathbb{P}(c\ accepts\ r) = sigmoid(\bar{\pi}_c(r))$$

Whit these probabilities we can simulate a daily marketplace request acceptance log for a routing instance, we simulate using a random matching algorithm (each route is offered to a random clouder) and we simulate the acceptance using the probability value determined by the formula above. We increase the price each time the route is rejected with a parametrized growth rate. We also tested an origin-based matching, considering that that's a fairer representation of the current matching algorithm used by Wareclouds. As we mentioned with the price parameter, we tweak all the manual parameters until our simulated scenarios were similar to the scenarios that we got from the real data. With this simulated log, we can finally fit out the probability model for the market $Q(r, c, p)$ and then implement the matching algorithm.

Annexed E. Table with results of all instances

This table has all the scenarios and their corresponding cost (Estimated Time of acceptance).

Instance size	Manual Solution Cost	Heuristic Solution Cost	Model Solution Cost	Runtime Hours	Scenario id
585	258.2	360.1	244.1	5.5	S0
378	222.4	245.0	195.5	5.5	S1
597	323.6	409.1	315.3	5.5	S2
384	252.7	279.7	203.1	5.5	S3
578	309.2	390.1	274.1	5.5	S4
344	223.3	245.1	196.4	5.5	S5
336	213.2	251.5	188.6	5.5	S6
363	240.4	296.8	209.6	5.5	S7
306	234.3	235.7	204.0	5.5	S8
587	292.5	359.8	265.5	5.5	S9
453	341.8	354.7	283.0	5.5	S10
352	195.9	242.0	174.4	5.5	S11
340	226.9	232.5	193.4	5.5	S12
373	260.0	278.9	232.9	5.5	S13
455	284.9	286.7	196.3	5.5	S14
414	301.9	313.4	249.1	5.5	S15

374	258.0	274.3	210.4	5.5	S16
336	206.7	222.8	175.3	5.5	S17
369	274.3	345.6	187.1	5.5	S18
515	253.9	320.3	219.7	5.5	S19
337	239.9	254.7	201.1	5.5	S20
269	178.6	224.9	154.8	5.5	S21
254	231.4	248.5	212.7	5.5	S22
309	208.6	240.4	191.7	5.5	S23
519	293.8	351.3	254.4	5.5	S24
252	150.4	153.6	121.7	5.5	S25
358	214.7	258.7	195.2	5.5	S26
269	207.0	214.4	181.5	5.5	S27
297	227.2	227.4	188.3	5.5	S28

Table 6. Instances summary