Pricing Routes: Compensation of Crowd-Drivers

Fabian Torres[1], Michel Gendreau[2, 3], and Walter Rei[3, 4]

[1]School of Architecture, Civil and Environmental Engineering, Transport and Mobility Laboratory, École Polythecnique Fédéral de Lausanne
[2]Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport
[3]Département de mathématiques et de génie industriel, Polytechnique Montréal
[4]Département de management et technologie, Université de Québec à Montréal

Abstract.

Compensating crowd-drivers (CD) to complete delivery tasks requires careful analysis. Crowd-drivers are on-demand independent workers that can reject undesirable delivery requests. In companies like Uber, pricing delivery tasks is important to adjust the supply of Uber drivers to match the demand. Unlike Uber, e-commerce companies (e.g., AmazonFlex) have a set of delivery requests that are more cost efficient when consolidating multiple delivery requests in a single route. The compensation influences the probability of route acceptance. As a consequence, route optimization and pricing need to be integrated together to minimize the expected cost and maximize the participation of CDs in a crowd-shipping platform.

In this study, we develop a framework to create an operational plan of routes, and price routes based on the market. The probability of route acceptance by CDs is modeled with an econometric model. The model considers different route attributes that influence the probability of acceptance (e.g., distance, location, compensation, load and number of stops). The operational problem is a variant of the Heterogeneous Vehicle Routing Problem where a fleet of commercial vehicles and a pool of stochastic CDs can complete delivery requests. We develop an Adaptive Large Neighborhood Search (ALNS) algorithm to integrate all decision in a single framework. In addition, we investigate strategic objectives for the long-term success and sustainability of the platform and provide managerial insights. In particular, our study demonstrates three main results: 1) A fleet of only crowd-drivers is more robust than a fleet of commercial vehicles for the various demand scenarios 2) The compensation of CDs remains stable while incrementing participation and 3) A large increase of participation can be achieved with a small increase of the expected cost of deliveries. Our analysis offers insights for effectively pricing multiple on-demand delivery requests with independent workers.

Keywords: Crowd-shipping, last-mile, Dynamic Pricing, Stochastic Optimization, Adaptive Large Neighborhood Search, Discrete Choice Model, revenue management, multi-objective.

Acknowledgments

This research was partially funded by the Canadian Natural Sciences and Engineering Research Council (NSERC) under its Discovery Grants research program, as well as the Canada Research Chair in Stochastic Optimization of Transport and Logistics Systems, by Polytechnique Montréal, and by the Ecuadorian Secretaría de Educación Superior, Ciencia, Tecnología e Inovación (SENESCYT) under its international scholarships program and Innosuisse.

1 Introduction

In the sharing economy, Uber has spearheaded the crowd-sourced transportation service; initially, by taxi service, followed by UberEats for meal delivery. However, the expansion of the Uber model to provide different logistical services have not been extensively explored. For example, a large portion of city logistics consist of parcel delivery for e-commerce. The Uber-like operational business model consists of multiple pickup and delivery operations for on-demand workers. In contrast, the delivery of packages for e-commerce requires delivering a large number of packages in a single route to make deliveries cost-efficient. Online shoppers expect low-cost next-day delivery and consider the total cost (i.e., including delivery cost) before purchasing products on e-commerce platforms. In addition, fulfillment centers are distant from residential areas and city centers, making single pickup and delivery operations expensive. As a consequence, the capacity to fit multiple packages in a vehicle from the fulfillment center is relevant for long routes with multiple visits to be cost-efficient. These properties have delayed the translation of Uber models for parcel delivery. Nevertheless, large companies are expanding their logistic operations to include crowd-shipping. Amazon has shown the feasibility of the crowd-shipping business model by implementing their own crowd-shipping platform called AmazonFlex. Like Uber, drivers download the Flex app and click on offers to work time blocks, during which they are paid to execute deliveries from a particular warehouse, over a particular time window. Unlike Uber, they allow drivers to schedule work up to a week in advance. Challenges involve scheduling drivers over time, in the presence of long lead-times, uncertainties in both demand and supply, while minimizing cost and the risks of late deliveries or excess drivers (AmazonFlex, 2021a). To participate in AmazonFlex, FlexDrivers are required to have a vehicle of a certain size (i.e., a 4-door sedan). Routes are then suggested to potential drivers that can accept or reject routes based on their own preference. In Figure 1a, we exhibit the AmazonFlex app interface that displays routes to drivers and offers a compensation. At the left of the user interface, route attributes are shown and at the right side of the interface the compensation (i.e., price) of the route is displayed.

For e-commerce platforms (e.g., Amazon) deliveries are known a day or two in advance, thus, the operational planning of deliveries consist of deciding on what vehicles will be used to complete the daily deliveries. Hence, the set of delivery requests can be done by a mix of Professional Drivers (PD) and Crowd-Drivers (CD). In Figure 1b, we show the main decisions to implement a crowd-shipping model. Routes are planned and then priced based on the probability of acceptance of CDs with an econometric model.

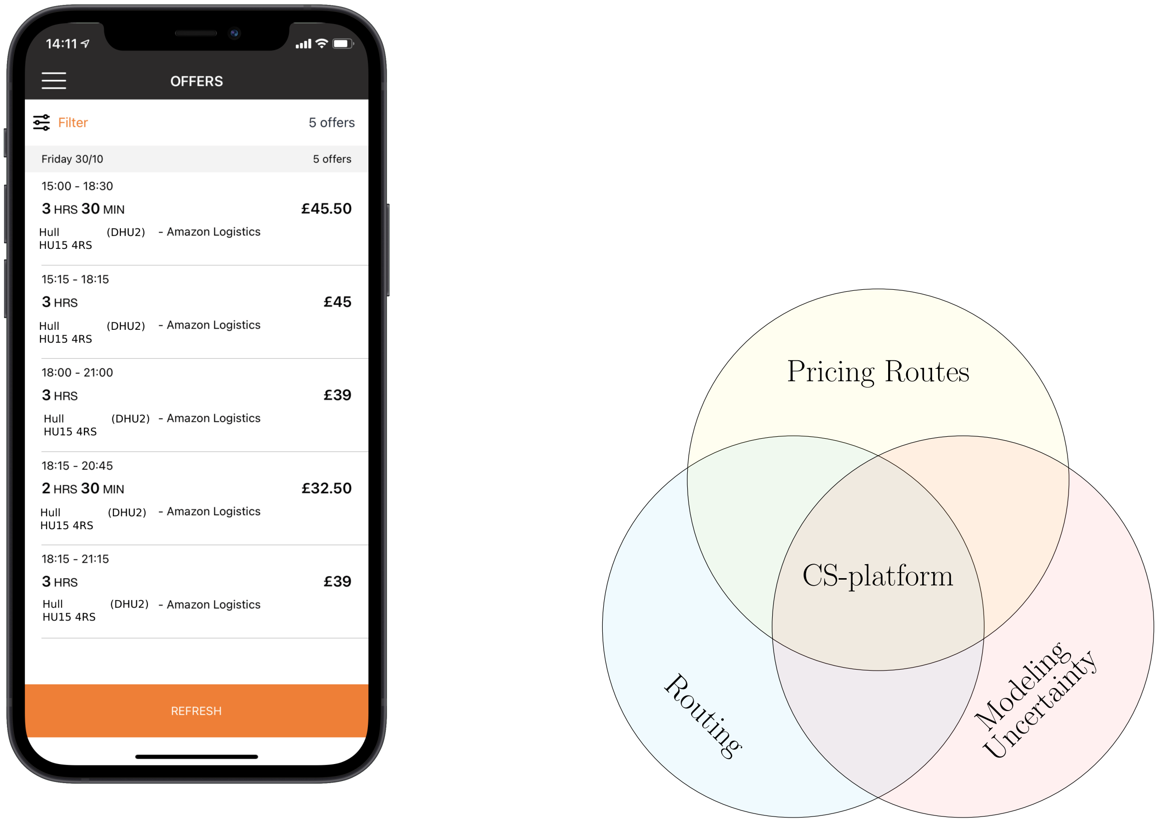
We consider a stochastic two-stage decision problem, in the first-stage the CS-platform plans routes for PDs (i.e., with a known cost) and CDs (i.e., with a price). Naturally, to reduce the cost of deliveries, the CS-platform would like to price routes as low as possible. However, the probability of acceptance of CD-routes are monotonically increasing with the price that the CS-platform is willing to pay, i.e., if the price of a route is too low, the probability of acceptance will also be too low. In the second-stage, CDs accept or reject routes. Unfortunately, missing deliveries can impact customer satisfaction and reduce customer trust in the CS-platform. In the second-stage, rejected routes require expensive recourse actions to complete deliveries to maintain a good service quality. Examples of second-stage recourse actions include paying overtime to PDs to extemporaneously complete routes or pay external couriers. To model the uncertainty of route acceptance, we choose to use tools from econometrics. Econometric models are typically used to model complex behaviour in a market. In contrast to machine learning models, econometric models allow interpretation of the parameters and allow us to derive valuable managerial insights. Hence, we assume that a binary logit model is available that models the probability that the pool of drivers will accept or reject a route based on routes attributes (e.g., price, route length, number of stops, route load and location). Hence, the routes and the price influence the probability of acceptance, i.e., the attributes of routes are variables in the logit model. Balancing carefully the creation of routes, pricing and the probability of acceptance (i.e., logit model) need to be done together in the two-stage framework to ensure the best policy that reduces the cost of deliveries.

The long term feasibility of a CS-platform depends on the participation of CD and their satisfaction with the system. CDs are stakeholder of the platform and if they are unsatisfied, they can migrate to other platforms. In this paper, we argue that route pricing has a strategic component to maintain long-term participation in the platform and to keep CDs' satisfaction. Pricing routes with the sole objective of minimizing the expected cost might not be the best strategy for the long-term. In fact, asymmetric information exist between the CD and the CS-platform. CDs know more about their preferences than the CS-platform. They will choose the best routes and leave undesirable, low paid, long routes with multiple stops and migrate to another platform. To reduce the effect of adverse selection, we propose the second objective to maximize participation and generate a stable compensation to CDs.

Contributions

Motivated by these challenges, this paper presents a novel framework to address the complexities inherent in pricing CD-routes. Specifically, we focus on integrating route optimization and pricing in crowd-shipping environments. Our main contributions are threefold:

- We integrate three important facets together: 1) the operational component of routing vehicles, which consists of grouping delivery requests to minimize the expected cost. 2) The pricing component to determine the compensation to pay CDs if they complete routes, which is directly linked to the routing, and 3) the uncertainty of route acceptance modeled with an econometric model. The integration of combinatorial optimization with advanced econometric models is a growing area of research Haering et al. (2023). Discrete choice models allow for a better representation of uncertainty in optimization problems and it allows the interpretation of results.



(a) AmazonFlex interface (AmazonFlex, 2021b)

(b) Decisions for CS-platform

Figure 1: Routing and pricing integration

- We introduce an Adaptive Large Neighborhood Search (ALNS) meta-heuristic method to integrate the pricing with route optimization and with the uncertainty of route acceptance of crowd-drivers. We create new construction operators in the ALNS that use information about the probability of route-acceptance to rebuild the destroyed solution. In addition, we embed in the ALNS a Route Assignment and Pricing (RAP) procedure that takes a subset of routes and determines their optimal price and probability of acceptance. We show that our ALNS algorithm outperforms the column generation heuristic for a simpler case study proposed in Torres et al. (2022b). We present extensive computational experiments and sensitivity analysis of the parameters of the econometric model that provide insights in the changes of behaviour of crowd-drivers.

- Finally, we consider a multi-objective problem where the first objective is to minimize the expected cost of routing and the second objective is to maximize the participation of crowd-drivers. We investigate three strategic questions: 1) how can the participation of crowd-drivers be increased? 2) What is the average compensation of crowd-drivers 3) Can the fleet be replaced by crowd-drivers? We present valuable insights about each question that can be used in a CS-platform and show the ability of our framework to offer managerial insights.

The remainder of this paper is organized as follows. Section 2 provides a comprehensive review of the related literature, contextualizing our research within the broader landscape of crowd-shipping and operational research. In Section 3, we delineate the adaptation of our model to the specific problem domain. Section 4 describes the problem of optimally pricing a set of routes using the golden section search algorithm. Section 5 presents a detailed exposition of our ALNS meta-heuristic methodology, describing its design, implementation and integration with the pricing. In Section 6, we present the results of extensive computational experiments, and show the reliability of ALNS and a sensitivity analysis with the parameters of the econometric model and provide insights of the changes of behaviour of crowd-drivers. In Section 7, we discuss the strategic long-term objectives shedding light on optimal pricing strategies. Finally, Section 8 concludes the paper and offers future research directions.

2 Related work

A large portion of the literature on crowd-shipping focuses on deterministic problems that consider individuals that have a planned trip and that can carry a package, where the price of requests are already determined by a rule called a compensation scheme and CDs are considered to be deterministic. (Mohri et al., 2023; Archetti et al., 2016, 2021; Alnaggar et al., 2021; Macrina et al., 2017, 2020; Dahle et al., 2019; Macrina and Guerriero, 2018; Vincent et al., 2022, 2023; Stokkink et al., 2024; Stokkink and Geroliminis, 2023; Zhang and Zhang, 2024; Kızıl and Yıldız, 2023; Tao et al., 2023; Yang et al., 2024; Pugliese et al., 2023; Le et al., 2021). Most of the deterministic variants consider a matching problem where delivery requests for CDs are assigned to predetermined planned trips.

Another part of the literature, more closely related to this paper, studies CDs as agents that can accept or reject routes with a given probability (Gdowska et al., 2018; Mousavi et al., 2021; Dayarian and Savelsbergh, 2020; Skålnes et al., 2020). Generally, the probability is assumed to be exogenous and to be a predetermined value. In some stochastic variants (Silva et al., 2023; Torres et al., 2022b,a; Alnaggar et al., 2024) the probability is described by distribution. Compensation guarantees of CDs was considered by Alnaggar et al. (2024) in a dynamic setting where CDs and deliveries arrive dynamically with a given probability. CDs delivery a single package and different deterministic compensation policies are examined to evaluate CDs' welfare.

In Barbosa et al. (2023), the compensation is consider to be a variable that can influence the probability of acceptance of a single delivery request by a CD. The authors use a binary logistic regression with 3 explanatory variables that determine the probability of acceptance, i.e., distance, load, and compensation.

Overall, the compensation of CDs (even in stochastic variants) has largely been consider as a fixed cost proportional to the deviation of the planned trajectory of CDs (e.g., Archetti et al. (2016)). However, there are a few studies that consider the compensation as a variable that can influence the acceptance of a delivery request Barbosa et al. (2023). Clearly, a higher compensation will persuade a CD to accept a request, thus, to minimize the expected cost, a higher compensation needs to be considered. Pricing problems are problems where the price of a product has to be determined to maximize the profit with demand uncertainty (e.g., Haering et al. (2023)). In addition, the combination of Discrete Choice Models (DCM) and combinatorial optimization is a growing area of reasearch (Ricard and Bierlaire (2024)). In our case, we would like to determine the price of a route to minimize the cost of deliveries and maximize participation using a DCM to model the probability of route acceptance.

In last-mile deliveries, CS-platfomrs (e.g., AmazonFlex) have a large set of delivery requests that benefit from consolidation in a single route for cost efficiency. It is imperative to develop a framework that considers multiple deliveries by CDs, dynamic pricing to set the price for each route, multiple objectives (e.g., to maximize participation) and a DCM that allows the representation of the acceptance probability with different route attributes. In table 1, we summarize the contribution of this paper with respect to the current literature. The first column specifies the studies that consider dynamic pricing, i.e., where the price is adjusted to optimize an objective. The second column (i.e., **Mult.-Deliv**) indicates the studies that consider the possibility of 2 or more delivery requests performed by CDs. The third column (i.e., **CD-VRP**) indicates the studies that solve a VRP for CD delivery tasks. The forth column (i.e., **DCM**)indicates the use of a Discrete Choice Model to explain the probability of acceptance, the next column indicates the use of recourse actions in case of failed deliveries due to a lack of supply of CDs, and the last column (i.e., **Muli.-Obj**) indicates the stochastic variants that consider another objective besides minimizing the expected cost.

Study	**Pricing**	**Mult.-Deliv.**	**CD-VRP**	**DCM**	**Recourse**	**Mult.-Obj**
Dayarian and Savelsbergh (2020)	·	✓	·	·	·	·
Gdowska et al. (2018)	·	·	·	·	·	·
Barbosa et al. (2023)	✓	·	·	✓	·	·
Torres et al. (2022b)	·	✓	✓	·	✓	·
Dahle et al. (2017)	·	·	·	·	·	·
Torres et al. (2022a)	·	✓	✓	·	✓	·
Mousavi et al. (2021)	·	·	·	·	·	·
Skålnes et al. (2020)	·	·	·	·	·	·
Le et al. (2021)	·	·	·	·	·	·
This paper	✓	✓	✓	✓	✓	✓

Table 1: Summary of stochastic variants

3 Problem description

In this Section, we formally define the problem and present a two-stage stochastic programming model. The problem is a variant of the Heterogeneous Vehicle Routing Problem (HVRP), where the fleet of vehicles have different capacities and different attributes. The delivery requests have to be fulfilled from a central depot with the available vehicles. In the case of crowd-shipping the mixed fleet of vehicles consist of homogeneous company vehicles driven by professional drivers (PD) and a pool of CDs that can reject routes with a given probability. The compensation of PD is given deterministically by the distance of each route, in contrast, the compensation of CDs is determined based on the probability of acceptance of the route by CDs (see Subsection 3.2).

We consider a two-stage stochastic model. In the first stage, the set of delivery requests is separated into PD-routes and CD-routes. In the second stage, some CD-routes can be rejected. If CDs refuse to complete a set of routes, there will be a set of customers with unfulfilled deliveries. In order to fulfill all deliveries, expensive recourse actions need to be taken. In our setting, we consider the cost of the recourse actions to be the cost to complete rejected CD-routes by a PD times a penalty, i.e., $\aleph > 1$, the penalty represents an extra cost for completing rejected CD-routes extemporaneously (e.g., paying overtime to drivers). To avoid paying the cost of expensive recourse actions, the price of each CD-route has to be estimated to influence and increase the probability of acceptance of each CD-route. This property, requires the solution of a pricing problem for each subset of CD-routes to determine the adequate compensation to reduce the risk of recourse actions and the optimize the objectives.

Specifically, let d_r be the total distance of a route. We consider the cost of a PD-route (i.e., compensation to the professional driver) to be equal to the distance of the route, i.e., d_r. On the other hand, the expected cost of a CD-route depends on the price, i.e., c_r, the probability of route acceptance (i.e., p_r) and the recourse cost in case the CD-route is rejected. The expected cost of a CD-route "r" is the following:

$$p_r c_r + (1 - p_r) \times \aleph d_r \tag{1}$$

A solution to the first-stage problem will consist of a set of PD-routes with a cost of d_r for each route given to a PD, and it will consist of a set of CD-routes with an expected cost equal to equation (1). In equation (1), the price of the route is paid to the CD only if the CD accepts the route, on the other hand, if the CD rejects the route the cost is the cost of a PD time the penalty (i.e., $\aleph d_r$). The probability is modeled with a DCM and will be described in more detail in subsection 3.2, while the optimisation model is described in the following subsection.

3.1 Optimization Model

In this Subsection, we present a variant of the set partitioning formulation for the Heterogeneous Vehicle Routing Problem with CDs. The model was first introduced by Torres et al. (2022b) for a routing problem with stochastic supply of CDs.

Let Ω be a large set that contains all feasible routes and let ω_r be a binary variable that equals to one if PD-route $r \in \Omega$ is selected in the solution and zero otherwise and let $\bar{\omega}$ be a binary variable that is equal to 1 if CD-route is selected in the solution and zero otherwise. Let a_i^r be a parameter that is equal to the number of times customer $i \in N$ is visited in route $r \in \Omega$:

$$\min \sum_{r \in \Omega} d_r \omega_r + \mathscr{Q}(\bar{\omega}) \tag{2}$$

$$\text{s.t.} \quad \sum_{r \in \Omega} a_i^r \omega_r + \sum_{r \in \Omega} a_i^r \bar{\omega}_r = 1 \qquad \forall i \in N \tag{3}$$

$$\omega_r, \bar{\omega_r} \in \{0, 1\} \qquad \forall r \in \Omega$$

In the first stage, the objective function (2) minimizes the cost of PD-routes (i.e., $d_r\omega_r$), in the second stage, we minimize the expected cost of CD-routes (i.e., $\mathscr{Q}(\bar{\omega})$). Constraints (3) ensures that all customers are visited exactly once by either a PD or a CD.

Let $\mathcal{H} \subset \Omega$ be the set of CD-routes that are equal to one in a feasible solution to the first stage problem and let α be a weight that is used as a discount factor to increase participation by discounting the price paid to CD. In this way, the second objective to maximize participation of CDs can be achieved by setting the discount factor to a sufficiently small value. The second stage recourse function $\mathscr{Q}(\bar{\omega})$ in the objective (2), is then formulated as follows:

$$\mathscr{Q}(\bar{\omega}) = \min_{c_r \in \mathbb{R}} \sum_{r \in \mathcal{H}} \bar{\omega}_r \left(\alpha p_r c_r + (1 - p_r)\aleph d_r\right) \tag{4}$$

$$\mathcal{H} = \{r : \bar{\omega}_r = 1, \forall r \in \Omega\}$$

The second stage recourse function (4) is a minimization problem where the objective is to minimize the expected cost of CD-routes by estimating the price. Naturally, we would want to set the price to zero to minimize the payment to CDs and reduce the cost. However, the probability of route acceptance is a function of the price. If the price is too low, then the CD-route will be rejected and the CS-platform will have to pay the cost $\aleph d_r$. An optimal price has to be set to minimize the expected cost.

In Figure 2, we can visualise the two-stage model, in the first stage we build routes that pass to the second stage where the price of the subset of CD-routes is determined and then, the expected cost is considered in the first stage to find routes that have better expected costs.

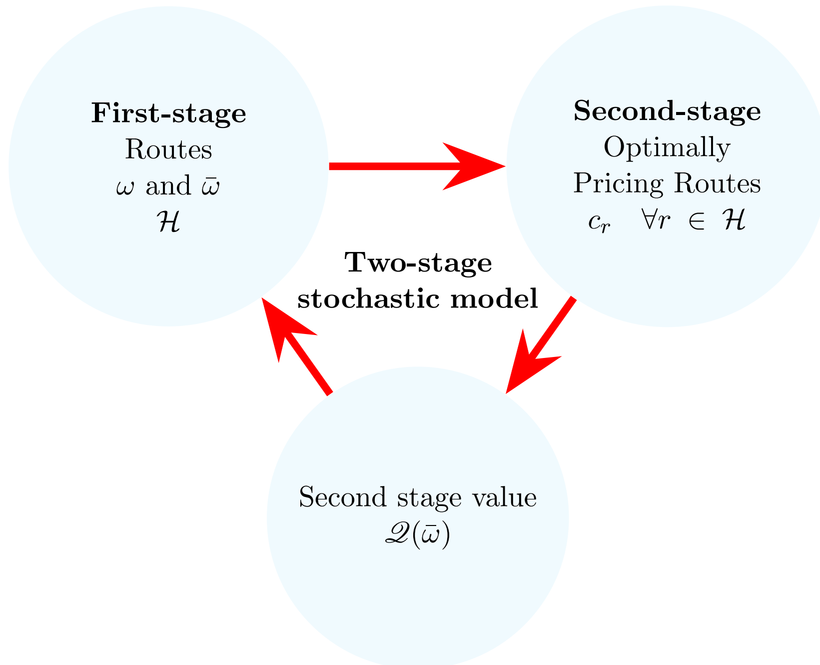


Figure 2: Two-stage model

3.2 Choice Model

The preference of CDs is modeled with a binary logit model. Clearly, a delivery task is a difficult activity that drivers do not do for pleasure. Hence, paying drivers is necessary to motivate them to complete routes. Higher compensation will monotonically increase the likelihood of route acceptance. Conversely, other attributes of a route that increase the difficulty of completing the route (e.g., distance traveled) will monotonically decrease the probability of route acceptance. Any attributes can be considered in our methodology but for the sake of example, we have selected a few attributes of a route that can influence the probability of acceptance:

- **Route length (x_1):** Routes that are long are less desirable for CD. For a given route the route length can be easily computed as the total distance traveled from the depot to all the visited customers and back to the depot.

- **Load (x_2):** CD-vehicles are not designed to transport packages. Thus, we assume that a lighter load is a desirable characteristic of a route. The load is the sum of all the demands of customers in the CD-route.

- **Stops (x_3):** Parking and stopping the vehicle to deliver packages is not desirable by drivers. Thus, we assume that routes with less stops will be preferred. The stops are the number of customers in a route.

- **Location (x_4):** Some neighborhoods are not as pleasant as others. We consider a penalty for customers that are located in a bad area. Every customer is give a value based on the location. The sum of the values of all customers in a CD-route give an estimate of how desirable a route is.

- **Price (c):** Crowd-drivers are compensated to deliver packages. Thus, we assume that they will prefer to select routes that have a higher compensation. The price of a route has to be estimated in order to minimize the expected cost.

Let $\mathcal{U}_r$ be the utility of a route, let $\mathcal{A}$ be the set of attributes, and let β_i be the set of parameters of the logit model for each attribute. The opt-out option is described by the utility $\mathcal{U}_0 = 0$. The following is the choice model:

$$\mathcal{U}_r = \sum_{i \in \mathcal{A}} \beta_i x_i^r + \beta_c c_r + \varepsilon \quad \forall r \in \mathcal{H} \tag{5}$$
$$\mathcal{U}_0 = 0$$

Equation (5) models the utility of a route based on random utility theory where the error term (i.e., ε) follows a Gumbel distribution and represents unobserved attributes of the route. The probability of acceptance of a route is equal to $p_r = \frac{e^{V_r}}{e^{V_r}+1}$, where V_r is the deterministic part of the utility.

3.3 Adverse selection

In this paper, we do not consider the dynamic aspects of CDs arrival, however, we briefly discuss the possible effects in change of behaviour here. The long-term sustainability of the CS-platform depends on the participation of CDs and their satisfaction with the CS-system. CD-routes with utilities or prices that fluctuate excessively are not a desirable outcome for the long term satisfaction of CDs. Adverse selection can cause the quality of CD-routes to decrease throughout the day as higher quality routes are removed from the choice set first. When considering the dynamic arrival of CDs, the first to arrive will choose the best route, eliminating this choice from future CDs. In Figure 3, we show a set of routes with utilities ranked from highest to lowest where CDs arrive dynamically. CDs that arrive early, will be able to pick routes that have a higher utility, leaving routes that have a lower utility for future drivers. Only routes that are not desirable will be left. If this happens CDs can just give up all together are return the next day hoping to get a better route if they come early. In contrast, in Figure 4, we show a sample of route choice by a set of M CDs that arrive in sequence to select routes if all utilities or compensation for routes are similar, adverse selection will not happen, and thus, the quality of routes will not decrease throughout the day. To reduce adverse selection, more information about CDs' preferences is required to create routes that are of higher quality. In addition, strategies that increase participation or standardize the compensation can be used (e.g., considering a discount for CD-routes in the objective). In Section 7, we focus on strategic objectives that can help reduce adverse selection, e.g., increasing participation and standardizing compensation for CDs.

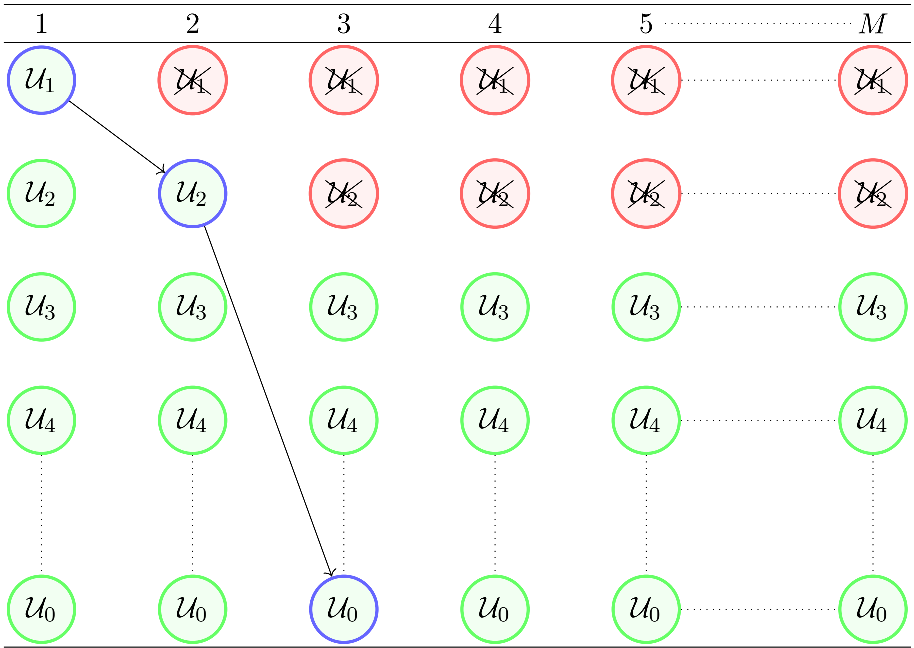


Figure 3: Route choices with adverse selection

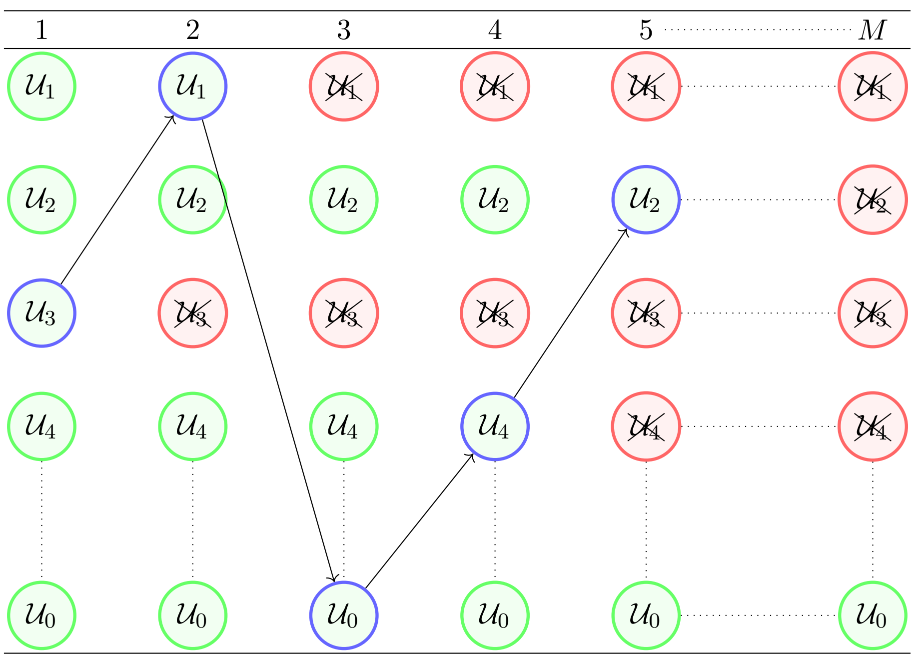


Figure 4: Route choices without adverse selection

4 Pricing Routes

In this Section, we consider a solution to the first stage where we are given a set of CD-routes (i.e., $\mathcal{H}$). The second stage problem (4), is to determine the optimal price of each route with the probability of route acceptance. Hence, we will now attempt, in this Section, to optimize the second stage (4) by determining the optimal price with the probabilistic model described in detail in Section 3.2.

Let x_r be the sum of all the attributes and parameters of route $r \in \mathcal{H}$ without the price (i.e., $x_r = \sum_{i \in \mathcal{A}} \beta_i x_i$). By performing some basic algebraic operations on the utility function (5) to clear the price on the left hand side, we get the following equality for the price:

$$c_r = \frac{V_r - x_r}{\beta_c} \tag{6}$$

The price of a CD-route (i.e., c_r) in equation 6 is equal to the deterministic part of the utility function (i.e., V_r) minus the sum of the attributes (i.e., x_r) and divided by the parameter of the price (i.e., β_c).

Now, by replacing the price in the objective function with equation (6) we arrive at the following second stage problem:

$$\mathscr{Q}(\bar{\omega}) = \min \sum_{r \in \mathcal{H}} \bar{\omega}_r \frac{e^{V_r}}{e^{V_r} + 1} \cdot \frac{V_r - x_r}{\beta_c} + \sum_{r \in \mathcal{H}} \bar{\omega}_r \left(1 - \frac{e^{V_r}}{e^{V_r} + 1}\right) \cdot \aleph d_r \tag{7}$$

The function (7) is nonlinear with multiple variables for each route. However, the problem is separable by route. We can solve the optimal price for each route separately without affecting the global optimal.

Lemma 1. *If a price c_r^* for a given route is the price that minimizes the expected cost of route r, then, c_r^* is also part of the global optimal solution to problem* (7).

Proof. It can be seen that the problem is separated by routes and that the price of one route does not have an effect on the price of other routes, due to the i.i.d. assumption from the logit model. □

In order to find a solution we must prove that the function for each route is unimodal and that there is a unique global minimum that exists. To proof this, we have to use Lambert W function:

Proposition 1. *The unique global minimum of the expected cost of a single route $r \in \mathcal{H}$ is at $V_r = x_r + \beta_c \aleph d_r - 1 - W\left(e^{x_r + \beta_c \aleph d_r - 1}\right)$.*

Proof. See appendix A □

With proposition 1 we can compute the optimal value of any CD-route and by replacing the optimal value in equation (6), we can get the optimal price for each route.

Corollary 1. *Given a fixed set of CD-routes, i.e., H, the second stage problem (7) has a unique global minimum at $V_r = x_r + \beta_c \aleph d_r - 1 - W\left(e^{x_r+\beta_c\aleph d_r - 1}\right)$ for all $r \in \mathcal{H}$.*

Corollary 1 follows from Proposition 1. Hence, given a set of CD-routes from the first stage problem we can find the optimal price for each route that minimizes the expected cost. However, the solution to the pricing problem is not optimal for model (2) that considers the first-stage and the second-stage costs together. In fact, the subset of CD-routes can change to a better configuration that reduces the expected cost. The binary variables, $\bar{\omega}$ can change leading to better solutions.

Proposition 2. *If the price of a route is larger than the distance of a route, i.e., $c_r > d_r$, a cost reduction can be achieved by using a PD to complete the route in the first stage, i.e., $\omega_r = 1, \bar{\omega_r} = 0$.*

Proof. Recall that PD have a capacity that is larger than that of a CD, thus, any route assigned by a CD can also be assigned to a PD. In addition, the cost of a PD is equal to the distance. Hence, the cost reduction will be equal to $c_r - d_r$. $\square$

Corollary 2. *Let $\mathbb{E}\{c_r^*\}$ be the optimal expected value for the second-stage function for route $r \in \mathcal{H}$ including the expected cost of the recourse action, i.e., $\aleph d_r$. If the expected cost is larger than the cost of a PD (i.e., $\mathbb{E}\{c_r^*\} > d_r$) then, removing CD-route "r" from the set of routes (i.e., $\mathcal{H} \setminus \{r\}$) reduces the global cost of the two-stage problem.*

In Figure 5 we plot function 7 for a single route. The optimal value is bellow the threshold of d_r, thus, the optimal solution is the value that minimizes the function.

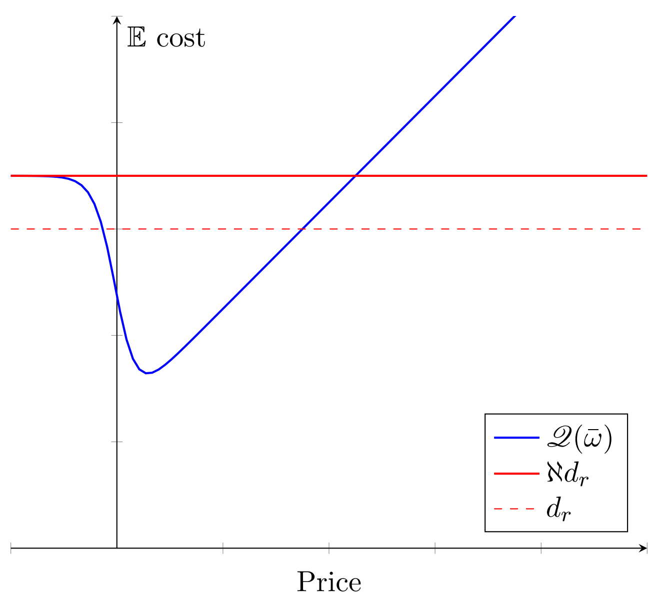


Figure 5: Expected cost and Price

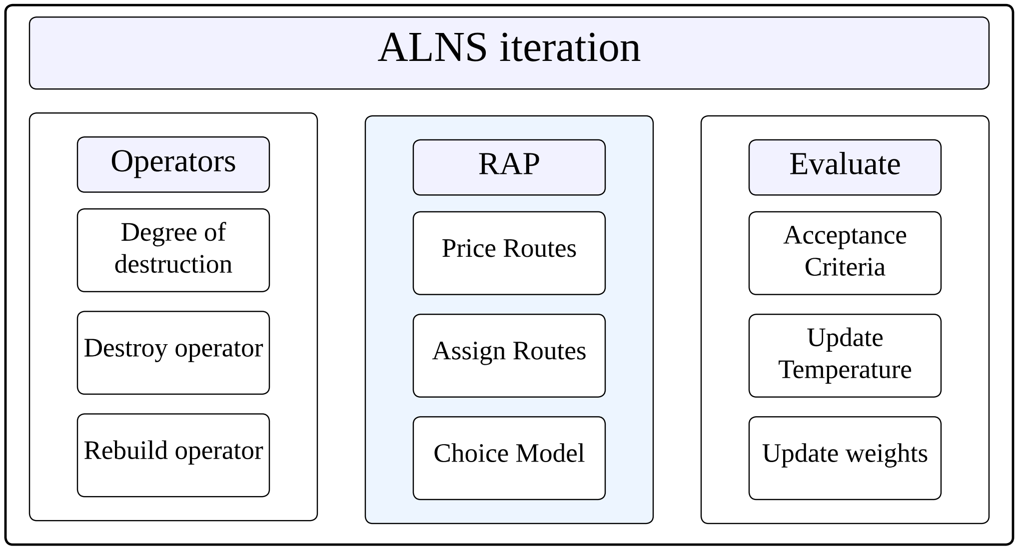


Figure 6: Single iteration of ALNS

5 Adaptive Large Neighborhood Search with Pricing under Uncertainty

In this section, we describe the ALNS algorithm that we use in this paper. In Section 5.1, we outline the general procedure of ALNS. In Section 5.2, we present the removal operators, while, in Section 5.3, we present the insertion operators. Finally, in Section 5.4, we introduce a new procedure that prices routes optimally assigns the route to a CD or PD.

5.1 Overview of ALNS

ALNS was first introduced in Shaw (1998). The main idea is to improve a solution by destroying and repairing the solution iteratively. Heuristics based on ALNS have been successfully used to solve various vehicle routing and scheduling problems (Akpinar, 2016; Korsvik et al., 2011; Dayarian et al., 2016). In Algorithm 1, we delineate the general framework of the ALNS mete-heuristic that is used. An initial solution is created using the well-known Clarke and Wright savings heuristic. In each iteration, generate the degree of destruction which is the percentage of customers to be removed from the current solution. Next, we select a removal operator to remove customers. Then, we select an insertion operator and re-insert the removed customers back into the solution, and finally, we price the routes and assign them to either PD or CD with the Route Assignment and Pricing (RAP) procedure described in Section 5.4. The new solution is accepted if it improves the incumbent solution; otherwise, we use the acceptance criteria which is borrowed from simulated annealing (see Van Laarhoven and Aarts (1987); Dowsland and Thompson (2012); Kirkpatrick et al. (1983); Afifi et al. (2013)). The initial selected temperature is gradually reduced with each iteration. As the temperature cools down, the probability of accepting a non-improving solution decreases. When the temperature cools down below a certain threshold, i.e., Temperature $< \epsilon$, we stop. At each iteration, the weight of the operators are updated based on their performance in improving the solution. The removal and insertion operators used within the ALNS are adapted or inspired by operators used in other works (Pisinger and Ropke, 2019; Koç et al., 2014, 2015; Pisinger and Ropke, 2007; Paraskevopoulos et al., 2008).

5.2 Removal operators

The removal operators help escape local optima by destroying a large portion of the solution in each iteration. The amount of the solution to be destroyed is called the degree of destruction, in our case, we set the degree of destruction to be a random number of customers to be removed from 5 to 25% of the solution. Using different removal operators can help diversify the neighbourhoods that are explored. We consider five removal operators, Near Removal (NR), Random Removal (RR), Random Route Removal (RRR), First Customers Removal (FCR), and Last Customers Removal (LCR). In each iteration, a single removal operator is randomly selected to destroy the solution, based on the roulette wheel probability.

- *Near Removal (NR):* Randomly selects a customer i and sets it to be the current customer, then proceeds to remove the nearest neighbor and updates the current customer to nearest and repeats the procedure until the degree of destruction is achieved.

- *Random Removal (RR):* Randomly selects a set of customers to remove from the solution with uniform probability.

- *Random Route Removal (RRR):* Randomly selects a set of routes to remove from the solution with uniform probability. When applied, this operator randomly removes between 2 routes and 25% of all routes in the solution.

- *First Customers Removal (FCR):* Removes a random number of customers from the beginning of all routes. We iterate over all routes and select a random number of customers to remove with equal probability. A random number of customers from 0 to at most 50% are removed from each route. This is similar to a time-based removal. Customers that are scheduled at the beginning of a route tend to have earlier time windows and can easily be replaced by customers from the beginning of other routes.

- *Last Customers Removal (LCR):* Removes a random number of customers from the end of each route. Similar to FCR, the last customers in a given route have later time windows; by removing customers at the end of routes, the removed customers have compatible time windows and can be switched when inserted back into the solution. Just like FCR, the number of customers removed for each route is random.

5.3 Insertion operators

Insertion operators rebuild the destroyed solution and try to find better solutions in different neighborhoods. We consider five insertion operators: Greedy insertion (GI), Greedy insertion with noise (GIN), PD first insertion (PDFI), CD first insertion (CDFI), and utility insertion (UI). In each iteration a single insertion operator is randomly selected to rebuild the destroyed solution.

- *Greedy insertion (GI):* Selects a customer from the list of removed customers and inserts it to the best feasible position. Let ℓ be the customer selected for insertion and let $\gamma_s(i,j) = (d_{i\ell} + d_{\ell j} - d_{ij})$ be the variable cost increase of inserting customer ℓ in between customers (i,j); Customer ℓ is inserted at the best place that has the minimum cost of insertion, i.e., $(i,j) = \arg\min_{(i,j)}\{\gamma_s(i,j)\}$

- *Greedy insertion with noise (GIN):* This operator is similar to GI except that a noise function is considered. A customer is selected from the list of removed customers and inserted at the best location based on a noise function. The noise function diversifies the solution by allowing customers to be inserted in a different place than the cheapest insertion. The cost of insertion is $\gamma_s(i,j)(1 + 0.1\phi)$, where ϕ is a random number from -2 to 2.

- *PD first insertion (PDFI):* Uses GI to insert a customer from the removed list to the PD-routes first. If no feasible insertion in a PD route exists, then the customer is inserted in the cheapest position within a CD-route.

- *CD first insertion (CDFI):* It is the opposite of PDFI. First it finds the cheapest insertion in CD-routes, if no feasible insertion exits, then the customer is inserted in the cheapest position of PD-routes.

- *Utility insertion (UI):* It consider the total increase of the utility function, and inserts the customer in the place with least decrease of the utility.

5.4 Route Assignment and Pricing (RAP)

The Route Assignment and Pricing procedure integrates the routing with the pricing decisions. Given a set of rebuilt routes at each iteration of the ALNS algorithm, the RAP procedure has to price and assign some routes to either CD or PD. Routes assigned to PD have a cost equal to the distance route, while the expected cost of a CD-route is equal to the price and the cost of recourse actions.

The result of Corollary 1 shows that the objective function is unimodal and has a unique global optima. To find the optimal solution we use the golden section search method, which is guaranteed to find the optimal point with some small error in a finite number of iterations (Nocedal and Wright, 1999).

We begin the procedure by separating all routes into two sets of routes, CD-routes and PD-routes, based on the capacity and load. For instance, if a route has a load that exceeds the capacity of a CD, then we assign the route to a PD. Otherwise, we set the route to be a CD-route by default. Given a set of CD-routes, i.e., $\mathcal{H}$, we commence the following procedure for each route:

1. Find the optimal solution that minimizes the expected cost using the golden section search.

2. Check the bound of the expected cost, if $\mathbb{E}\{c_r^*\} > d_r$, then assign route to PD.

3. Else assign route to CD.

In Algorithm 1 we summarize the ALNS procedure. The algorithm starts with an initial solution then it selects a removal operator. Next the algorithm RAPs to rebuild, re-optimize and reassign routes to CDs. The insertion procedure is selected with the roulette wheel probability and the ALNS algorithm RAPs again. Afterwards, the ALNS algorithms continues as usual, evaluating the solution with simulated annealing criteria and updating the corresponding weight based on performance.

Algorithm 1: Overview of ALNS

```
Incumbent sol = Initial solution;
Sol = Incumbent sol;
Set k;
Set Temperature;
Set Cooling rate;
for it = 1 to k do
    Sol* = Sol;
    Select removal operator;
    RAP, see Section 5.4;
    Select insertion operator;
    RAP, see Section 5.4;
    if Sol* value < Incumbent sol value then
        Incumbent sol = Sol*;
    end
    if Accept(Sol*, Sol) then
        Sol = Sol*;
    end
    Cool Temperature;
    if Temperature < ϵ then
        Stop;
    end
end
return Incumbent sol;
```

6 Computational Results

In this section, we perform computational experiments to evaluate the solution approach. The ALNS algorithm was implemented in Java SE 1.8.0 and it was run in a Red-Hat Enterprise Linux 9.0 system with an Intel(R) Xeon(R) Platinum 8360Y CPU at 2.40GHz, and 8GB of RAM.

The instances that are used in this Section are taken from the well-known Solomon instances for the VRP. Solomon instances are divided in 3 main groups, clustered instances (i.e., C) random-and-clustered instances (i.e., RC) and random instances (i.e., R). Initially, we consider C1, RC1 and R1 instances that have at most 100 customers to compare ALNS with existing methods. Later, we consider larger instances (i.e., C2, RC2 and R2) with 200 customers. The remainder of this section is organized as follows. In Section 6.1, we evaluate the performance of the ALNS algorithm when compared to the column generation algorithm presented in Torres et al. (2022b), the problem setting is for a simpler model with a rule based deterministic price, these problem instances where small with at most 100 customers. In all the following sections of this study, computational experiments are performed on larger 200-customer instances. In Section 6.3, evaluate the changes in routes when parameters in the choice model change, reflection variation in the preferences of CDs.

6.1 Performance

ALNS is a well-known metaheuristic method that finds good feasible solutions for various combinatorial optimization problems. However, metaheuristics do not provide any guarantee on their performance or the quality of the solutions. Hence, it is necessary to test and compare the solutions with existing implementations in the literature to have confidence in using ALNS. In the following two subsections, we show a comparison with a similar problem in the literature, and then, we show the stability of ALNS when applied several times.

6.1.1 Simple case

To evaluate the quality of the solutions obtained by the ALNS algorithm, we start by comparing a much simpler version of our setting, introduced by Torres et al. (2022b). Unlike this paper, in that setting, the price of routes follows deterministic rules and can be calculated easily with the length of the routes. In addition, the supply of CDs is modeled with a binomial distribution instead of a choice model like in our setting. However, to compare we replace the choice model with the binomial distribution and use the same deterministic rule for the price of each route. The solution method proposed by Torres et al. (2022b) was an exact branch-and-price algorithm with a column generation heuristic. Exact algorithms (e.g., branch-and-price) work well for small instances (e.g., with 25 customers) but fail to scale up for larger instances (e.g., with more than 100 customers). Hence, the comparison is only for small instances with 25, 50 and 100 customers.

Table 2 compares the branch-and-price algorithm, i.e., B&P, and the column generation algorithm, i.e., C-Gen, proposed by Torres et al. (2022b) with the ALNS algorithm presented in this paper.

The first column identifies the group of instances and the second reports the number of customers. Next, we present the exact B&P results, first, the lower bound, i.e., **LB**, then, the solution time in seconds, i.e., **T(s)**. Afterwards, we show the performance of C-Gen reported, the first column is the solution value, i.e., **Sol**, followed by the time and the average gap from the lower bound as a percentage, i.e., **Gap(%)**. The final columns present the results of the ALNS heuristic that we implemented. Since ALNS is a randomized algorithm, we executed the procedure 5 times and we report the results of the best solution out of 5 runs. The first column shows the average of the best solutions found in 5 runs for all instances, the next column shows the average total time of 5 runs, then, the best solution is compared with the lower bound of B&P, and the last column shows the deviation as a percentage, i.e., **Dev(%)**, of the solution values of ALNS with respect to the solution values of C-Gen.

Table 2: ALNS performance

		B&P		C-Gen			ALNS			
Ins	N	**LB**	**T(s)**	**Sol**	**T(s)**	**Gap(%)**	**Sol**	**T(s)**	**Gap(%)**	**Dev(%)**
C1	25	434.85	355	435.92	8	0.27	440.0	40	1.17	0.94
	50	817.77	6604	828.1	135	1.28	845.07	90	3.23	2.05
	100	1776.5	10394	1797.3	450	1.18	1803.21	225	1.48	0.33
R1	25	652.0	102	652.76	8	0.14	663.02	45	1.69	1.57
	50	1228.76	6511	1254.0	141	2.26	1250.74	120	1.79	**-0.26**
	100	2096.51	9145	2198.2	2501	5.61	2188.77	330	4.40	**-0.43**
RC1	25	618.12	32	618.8	3	0.01	636.23	45	2.93	2.4
	50	1161.3	6573	1193.6	44	2.53	1195.0	100	2.91	0.14
	100	2212.0	10800	2376.6	977	7.38	2318.92	335	4.83	**-2.42**

We show that ALNS is much faster than C-Gen and it provides better average solution values for the larger 100-customer instances. On average, ALNS terminates within 5 minutes for 5 runs, while C-Gen can take up to 41 minutes on average for the R1 instances. However, the smaller 25-customer instances are solved faster with C-Gen and the gaps are better. In practice, since platforms can have hundreds or thousands of delivery requests that need to be fulfilled, a method that can solve large instances quickly is desired. The results presented in Table 2 clearly show that ALNS performs better when applied to solve larger instances, whereas the solutions provided by C-Gen tend to decrease in quality on these instances. Recall that the bounds produced by B&P are not exact solutions and could be arbitrarily weak for 100-customer instances. Yet, ALNS still provides solutions that are less than 5% from these weak bounds.

6.2 Base case and stability of ALNS

In this subsection, we use ALNS to solve the full problem described in this paper, including the complex relationships of routing and pricing decisions with the choice model. We consider a base case with 200-customer instances (i.e., C2, RC2 and R2). Table 3 shows the parameters that are used for the base case. The capacity parameters (i.e., Q and Q') are the capacity to fit packages in each vehicle type. Each customer has a predetermined value for the demand which is specified in the Solomon instances. The penalty $\aleph$ is the increase cost of using a PD in the second stage to fulfill rejected routes. Next, the choice model parameters that influence the choice of CDs.

Parameters	**Description**	**Value**
Q	PD-capacity	200
Q'	CD-capacity	100
$\aleph$	Penalty for recourse	2.0
β_1	Route length parameter	-0.5
β_2	Load Parameter	-0.1
β_3	Stops parameter	-1.0
β_4	Location parameter	-0.001
β_c	Price parameter	1.0

Table 3: Parameters for the base case

ALNS is a stochastic algorithm, and thus, can produce different solutions every time it is executed. Hence, to evaluate the stability of the solution values, we test ALNS by running the algorithm multiple times on each 200-customer instance and observing the variation of each result.

In Table 4, we show the results for 20 runs of the ALNS algorithm for each instance group, i.e., C2, RC2 and R2, with 200 customers. The first column shows the instance group. The second column shows the specific instance (i.e., there are 10 instances for each group). The expected cost is shown in the following 4 columns. First, we report the mean value, max value, min value, and the standard deviation (i.e., σ) respectively. The mean value shows the average value of the objective function for all 20 runs of ALNS. The max value shows the worst solution that was generated and the min value shows the best solution generated.

For all the instance groups (i.e., C2, RC2, and R2) we can observe the stability of the ALNS algorithm for the application of this paper. The solution values do not vary in any significant way as it is shown by the small values of the standard deviation with respect to the values of the solution. Additionally, the worst solutions found are not far away from the best solution found, thus, further showing the robustness of ALNS to find consistent solutions. The largest standard deviation is for instance RC2-7 with $\sigma = 32.00$ and the smallest is of $\sigma = 1.50$ for the cluster instance C2-1.

Ins	i	Mean	Max	Min	σ
C2	1	2667.47	2670.09	2666.31	1.50
	2	2649.39	2665.57	2639.71	11.20
	3	2646.08	2656.74	2625.97	13.70
	4	2606.38	2611.60	2599.69	5.19
	5	2663.17	2666.50	2658.98	3.14
	6	2664.46	2671.85	2657.01	6.07
	7	2672.73	2708.58	2657.23	24.06
	8	2637.26	2641.56	2633.36	3.63
	9	2633.14	2646.18	2618.81	14.72
	10	2615.23	2643.34	2598.92	19.42
RC2	1	2851.48	2868.25	2827.16	17.56
	2	2772.97	2783.02	2762.11	10.93
	3	2703.49	2719.47	2688.94	12.95
	4	2688.18	2724.18	2663.15	25.71
	5	2803.05	2830.76	2787.79	19.04
	6	2842.92	2870.09	2831.01	18.24
	7	2785.19	2819.07	2747.85	32.00
	8	2758.10	2780.17	2746.18	15.10
	9	2753.43	2765.36	2735.90	12.65
	10	2727.12	2735.18	2718.39	6.88
R2	1	3148.72	3166.17	3134.05	14.84
	2	2986.80	3018.11	2955.31	26.51
	3	2851.41	2865.78	2827.31	16.98
	4	2777.28	2784.30	2771.91	5.48
	5	3011.53	3040.46	3000.70	19.35
	6	2895.31	2922.96	2881.96	19.25
	7	2798.12	2809.91	2787.27	10.56
	8	2755.95	2771.68	2730.34	17.87
	9	2939.02	2949.16	2929.22	8.45
	10	2862.47	2885.78	2844.21	17.36

Table 4: Twenty runs of ALNS for each instance

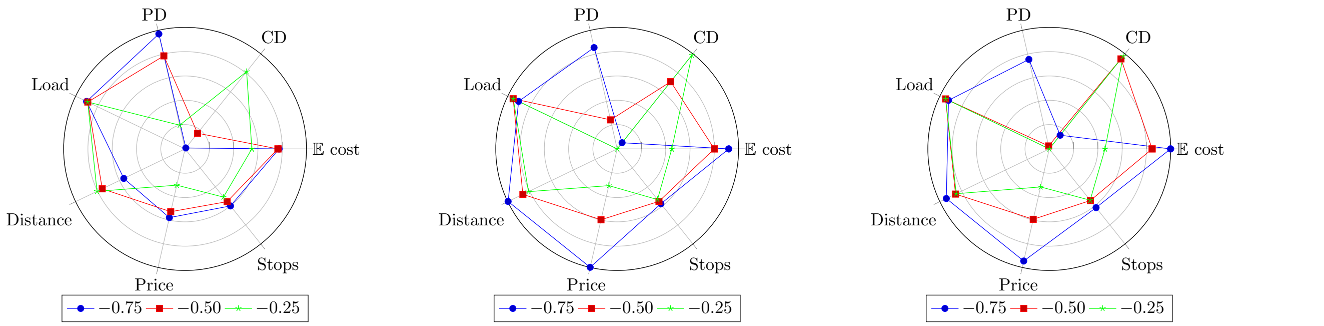
For all the instance groups (i.e., C2, RC2, and R2) we can observe the stability of the ALNS algorithm for the application of this paper. The solution values do not vary in any significant way as it is shown by the small values of the standard deviation with respect to the values of the solution. Additionally, the worst solutions found are not far away from the best solution found, thus, further showing the robustness of ALNS to find consistent solutions. The largest standard deviation is for instance RC2-7 with $\sigma = 32.00$ and the smallest is of $\sigma = 1.50$ for the cluster instance C2-1.

6.3 How do routes change if CD-preferences are different ?

In this subsection, we look at the influence of the parameters of the choice model on the solutions of C2, R2 and CR2 instances. We start with the distance, load, stops, and location.

6.3.1 Distance

The distance parameter (i.e., β_1) shows the willingness to pay (i.e., be compensated) of CDs for an extra unit of distance (e.g., km). We change the value of the parameter to -0.25, -0.50 and -0.75. In Figure 7, we show the changes in the average expected cost, average CDs, average PDs, average load, average distance, average price, and average stops for the main instance groups. The axis values in Figure 7 start from 0, to the maximum value for all instances. The results for C2, RC2, and R2 are presented in subfigures, 7a, 7b and 7c respectively.



(a) C2 (b) RC2 (c) R2

Figure 7: Results on C2, RC2 and R2 instances for β_1 = -0.25, -0.5, and -0.75

We observe in Figure 7 that the average load, stops and distance per route do not change significantly while changing the distance parameter (i.e., β_1). Conversely, major changes are visible on the number of CDs and PDs used in a solution for all instances. As the parameter decreases (i.e., $\beta_1 = -0.75$) the fleet composition shifts to be more dependent on PDs. Larger values of the parameter cause a increase of the participation of CDs. This result shows that the properties of CD-routes do not change much but rather the participation of CDs changes.

6.3.2 Load and stops

The load is the total amount of space occupied in the CD-vehicle, and the stops are the total number of deliveries made in a CD-route. In Figure 8, we show the results by changing each parameter and comparing it with the average number of CD-routes, the average load and the average stops per route for each instance group, i.e., C2, RC2 and R2.

The load parameter (i.e., β_2) indicates the sensitivity to changes in the load of a route on route acceptance. In Figure 8a, we show the results for different values of the load parameter (i.e.,

$\beta_2 = -0.10, \beta_2 = -0.25, \beta_2 - 0.5)$ for each instance class. On the left axis, we report the average number of CD-routes represented by the vertical bars, while on the right axis, we report the average load per CD-route. Surprisingly, the average load of CD-routes does not change significantly, even for large values of the load parameter. However, the average number of CD-routes in each solution decreases from a maximum of 29.5 for instances R2($\beta_2 = -0.10$) to a minimum of 0 CDs for instances C2($\beta_2 = -0.50$).

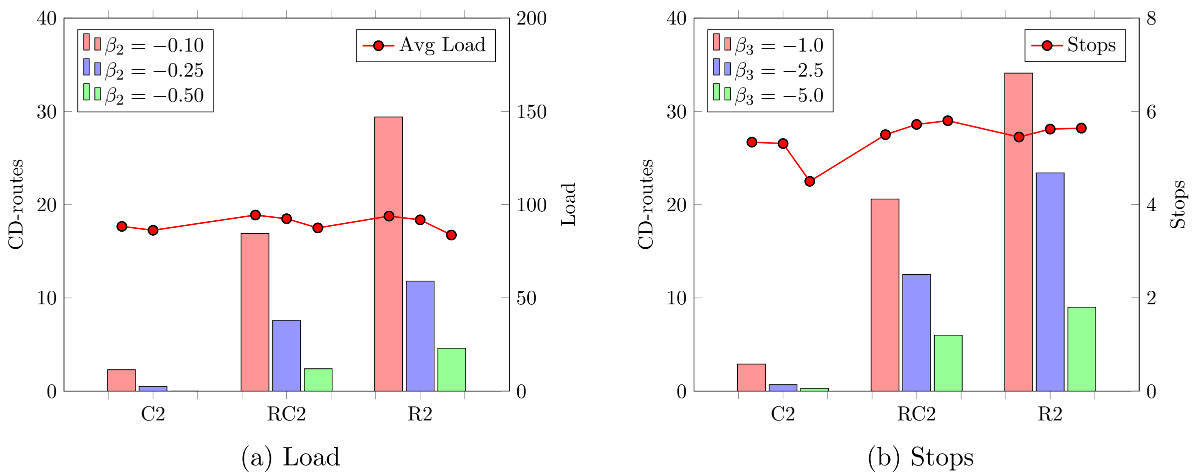
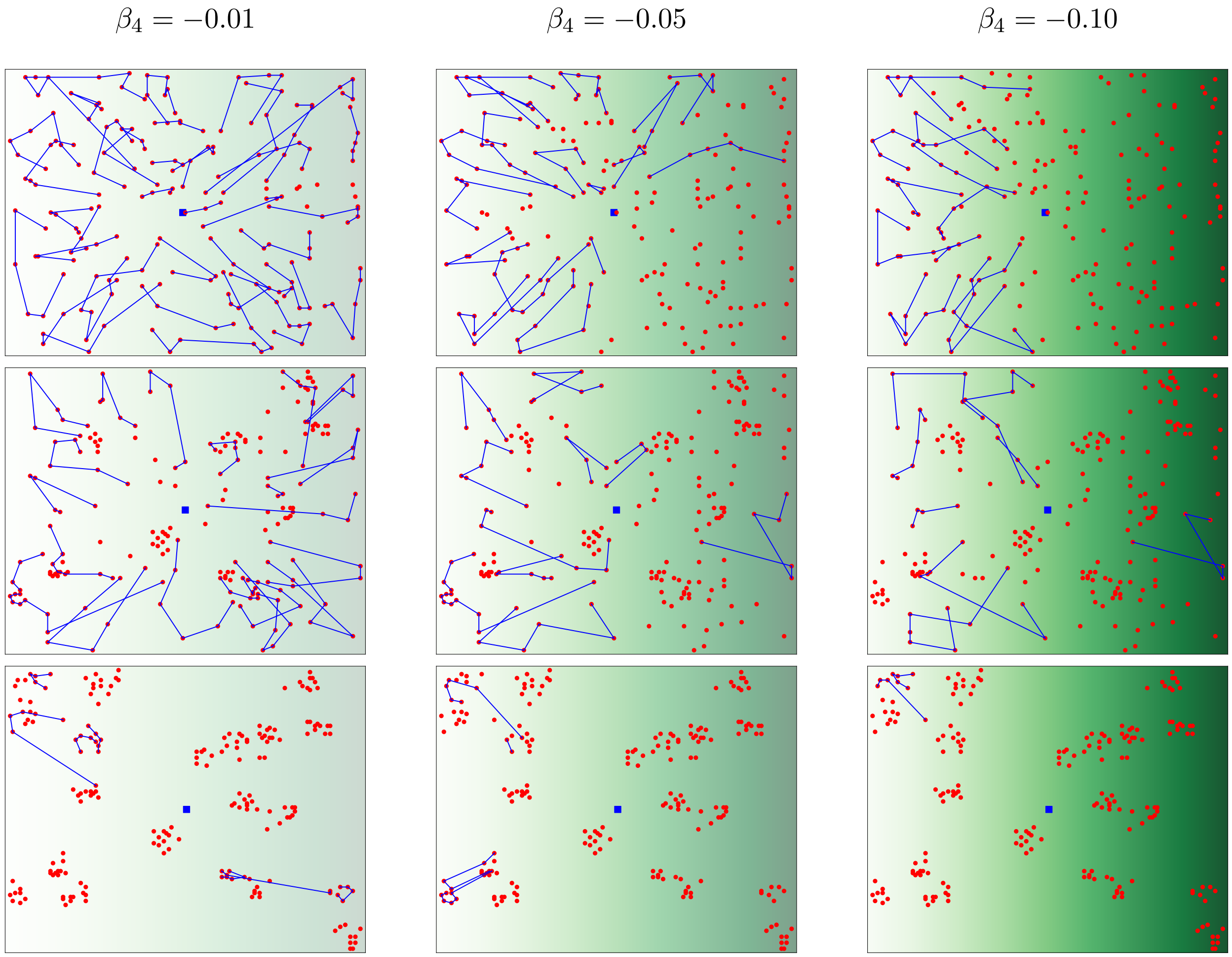


Figure 8: Average CD-routes vs Average load for different values of the load parameter

The stops parameter (i.e., β_3) represents the weight of the total number of stops in a route in the decision of CDs. Similarly, to load and distance, the number of stops does not decrease as the corresponding parameter increases in significance. Rather, the number of CD-routes decreases. In Figure 8b, we show the changes in the "stops parameter" and how the number of CD-routes are decreased while the average number of stops per route stays relatively stable.

6.3.3 Location

The choice model includes the location parameter (i.e., β_4) that considers the negative measure of the location of customers in the utility function of CDs. For simplicity, we have considered that the location penalty of each customer is directly linked to the x-coordinate of the customers location. In practice, any value can be assigned to a customer based on their location and it does not affect the implementation of ALNS. Hence, the location parameter multiplies the x-coordinate of all customers in a route. As a consequence, the further the customers location is to the right (i.e., east of the city), the less desirable the location is to visit for a CD. Figure 9 displays the solutions of chosen instances with only CD-routes. The top row are random instances, the middle row are random-clustered instances and the last row are the clustered instances. We show that as the location parameter decreases, the solution of the problem avoids creating CD-routes in the "bad" area of the city due to the high probability of route rejection. In Figure 9c, almost no routes are available at the right side of the depot, Figure 9a shows a that CD-routes are indifferent of the location, while Figure 9b shows a mix between the two.



(a) Low (b) Medium (c) High

Figure 9: CD-routes for random, random-clustered and clustered instances with different location parameters

6.3.4 Price

The price parameter, i.e., β_c, of the logit model shows the sensitivity of CDs to the compensation of routes. Higher values of the price parameter increase the probability of route acceptance. In Figure 10, we present the average number of CD-routes in solutions for C2, RC2 and R2 instances. We can see a significant change in the number of CD-routes in the solution as the parameter changes values. The R and RC2 instances reach a peak (i.e., maximum number of routes) soon after the parameter is equal to 1, while C2 instances reach a peak slower after the price parameter is equal to 1.55.

Figure 10 indicates that the sensitivity of the fleet of CD to the price of the routes can have different peaks based on the structure of the instance (e.g., clustered).

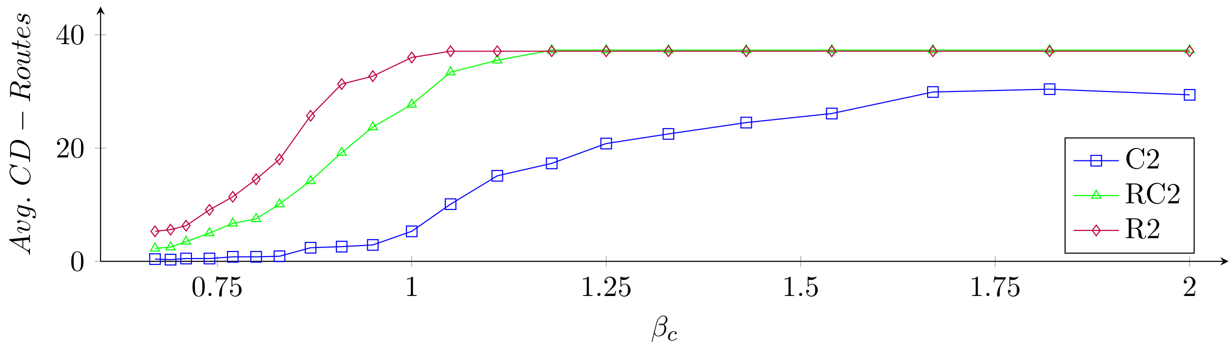


Figure 10: Sensitivity of β_c with Avg. CD-Routes

7 Strategic objectives

In previous sections, it is assumed that the CS-platform wants to minimize the total expected costs. However, minimizing the expected cost is a myopic objective that seeks to minimize the immediate cost at the current day and it does not consider the long-term strategic objectives of the CS-platform. For instance, the long-term goals can be to achieve key levels of participation or establish a fair and consistent compensation to drivers. In fact, CDs are independent agents that can decide to work or not, considering their satisfaction with the CS-platform is just as important as the cost reduction.

7.1 How can participation be increased?

The objective to increase the participation of CDs has to be balanced with minimizing the cost. Increasing the compensation of CDs will increase the probability of route acceptance, as a consequence, the participation will increase. However, a high compensation will result in a higher delivery cost. Hence, in order to increase participation by increasing the compensation, we should be willing to tolerate an increase of the expected cost. A trade-off between the price and the cost is adequate. We consider percentage of the price of a route that we are willing to subsidize in order to increase the participation. For example, a 10% discount factor will reduce the price of all CD-routes by 10% in the objective function.

$$\mathscr{Q}(\cdot) = \min_{c_r \in \mathbb{R}} \sum_{r \in \mathcal{H}} P(c_r)\alpha \times c_r + (1 - P(c_r))\aleph d_r \tag{8}$$

Equation (8) shows the discount parameter α deducing the compensation value in the objective function but not the cost of the recourse actions that will need to be done in case the route is rejected.

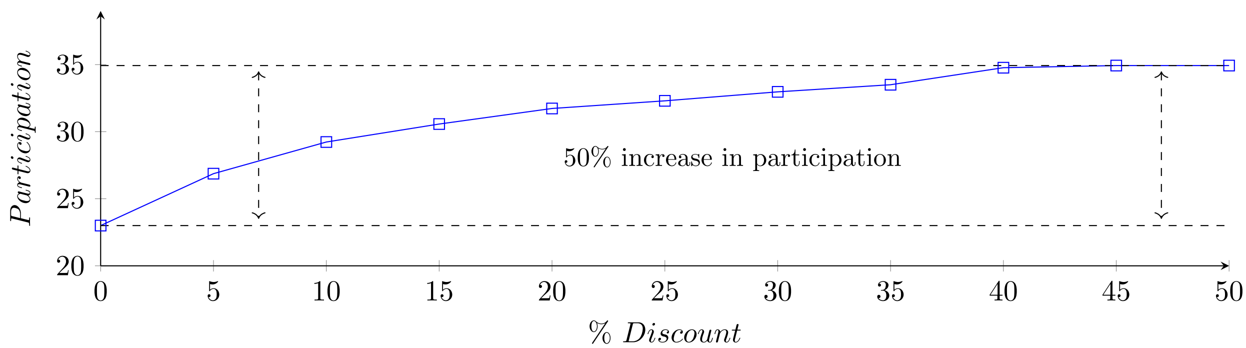


Figure 11: Discounted CD-routes increase participation

Figure 11, shows the participation of CDs is maximized at a discount of 40% to 50%. The average number of CDs goes from 23 to 35, an increase of the participation by 12.

In Figure 12, we can see the trade-off of the two objectives. Increasing participation also leads to an increase of the expected cost. However, a small discount factor of 5% to 10%, increases participation by 7 CDs while only marginally increasing the expected cost.

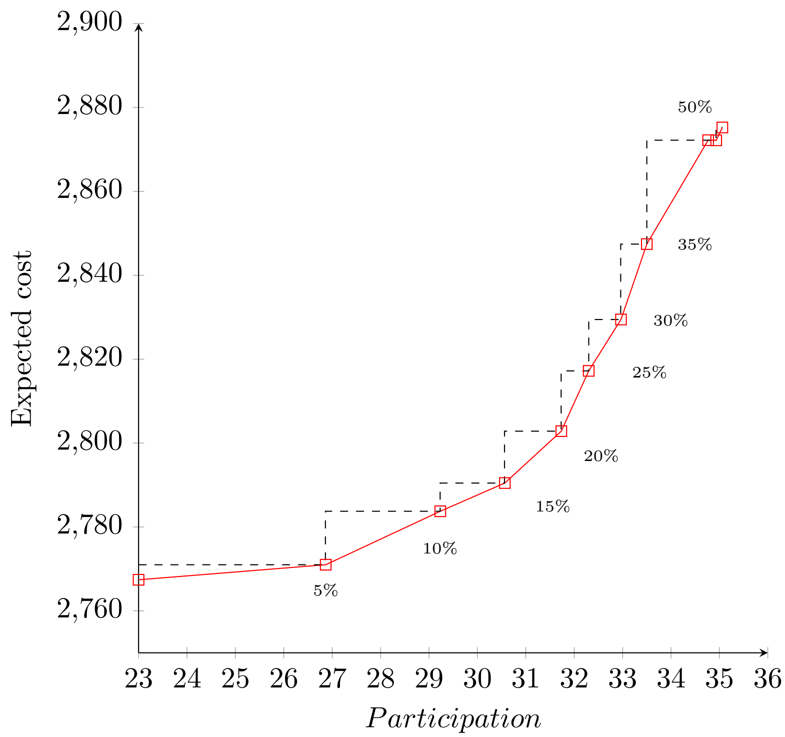


Figure 12: Trade-off between expected cost and participation

Participation can be increased by considering it as part of the objectives and accept increases in the expected cost of the delivery processes. The long-term goal to have a viable CS-platform depends on having a good level of participation, and thus, a small increase in the cost is a fair trade-off to consider.

7.2 What is the average compensation of CDs?

In Figure 13, we show the average compensation of CDs and the participation as the discount factor is increased. The average compensation does not change significantly from \$74.37 per route. The reason for this stability of the compensation is that the objective is still to minimize the expected cost, as a consequence, the price is also minimized. The models has little incentive to increase the price for routes that are highly likely to get accepted, rather there is an incentive for the model to increase the price of routes with a low probability of acceptance to increase the probability of acceptance.

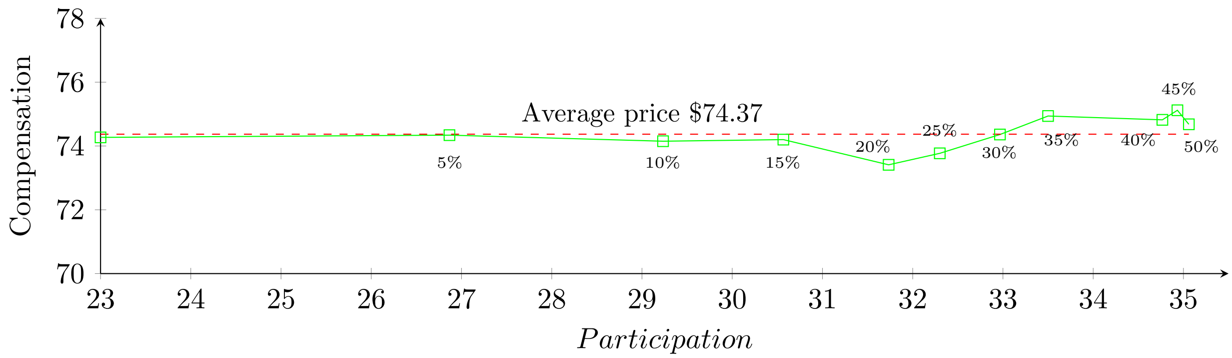


Figure 13: Trade-off between expected cost and participation

The stability of the compensation shows that the participation can be increased with the discount factor but the compensation is standardized.

7.3 Can the fleet of commercial vehicles be replaced by CDs?

The CS-platform has a fleet of commercial vehicles that it can use to deliver requests. In Figure 14, we plot the expected cost by increasing participation of CDs from 0 until all deliveries are entirely fulfilled by CDs. For C2 instances, the use of CDs decreases the cost at around a participation of 5 CDs, as the participation increases, the expected cost is higher with only a fleet of CDs. Conversely, for instances R2 and RC2, the cost is lower when we replace the entire fleet with CDs.

This result shows that, for some instances where delivery requests are clustered together, large commercial vehicles are more cost effective. Using a fleet of CDs can increase the cost instead of decreasing the cost. However, for instances where delivery requests are scattered, the flexibility of the crowd-shipping fleet causes the biggest cost reduction. In fact, Figure 14, shows that the

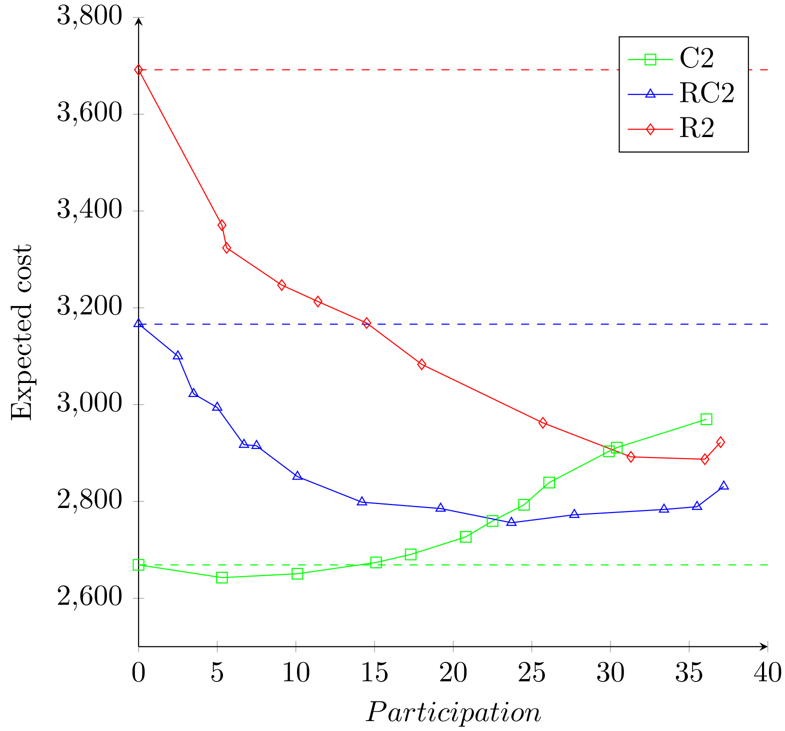


Figure 14: Replacing the fleet with CDs

solution with only CDs is more robust and the expected cost has less variation than a fleet of commercial vehicles. With commercial vehicles the expected cost ranges from a minimum of 2668 for C2 instances, to a maximum of 3691 for R2 instances. On the other hand, with only a fleet of CDs, the expected cost ranges from a minimum of 2831 for RC2 instances to a maximum of 2969 for C2 instances.

8 Conclusions and future research

Crowd-shipping for the "last-mile" requires the solution of routing problems while balancing the supply of crowd-drivers through pricing. In this paper, we introduced a framework that routes and prices together in a single model. The probability of route acceptance by the CDs was modeled with an econometric model that considers several route attributes (i.e., stops, load, distance, location and price) to determine the probability of acceptance. The solution complexity of the problem consist in solving a heterogeneous vehicle routing problem (which is a known NP-hard problem) with a pricing problem for each route. We developed a solution method based on a two-stage stochastic formulation and solved the pricing in the ALNS algorithm while simultaneously creating routes. We integrated an econometric model with an optimization model which represents the uncertainty of route acceptance more accurately, instead of an abstract scalar with no interpretation. The econometric model allowed us to interpret the predictions by changing the parameters for each route attribute.

The sensitivity analysis of the econometric model showed that the behaviour of drivers changes based on the attributes of routes. When crowd-drivers are more sensitive to the attributes of a route, the number of CDs in a solution decreases due to the drop in probability of acceptance. However, CD-route properties remain similar (e.g., the number of stops, load, price and distance). However, when CDs are more sensitive to the location attribute, we can see that less routes are present at those locations where there is a penalty for visiting customers in undesirable locations. This result shows the consistency of the framework that accurately models the preference of CDs to avoid a certain area. CD-routes that visit these undesirable areas will be less likely to be accepted and thus, they will have an expensive expected cost.

We offered managerial insights about managing a fleet of vehicles. In particular, we show that when we increase participation until there are no more PD in a solution, the variation in the expected cost in all scenarios is minimal in comparison with a fleet of only PD. Indeed, the expected cost is minimized when a mixed fleet is allowed, but the CD only solution is more stable.

We demonstrates that the participation of CDs can be increased by introducing a discount of the price of CDs in the objective function. A large increase of 7 CD from the optimal solution can be achieved with less than % 1 of an increase in cost. Finally, we show that the compensation of CDs remains stable even when the participation increases.

Future research needs to be done by considering different variants of the VRP and pricing routes for those variants, e.g., pick-up and delivery, multi-depot VRP, open VRPs etc. Another interesting research avenue is to continue to integrate econometric models in vehicle routing problems to model the uncertainty. The benefit of using econometric models for prediction, rather than machine learning, is that it allows us to understand and interpret the predictions as we showed in this paper.

Acknowledgments This research was partially funded by the Canadian Natural Sciences and Engineering Research Council (NSERC) under its Discovery Grants research program, as well as the Canada Research Chair in Stochastic Optimization of Transport and Logistics Systems, by Polytechnique Montréal, by the Ecuadorian Secretaría de Educación Superior, Ciencia, Tecnología e Inovación (SENESCYT) under its international scholarships program, and by Innosuisse (Swiss Innovation Agency).

References

S. Afifi, D.-C. Dang, and A. Moukrim. A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints. In *International Conference on Learning and Intelligent Optimization*, pages 259–265. Springer, 2013.

S. Akpinar. Hybrid large neighbourhood search algorithm for capacitated vehicle routing problem. *Expert Systems with Applications*, 61:28–38, 2016.

A. Alnaggar, F. Gzara, and J. H. Bookbinder. Crowdsourced delivery: A review of platforms and academic literature. *Omega*, 98:102139, 2021.

A. Alnaggar, F. Gzara, and J. H. Bookbinder. Compensation guarantees in crowdsourced delivery: Impact on platform and driver welfare. *Omega*, 122:102965, 2024.

AmazonFlex. AmazonFlex, 2021a. URL `https://flex.amazon.ca`.

AmazonFlex. AmazonFlex, 2021b. URL `https://flex.amazon.co.uk/earnings`.

C. Archetti, M. Savelsbergh, and M. G. Speranza. The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2):472–480, 2016.

C. Archetti, F. Guerriero, and G. Macrina. The online vehicle routing problem with occasional drivers. *Computers & Operations Research*, 127:105144, 2021.

M. Barbosa, J. P. Pedroso, and A. Viana. A data-driven compensation scheme for last-mile delivery with crowdsourcing. *Computers & Operations Research*, 150:106059, 2023.

L. Dahle, H. Andersson, and M. Christiansen. The vehicle routing problem with dynamic occasional drivers. In *International Conference on Computational Logistics*, pages 49–63. Springer, 2017.

L. Dahle, H. Andersson, M. Christiansen, and M. G. Speranza. The pickup and delivery problem with time windows and occasional drivers. *Computers & Operations Research*, 109:122–133, 2019.

I. Dayarian and M. Savelsbergh. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *Production and Operations Management*, 29(9):2153–2174, 2020.

I. Dayarian, T. G. Crainic, M. Gendreau, and W. Rei. An adaptive large-neighborhood search heuristic for a multi-period vehicle routing problem. *Transportation Research Part E: Logistics and Transportation Review*, 95:95–123, 2016.

K. A. Dowsland and J. Thompson. Simulated annealing. *Handbook of natural computing*, pages 1623–1655, 2012.

K. Gdowska, A. Viana, and J. P. Pedroso. Stochastic last-mile delivery with crowdshipping. *Transportation research procedia*, 30:90–100, 2018.

T. Haering, R. Legault, F. Torres, I. Ljubic, and M. Bierlaire. Exact algorithms for continuous pricing with advanced discrete choice demand models. Technical report, Technical Report TRANSPOR 231211, Transport and Mobility Laboratory, École . . . , 2023.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *science*, 220 (4598):671–680, 1983.

K. U. Kızıl and B. Yıldız. Public transport-based crowd-shipping with backup transfers. *Transportation Science*, 57(1):174–196, 2023.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. The fleet size and mix pollution-routing problem. *Transportation Research Part B: Methodological*, 70:239–254, 2014.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. A hybrid evolutionary algorithm for heterogeneous fleet vehicle routing problems with time windows. *Computers & Operations Research*, 64:11–27, 2015.

J. E. Korsvik, K. Fagerholt, and G. Laporte. A large neighbourhood search heuristic for ship routing and scheduling with split loads. *Computers & Operations Research*, 38(2):474–483, 2011.

T. V. Le, S. V. Ukkusuri, J. Xue, and T. Van Woensel. Designing pricing and compensation schemes by integrating matching and routing models for crowd-shipping systems. *Transportation Research Part E: Logistics and Transportation Review*, 149:102209, 2021.

G. Macrina and F. Guerriero. The green vehicle routing problem with occasional drivers. In *New Trends in Emerging Complex Real Life Problems*, pages 357–366. Springer, 2018.

G. Macrina, L. D. P. Pugliese, F. Guerriero, and D. Laganà. The vehicle routing problem with occasional drivers and time windows. In *International Conference on Optimization and Decision Science*, pages 577–587. Springer, 2017.

G. Macrina, L. D. P. Pugliese, F. Guerriero, and G. Laporte. Crowd-shipping with time windows and transshipment nodes. *Computers & Operations Research*, 113:104806, 2020.

S. S. Mohri, H. Ghaderi, N. Nassir, and R. G. Thompson. Crowdshipping for sustainable urban logistics: A systematic review of the literature. *Transportation Research Part E: Logistics and Transportation Review*, 178:103289, 2023.

K. Mousavi, M. Bodur, and M. J. Roorda. Stochastic last-mile delivery with crowd-shipping and mobile depots. *Transportation Science*, 2021.

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer, 1999.

D. C. Paraskevopoulos, P. P. Repoussis, C. D. Tarantilis, G. Ioannou, and G. P. Prastacos. A reactive variable neighborhood tabu search for the heterogeneous fleet vehicle routing problem with time windows. *Journal of Heuristics*, 14(5):425–455, 2008.

D. Pisinger and S. Ropke. A general heuristic for vehicle routing problems. *Computers & operations research*, 34(8):2403–2435, 2007.

D. Pisinger and S. Ropke. Large neighborhood search. In *Handbook of metaheuristics*, pages 99–127. Springer, 2019.

L. D. P. Pugliese, D. Ferone, G. Macrina, P. Festa, and F. Guerriero. The crowd-shipping with penalty cost function and uncertain travel times. *Omega*, 115:102776, 2023.

L. M. Ricard and M. Bierlaire. 50 years of behavioral models for transportation and logistics. *TranspOR Report*, 2024.

P. Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *International conference on principles and practice of constraint programming*, pages 417–431. Springer, 1998.

M. Silva, J. P. Pedroso, and A. Viana. Stochastic crowd shipping last-mile delivery with correlated marginals and probabilistic constraints. *European Journal of Operational Research*, 307(1):249–265, 2023.

J. Skålnes, L. Dahle, H. Andersson, M. Christiansen, and L. M. Hvattum. The multistage stochastic vehicle routing problem with dynamic occasional drivers. In *International Conference on Computational Logistics*, pages 261–276. Springer, 2020.

P. Stokkink and N. Geroliminis. A continuum approximation approach to the depot location problem in a crowd-shipping system. *Transportation Research Part E: Logistics and Transportation Review*, 176:103207, 2023.

P. Stokkink, J.-F. Cordeau, and N. Geroliminis. A column and row generation approach to the crowd-shipping problem with transfers. *Omega*, page 103134, 2024.

Y. Tao, H. Zhuo, and X. Lai. The pickup and delivery problem with multiple depots and dynamic occasional drivers in crowdshipping delivery. *Computers & Industrial Engineering*, 182:109440, 2023.

F. Torres, M. Gendreau, and W. Rei. Crowdshipping: An open vrp variant with stochastic destinations. *Transportation Research Part C: Emerging Technologies*, 140:103677, 2022a.

F. Torres, M. Gendreau, and W. Rei. Vehicle routing with stochastic supply of crowd vehicles and time windows. *Transportation Science*, 56(3):631–653, 2022b.

P. J. Van Laarhoven and E. H. Aarts. Simulated annealing. In *Simulated annealing: Theory and applications*, pages 7–15. Springer, 1987.

F. Y. Vincent, P. Jodiawan, and A. P. Redi. Crowd-shipping problem with time windows, transshipment nodes, and delivery options. *Transportation Research Part E: Logistics and Transportation Review*, 157:102545, 2022.

F. Y. Vincent, G. Aloina, P. Jodiawan, A. Gunawan, and T.-C. Huang. The vehicle routing problem with simultaneous pickup and delivery and occasional drivers. *Expert Systems with Applications*, 214:119118, 2023.

D. Yang, M. F. Hyland, and R. Jayakrishnan. Tackling the crowdsourced shared-trip delivery problem at scale with a novel decomposition heuristic. *Transportation Research Part E: Logistics and Transportation Review*, 188:103633, 2024.

Z. Zhang and F. Zhang. Optimal operation strategies of an urban crowdshipping platform in asset-light, asset-medium, or asset-heavy business format. *Transportation Research Part B: Methodological*, page 102992, 2024.

A Proof of Proposition 1

The second stage expected cost is given by the following formula:

$$\frac{e^V}{e^V+1} \cdot \sum_{r \in \mathcal{H}} \left(\frac{V - x_r}{\beta_c} \right) + \left(1 - \frac{e^V}{e^V+1} \right) \cdot \sum_{r \in \mathcal{H}} \aleph d_r$$

By simplifying we obtain the equivalent formula:

$$\frac{e^V \cdot \left(\frac{|\mathcal{H}|}{\beta_c} V - \sum_{r \in \mathcal{H}} \frac{x_r}{\beta_c} - \sum_{r \in \mathcal{H}} \aleph d_r \right)}{e^V + 1} + \sum_{r \in \mathcal{H}} \aleph d_r$$

The derivative of the cost function is the following:

$$\frac{\mathrm{e}^V \cdot \left(\frac{|\mathcal{H}|}{\beta_c} \mathrm{e}^V + \frac{|\mathcal{H}|}{\beta_c} V - \sum_{r \in \mathcal{H}} \aleph d_r - \sum_{r \in \mathcal{H}} \frac{x_r}{\beta_c} + \frac{|\mathcal{H}|}{\beta_c} \right)}{\left(\mathrm{e}^V + 1 \right)^2} \tag{9}$$

By observation, we can see that the derivative will equal zero if the following equation is true:

$$\left(\frac{|\mathcal{H}|}{\beta_c} \mathrm{e}^V + \frac{|\mathcal{H}|}{\beta_c} V - \sum_{r \in \mathcal{H}} \aleph d_r - \sum_{r \in \mathcal{H}} \frac{x_r}{\beta_c} + \frac{|\mathcal{H}|}{\beta_c} \right) = 0$$

We simplify the equation to obtain the following:

$$\mathrm{e}^V + V = \frac{\sum_{r \in \mathcal{H}} \frac{x_r}{\beta_c} + \sum_{r \in \mathcal{H}} \aleph d_r - \frac{|\mathcal{H}|}{\beta_c}}{\frac{|\mathcal{H}|}{\beta_c}}$$

For visualization let ψ be a constant defined as following:

$$\psi = \frac{\sum_{r \in \mathcal{H}} \frac{x_r}{\beta_c} + \sum_{r \in \mathcal{H}} \aleph d_r - \frac{|\mathcal{H}|}{\beta_c}}{\frac{|\mathcal{H}|}{\beta_c}}$$

Therefore $\therefore$

$$\mathrm{e}^V + V = \psi$$

Now, by simplifying further and using the Lambert W function on both sides we get the desired result:

$$V = \psi - W\left(e^{\psi}\right)$$

It is easy to show that the limit of function (9), when the utility approaches minus infinity will be zero. While the limit when the utility approaches plus infinity is equal to the following fraction:

$$\lim_{V \to \infty} (9) = \frac{|\mathcal{H}|}{\beta_c}$$

On the other hand, the value of the objective function when the utility approaches minus infinity is equal to the following:

$$\lim_{V \to -\infty} (2) = \sum_{r \in \mathcal{H}} \aleph d_r$$

Thus, the derivative of the cost function has a unique point at which the solution is zero, and it asymptotically approaches zero only when the utility approaches minus infinity. We now have to proof that the unique point is the minimum of the function and not a maxima nor a saddle point. To achieve this, we look at the change of sign of the derivative right before the critical point and after the critical point.

$$\frac{|\mathcal{H}|}{\beta_c} \left(\mathrm{e}^V + V + 1 \right) - \left(\sum_{r \in \mathcal{H}} \aleph d_r + \sum_{r \in \mathcal{H}} \frac{x_r}{\beta_c} \right) = 0 \tag{10}$$

We can see in equation (10) if we increase the value of the utility at the critical point, the derivative will be positive. Conversely, if we decrease the value of the utility at the critical point the derivative will be negative. Thus, the slope of the function is decreasing before the critical point and is increasing after the critical point, which proves that the critical point is indeed a minimum. □