CIRRELT-2022-03

Crowd-Shipping: Determining the Compensation of Crowd-Drivers with Stochastic Route Acceptance

Fabian Torres
Michel Gendreau
Walter Rei

January 2022

Bureau de Montréal
Université de Montréal
C.P. 6 128, succ. Centre-Ville
Montréal (Québec) H3C 3J7
Tél : 1-514-343-7575
Télécopie : 1-514-343-7121

Bureau de Québec
Université Laval,
2325, rue de la Terrasse
Pavillon Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Crowd-Shipping: Determining the Compensation of Crowd-Drivers with Stochastic Route Acceptance

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

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Mathematics and Industrial Engineering, Polytechnique Montréal

[3] Department of Management and Technology, Université du Québec à Montréal

Abstract. E-commerce continues to grow all over the world. The recent pandemic caused by COVID-19 has increased this trend. Concurrently, crowd-shipping is emerging as a viable solution to fulfill last-mile deliveries, with AmazonFlex taking the lead in implementing such distribution models.

In this paper, we look at a problem of crowd-shipping were a crowd-shipping platform must fulfill

delivery requests from a central depot with a fleet of professional vehicles and a pool of crowddrivers. The latter can accept or reject routes. The probability of route acceptance is dependent on the set of routes that are offered to crowd-drivers. The best compensated route is the most likely to be accepted. We develop a large neighborhood search heuristic to solve this routing problem. To investigate the practical viability of such distribution models, we show the market equilibrium when no fluctuation in supply is considered, versus the market equilibrium when the stochastic route acceptance of crowd-drivers is considered. The best compensation for crowd-drivers that minimizes the total expected cost of the routing problem is determined. We show in our numerical experiments that a 6% cost reduction can be achieved by adjusting the compensation level when we consider stochastic route acceptance.

Keywords: Crowd-shipping, crowd-logistics, crowd drivers, occasional drivers, city logistics, stochastic, Large Neighborhood Search.

Acknowledgements. This research was partially funded by the Natural Sciences and Engineering Research Council of Canada (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.

---

* Corresponding author: fabiantodu@gmail.com

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2022

1 Introduction

E-commerce continues to rise all over the world. People increasingly prefer to stay safe at home and shop online rather than to expose themselves to dangerous pathogens, e.g., COVID-19, by shopping in crowded areas (Bhatti et al., 2020). Concurrently, crowd-shipping is emerging as a new way for distribution platforms to fulfill the increasing demand for home delivery (He et al., 2021). Crowd-shipping platforms can use individuals in society that have a vehicle to deliver packages and compensate them appropriately. Individuals own their vehicles and can work whenever they want. Crowd-shipping has the potential to lower the cost of operation, reduce the environmental impact by using vehicles that are already on the road network, provide a flexible way of making an income with few entry barriers for the unemployed, and simultaneously, provide important home-delivery services in times where infectious diseases are present in society. However, there are a series of challenges that arise from crowd-shipping (Sampaio et al., 2019). Vehicles owned by CDs are not designed to transport parcels, but rather, they are made with the intention of transporting people and for the regular use of individuals. The size, model, and thus, the capacity of the vehicles can vary widely. In addition, items sold by online retailers are heterogeneous, ranging from small objects, e.g., smart phones, pencils and books, to large products, e.g., microwaves or bookshelves. When the packages that need to be delivered are heterogeneous, the capacity of vehicles matters. For instance, multiple notebooks can fit in a bicycle, but multiple microwaves cannot. How can a set of delivery requests be assigned to a crowd-driver without previous knowledge of the available capacity?

Crowd-shipping platforms are being created in large cities to solve these problems, e.g., Amazon Flex (AmazonFlex, 2021). However, crowd-drivers (CD) are free agents and can decide what crowd-shipping platform to work for. They could work for Amazon Flex one day delivering parcels, and the next day, they could work for Uber or Lyft transporting people. They could even decide to work for different platforms at the same time, i.e., delivering packages for multiple platforms or delivering packages while they transport people. Crowd-shipping platforms have to compete with each other and any other crowd-based transportation platforms for the participation of CDs. Thus, the supply of CDs on any given day becomes uncertain.

Consider the market description of the supply and demand of CDs in Figure 1. In any major city, there is a group of people that have vehicles and are willing to participate in the sharing economy by utilizing their vehicle to fulfill transportation requests. There is also a set of platforms that have a set of customers with transportation requests and are interested in employing CDs to satisfy their customers' demands. CDs supply the service of delivery while the platforms have a demand for such service. The demand that platforms have for CDs vary from day to day and is primarily a function of the customers that use the platforms. For instance, sporting events can lead to a surge in demand for Uber drivers, while holidays, e.g., Christmas, can cause an increase in demand for home delivery. If crowd-shipping platforms share some CDs, then the supply of drivers

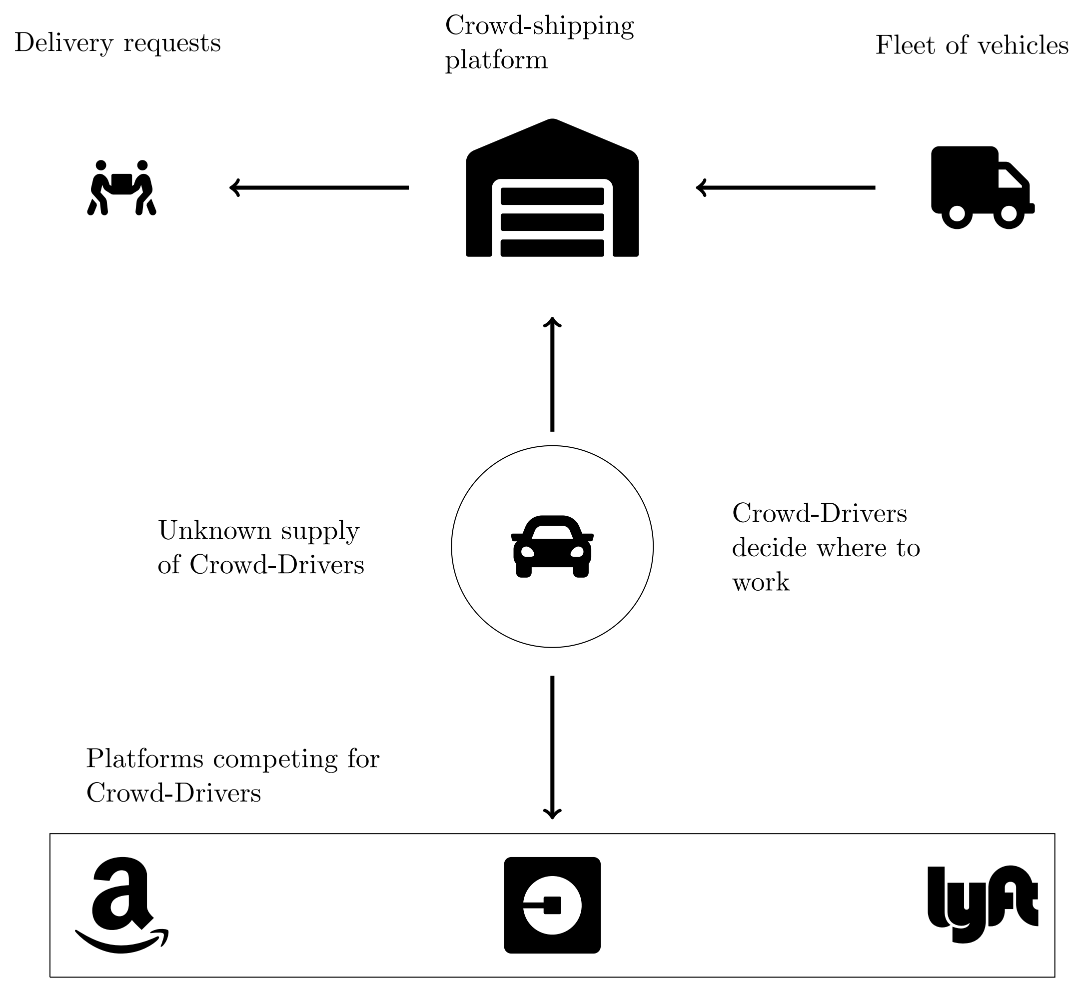

Figure 1: Market description of Crowd-Drivers and competing platforms

will change based on natural market fluctuations.

Crowd-drivers are assumed to be free agents that can decide to work or not on any given day. The personal preferences of CDs for routes will impact the probability of route acceptance. Let H be a set of routes that are offered to CDs. We assume that each CD has a preference ordering for routes in H. Based on her personal preference, a CD will look at the set H and order the set H from most preferred route to least preferred. In theory, CDs could have a different preference ordering, however, a recent study (Asdecker and Zirkelbach, 2020) shows that CDs are primarily motivated by compensation. Thus, we may assume that all CDs prefer higher compensated routes. The probability of route acceptance of a given route, e.g., r, depends on the position of r in the preference ordering of CDs. If CD-route r is better compensated than the other routes, then it will be more likely to be accepted. Hence, the probability of route acceptance is dependent on whether other better compensated routes exist in the list of CD-routes, i.e., H.

In order to increase participation, the crowd-shipping platform has to calibrate the compensation of CDs to the right level. For instance, by lowering the compensation of CDs, the cost of using CDs will naturally decrease. However, few CDs will want to participate for a small compensation and the participation will decrease. Thus, to fulfill delivery requests the platform has to use more professional vehicles (PV), increasing the total cost of operations. Conversely, increasing the compensation of CDs will also increase the participation in the platform, albeit, at a higher cost. The main objective of this paper is to develop a method to set the compensation to a specific level in order to minimize the total expected cost of operations. We assume that information exists about the supply curve of CDs at different compensation levels. The supply curve can be obtained by a market research initiative to find the average supply of CDs at different compensation levels. The demand curve for CDs is obtained by solving a series of vehicle routing problems (VRP) and determining the average number of CDs needed per compensation level to minimize the total expected cost.

In this paper, we consider a setting introduced by Torres et al. (2021a) where a crowd-shipping platform has a set of heterogeneous delivery requests that can be fulfilled by a professional fleet of vehicles, i.e., PVs, owned by the platform, or by a pool of CDs. The products sold online vary from groceries, electronics, books, to different appliances. Some products can require customer signatures, others can have specific time windows for delivery to take place. Failed deliveries can occur when customers are not present or when some unforeseeable event occurs. For example, Amazon Flex requires CDs to return all undelivered packages to the fulfillment center after they have finished their route (AmazonFlex, 2021). A two-stage stochastic model is proposed. In the first stage, routes are created for PVs and a subset of delivery requests are separated by creating CD-routes. In the second stage, the supply of CDs becomes known and recourse actions are taken to complete deliveries. When the supply of CD is less than the number of planned CD-routes, some CD-routes will be left without being fulfilled. PV are then used at a penalty to complete all CD-routes that were left unfulfilled due to the lack of supply.

The contributions of this paper are as follows: we develop a Large Neighborhood Search heuristic (LNS) to solve larger instances of the stochastic VRP variant presented in Torres et al. (2021a), and show that LNS outperforms the column generation heuristic proposed in Torres et al. (2021a). We introduce a procedure within the LNS algorithm that, given a set of routes, optimally assigns them to PVs and CDs to minimize the expected cost of CD-routes based on the preference ordering of CDs. Finally, we derive the best compensation level for CDs that minimizes the total cost of deliveries.

The remainder of this paper is organized as follows. In Section 2, we present a review of the related literature. In Section 3, we indicate how we adapt the model used by Torres et al. (2021a) to the setting of this paper. In Section 4, we describe the LNS heuristic that we use to solve the considered instances. In Section 5, we perform extensive computational experiments and determine the compensation of CDs that minimizes the total routing cost. Finally, in Section 6, we conclude and discuss future research directions.

2 Related work

In Archetti et al. (2016), a deterministic and static problem with occasional drivers was introduced to highlight the difficulties that can arise in crowd-shipping. A subset of delivery requests is separated to be fulfilled by occasional drivers. The remaining subset of delivery requests must be fulfilled by the fleet of PVs. In this setting, the flow of information is not considered. In practice, PVs will be deployed first, then, information about the number of available CDs is reveled. An extension of Archetti et al. (2016) was introduced by Gdowska et al. (2018), where a bi-level method is proposed. The set of delivery requests is separated into two subsets, the first has to be fulfilled by the fleet of PVs, while the second subset will be offered to CDs. However, CDs can reject delivery requests with a given probability. A penalty is considered per delivery if CDs reject the delivery request. Due to the complexity of the problem, only small 15-customer instances are solved with the heuristic proposed.

Multiple deterministic extensions of the work of Archetti et al. (2016) have been proposed (Mancini and Gansterer, 2021; Macrina et al., 2017; Macrina and Guerriero, 2018; Dahle et al., 2019) but few consider stochastic aspects of the problem.

In a recent review by Alnaggar et al. (2021), it is shown that a gap exists between real crowd-shipping platforms and the scientific literature on crowd-shipping. More precisely, in real crowd-shipping platforms, e.g., AmazonFlex (2021), the available capacity of CDs is considered, while in the scientific literature few variants consider the capacity of CDs. Furthermore, real deliveries have a high level of uncertainty due to the preferences of CDs and the possibility of failed deliveries, yet, few stochastic variants exist in the literature that consider these properties.

In Torres et al. (2021a), the authors present a stochastic VRP setting that we extend in this paper. A crowd-shipping platform has a set of heterogeneous delivery requests that must be fulfilled from a central depot by a fleet of PVs and a pool of capacitated CDs. Some items require signatures to be delivered and failed deliveries can happen, in which case, a return trip to the depot is required by CDs. It is assumed that CDs get compensated for the return trip regardless of whether a return trip actually occurs or not. The supply of CDs is considered to be stochastic, thus, the number of CDs available in a given day is unknown beforehand. The following two-stage stochastic formulation is proposed:

$$\min \sum_{r\in\Omega\cup\Omega'} c^r\lambda^r + \mathscr{Q}(\lambda) \tag{1}$$

$$\text{s.t.} \sum_{r\in\Omega\cup\Omega'} a_i^r\lambda^r = 1 \qquad \forall i\in N \tag{2}$$

$$\sum_{r\in\Omega'} \lambda_r \leq M \tag{3}$$

$$\lambda_r \in \{0,1\} \qquad \forall r\in\Omega\cup\Omega'$$

Binary variables λ^r are equal to one if route $r \in \Omega\cup\Omega'$ is performed by a vehicle in the solution, and equal to zero otherwise. The set Ω is the set that contains all feasible PV-routes, and Ω' is the set of all feasible CD-routes. The parameter a_i^r is equal to one if customer $i \in N$ is visited in route $r \in \Omega\cup\Omega'$ and zero otherwise. The set N is the set of all customers. Parameter c^r is the cost of route $r \in \Omega\cup\Omega'$. The value M is the upper bound on the number of CDs that can be available in a solution and it can be set to the total number of CDs in the pool. The second-stage problem is described by $\mathscr{Q}(\lambda)$. The objective (1) seeks to minimize the total cost of operation. Inequalities (2) state that all customers must be visited only once by one vehicle, and constraint (3) guarantee that CD-routes do not exceed an upper bound.

In the first stage, the set of customers are divided into two sets by the set partitioning model. The first set of customers must be visited by the fleet of PVs, and the second set of customers will possibly be visited by CDs. Routes are created in the model for both PVs and CDs in this stage. CDs are not forced to follow CD-routes, rather the routes are created to establish the compensation that needs to be paid to CDs and to bundle customers for CD-routes. The PVs start routes immediately, while CD routes are left for the second stage; these routing decisions are made with knowledge of the expected supply of CDs that will participate later in the day. A discrete probability function is assumed to be available to predict the number of CDs that will participate.

In the second stage, the number of CDs is revealed and CDs perform the routes that are assigned to them. Routes that are not fulfilled due to the lack of supply of CDs need to be performed by PVs at a penalty. Thus, creating routes for CDs in the first stage has the possible consequence of expensive recourse actions if the supply is low. Here, the probability of rejection of CD routes is considered dependent on other routes. In Torres et al. (2021a), the authors develop an exact branch-and-price algorithm to solve some instances with at most 100 customers. A column generation heuristic is proposed to solve all 100-customer instances. However, in Torres et al. (2021a), the increase of participation when the compensation increases is not addressed.

Table 1 shows the summary of the stochastic variants that were identified in the literature. In the first column, we show which studies consider failed delivery attempts, i.e., if some actions or plan is devised in the case that CDs are unable to deliver a package, e.g., when customers are absent. Next column, i.e., CD-Capacity, shows stochastic variants that consider a one-dimensional value to represent the free space in CD-vehicles. The following column, i.e., Probability, specifies the way in which the probability of route acceptance is formulated. When a delivery task is assigned to a CDs the probability of acceptance should be dependent on other delivery tasks offered to CDs.

Study	**Failed deliveries**	**CD-Capacity**	**Probability**	**Recourse**	**Market Eq.**
Dayarian and Savelsbergh (2020)	NO	NO	Independent	NO	NO
Gdowska et al. (2018)	NO	NO	Independent	NO	NO
Torres et al. (2021a)	YES	YES	Dependent	YES	NO
Dahle et al. (2017)	NO	NO	Independent	NO	NO
Torres et al. (2021b)	NO	YES	Dependent	YES	NO
Mousavi et al. (2021)	NO	NO	Independent	NO	NO
Skålnes et al. (2020)	NO	NO	Independent	NO	NO
This paper	YES	YES	Dependent	YES	YES

Table 1: Summary of stochastic variants

Next, the column Recourse indicates whether a recourse action is defined in the case that the availability of CDs is less than planned. Most stochastic variants in the literature consider some abstract penalty that represents an unspecified future action. Only Torres et al. (2021a) and Torres et al. (2021b) describe the recourse actions as using an additional PV to fulfill deliveries at a penalty.

The last column, i.e., Market Eq., shows that no stochastic variant in the literature considers the market equilibrium between the supply curve of CDs and the demand curve for CDs in a setting where route acceptance is stochastic.

In this paper, we extend the study done in Torres et al. (2021a) with the objective of finding the best compensation level for CDs that minimizes the total delivery cost, considering stochastic route acceptance and the supply and demand curves.

3 Problem description

We consider the setting presented in Torres et al. (2021a), that was briefly described in Sections 1 and 2. In this paper, CDs are rational agents that prefer routes that are better compensated. Thus, we assume that, given a set of routes to choose from, CDs will order the available routes from the most preferred to the least preferred; this is commonly referred to as a preference ordering. Potentially, CDs could have different preference orderings, e.g., a small number of CDs could prefer environmentally friendly routes. However, in Asdecker and Zirkelbach (2020), the authors show that CDs are primarily motivated by compensation. Hence, we assume homogeneity in the preference orderings of all CDs.

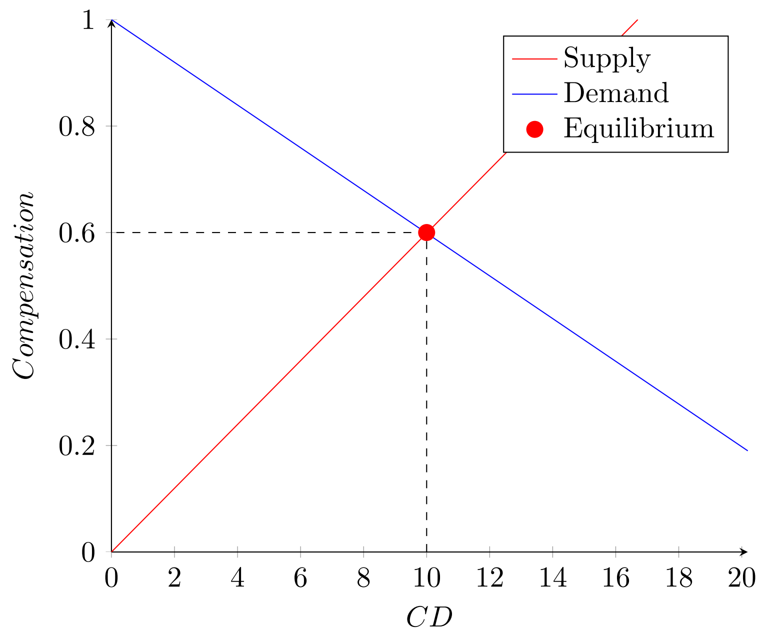

Figure 2: Example of a deterministic market equilibrium

In Figure 2, we describe a market equilibrium that is obtained with average values of supply and demand. We assume that a supply curve exists that maps the average number of CDs available for each compensation level. The compensation is expressed as a percentage of the cost of PVs. In practice, the average supply of CDs will fluctuate. Like in Torres et al. (2021a), we assume that the fluctuations of the average supply of CDs can be represented with a discrete probability function. Such function can be created with the use of historical data. As the compensation increases, the average number of CDs that accept routes also increases. In the case that the compensation is decreased, then the number of routes accepted by CDs also decreases. Notice that the number of CDs that accept routes and the number of routes accepted are interchangeable statements.

The demand curve is obtained by solving vehicle-routing-problem instances with an unlimited supply of CDs and PVs at different compensation levels. However, the crowd-shipping platform needs to find the compensation level that minimizes the total cost of operations under the stochastic setting described in Torres et al. (2021a) and in Section 1. The fluctuation of the supply of CDs will cause the platform to incur expensive recourse actions if CD-routes are rejected.

3.1 Cost structure

In the setting introduced in Torres et al. (2021a), PVs have a first-stage fixed cost of F and a first-stage variable cost equal to the distance traveled in the route, i.e., d_r. CDs have a fixed compensation equal to F' and variable compensation equal to $\beta' d_r$, where $\beta' < 1$ is a parameter that multiplies the distance traveled. The PV-capacity is set to Q and CD-capacity is Q'. CDs are assumed to have smaller capacity than PVs, i.e., $Q > Q'$.

The availability of CDs, i.e., ξ, is considered to be stochastic and a discrete probability distribution, i.e., $P(.)$, is assumed to be available. Given a set of CD-routes H, let $s \in 0, \ldots |H|$ be the preference of a CD-route in the preference ordering of CDs, i.e., the route with preference $s = 1$ is the best compensated route, with $s = 2$ is the second best compensated route, etc. For simplicity when the preference of a route is equal to zero, i.e., $s = 0$, the route is performed by a PV in the first stage.

In model (1)-(3), the first stage consists in creating routes for both PVs and CDs, with the respective costs, i.e., $F + d_r$ and $F' + \beta' d_r$. The second stage, i.e., $\mathscr{Q}(\lambda)$, is nonlinear and requires the sorting of CD-routes from better compensated to least compensated to find the preference for each route in that subset of CD-routes. The probability of route acceptance is high for the first route and decreases for the following routes in the preference ordering. In the second stage, CD-routes that are unfulfilled need to be completed by a PV at a penalty, i.e., $\alpha(F + d_r)$, where $\alpha > 1$ is the penalty that represents the cost of using a PV in the second stage, e.g., the penalty could represent overtime paid to professional drivers. We can separate the expected cost of any route in variable and fixed cost with the following parameters introduced in Torres et al. (2021a):

$$v_s = \begin{cases} 1, & \text{if } s = 0; \\ \beta' + P(\xi < s)(\alpha - \beta'), & \text{otherwise} \end{cases}$$

$$F_s = \begin{cases} F, & \text{if } s = 0; \\ F' + P(\xi < s)(\alpha F - F'), & \text{otherwise} \end{cases}$$

The expected cost of any route is equal to $F_s + v_s d_r$. Recall that for simplicity, $s = 0$ represents first-stage PV-routes. The stochastic parameter ξ represents the total number of CDs available to fulfill routes in the second stage. If the total number of available CDs is less than a route preference, i.e., $\xi < s$, then all CD-routes with preference higher than or equal to s will need to be fulfilled by a PV at a penalty.

4 Description of the Large Neighborhood Search Algorithm (LNS)

In this section, we describe the LNS algorithm that we use in this paper. In Section 4.1, we outline the general procedure of LNS. In Section 4.2, we present the removal operators, while, in Section 4.3, we present the insertion operators. Finally, in Section 4.4, we introduce a new procedure that sorts routes and optimally assigns a vehicle type and CD preference to each route.

4.1 Overview of LNS

LNS was first introduced in Shaw (1998). The main idea is to improve a solution by destroying and repairing the solution iteratively. Heuristics based on LNS 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 LNS heuristic that is used. An initial solution is created using the well-known Clarke and Wright savings heuristic. In each iteration, we select a removal operator to remove a random number of customers. We assign routes to vehicle types and assign the preference ordering for CDs with the Route Assignment Procedure (RAP) described in Section 4.4. Next, we select an insertion operator and re-insert the removed customers back into the solution, followed by RAP. The new solution is accepted if it improves the incumbent solution; otherwise, the acceptance criteria is borrowed from well-known simulated annealing methods, 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 reheat the temperature and continue with LNS. Reheating can help escape local optima by allowing different solutions to be accepted, see Afifi et al. (2013); Ting and Chen (2008); Abramson et al. (1999). Finally, once the total number of iterations equals a predetermined value, i.e., k, the algorithm stops and returns the best solution found.

The removal and insertion operators used within the LNS 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).

Algorithm 1: Overview of LNS

```
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 4.4;
    Select insertion operator;
    RAP, see Section 4.4;
    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
        Reheat;
    end
end
return Incumbent sol;
```

4.2 Removal operators

The removal operators help escape local optima by destroying a large portion of the solution in each iteration. Using different removal operators can help diversify the neighbourhoods that are explored. We consider four removal operators, 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. After extensive computational experiments, the probabilities of selection are set to: 0.7 for RR, 0.1 for RRR, 0.1 for FCR, and 0.1 for LCR, respectively.

- *Random Removal (RR):* Randomly selects a set of customers to remove from the solution with uniform probability. When applied, this operator randomly removes between 5 customers and 25% of customers.

- *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.

4.3 Insertion operators

Insertion operators rebuild the destroyed solution and try to find better solutions in different neighborhoods. We consider four insertion operators: Greedy insertion (GI), Greedy insertion with noise (GIN), PV first insertion (PVFI), and CD first insertion (CDFI). In each iteration a single insertion operator is randomly selected to rebuild the destroyed solution. After extensive computational experiments, the probabilities of selection are set to: 0.7 for GI, 0.1 for GIN, 0.1 for PVFI, and 0.1 for CDFI, respectively.

- *Greedy insertion (GI):* Selects a customer from the list of removed customers and inserts it to the best feasible position. Recall the parameters v_s and F_s from Section 3.1, the variable and fixed cost for each preference. Let i^* be the customer selected for insertion and let $\gamma_s(i,j) = v_s(d_{ii^*} + d_{i^*j} - d_{ij})$ be the variable cost increase of inserting customer i^* in between customers (i,j); the index "s" indicates the preference of the route that contains both customers (i,j), when $s = 0$, it represents a PV route. Let $F_s(i^*)$ be the increase in fixed cost; if the load of the route that contains (i,j) exceeds the capacity of a CD vehicle, then $F_s(i^*) = F - F_s$, otherwise, $F_s(i^*) = 0$. Customer i^* is inserted at the best place that has the minimum cost of insertion, i.e., $(i,j) = \arg\min_{(i,j)}\{\gamma_s(i,j) + F_s(i^*)\}$

- *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 $(\gamma_s(i,j) + F_s(i^*))(1 + 0.1\phi)$, where ϕ is a random number from -2 to 2.

- *PV first insertion (PVFI):* Uses GI to insert a customer from the removed list to the PV-routes first. If no feasible insertion in a PV route exists, then the customer is inserted in the cheapest position within a CD-route.
- *CD first insertion (CDFI):* It is the opposite of PVFI. First it finds the cheapest insertion in CD-routes, if no feasible insertion exits, then the customer is inserted in the cheapest position of PV-routes.

4.4 Route assignment procedure (RAP)

The expected cost of each route depends on the preference the route has in the preference ordering of CDs for all the CD-routes that are offered in the solution. In order to determine the expected cost of a given solution, it is necessary to look at all the routes together and assign the optimal vehicle along with the CD-preference to each route. Let H be an ordered list of routes such that the first route of the list is the longest route, i.e., the $d_r \geq d_{r+1}$ for $r \in H$. If route $r \in H$ has a total load that exceeds the capacity of a CD, then it has to be fulfilled by a PV. Otherwise, the first route that appears in list H with a total load of less than the capacity of a CD is labeled as a CD-route with first preference. The second route of list H with a total load of less than the CD-capacity is labeled with second preference, etc. This assignment process is applied until the expected cost of the route is greater than the cost of using a PV to perform it. When this point is reached, all remaining routes in H are labeled PV-routes. The following is an overview of RAP:

1. Sort the list of routes based on total distance from longest to shortest.
2. Iterate over the ordered list of routes; set $s = 1$.
3. If the total load of the current route exceeds Q', label the route as a PV-route. Adjust the fixed and variable cost of the route accordingly. Move to the following route on the list.
4. If the current route has a total load of less than Q', and the expected cost of the preference s is less than the cost of a PV, then label the route as a CD-route with preference s. Then, add the expected cost accordingly, set $s = s + 1$ and move to the following route on the list. If the cost for the current route with preference s exceeds the cost of a PV, then label the current route as a PV-route.
5. Return the total expected cost of all routes with the assigned vehicle type and CD preference.

Given a set of routes, this simple procedure finds the optimal assignment of the routes with the corresponding preference of the CDs. RAP is applied whenever a removal or insertion operator is used in the solution procedure. When customers are removed, some routes become smaller, which allows RAP to assign CD-routes, and more accurately estimate the insertion cost when inserting customers back into the current solution. By applying RAP after an insertion operator is applied, we can use the acceptance criterion with the expected cost of the current solution.

5 Computational results

In this section, we perform a series of computational experiments to evaluate the solution approach comparing it with the branch-and-price and column generation in Torres et al. (2021a), and provide valuable insights into the properties of crowd-shipping. The LNS algorithm was implemented in Java SE 1.8.0 and it was run in a Linux-CentOS 7 system with an Intel core E5-2683 at 2.1GHz, and 8GB of ram.

In Torres et al. (2021a), the short-routes Solomon instances with 25, 50 and 100 customers are considered and modified by adding a pool of 100 CDs. A binomial distribution is used to represent the fluctuations, i.e., $\xi \sim B(p, M)$. Table 2 shows the parameters that were used for the base case.

Parameters	**Description**	**Value**
Q	PV-capacity	200
Q'	CD-capacity	100
F	PV fixed cost	100
F'	CD fixed cost	50
α	Second stage penalty	2.0
β'	CD variable cost	0.5
p	Binomial distribution probability	0.05
M	Size of the pool of CDs	100
$p \times M$	Average number of CDs	5

Table 2: Parameters for the base case

LNS is a randomized algorithm, hence, we run the algorithm 5 times for all experiments and report the best solution with lowest cost of the 5 results. All the times reported are in seconds.

The remainder of this section is organized as follows. In Section 5.1, we evaluate the performance of the LNS algorithm when compared to the column generation algorithm presented in Torres et al. (2021a). In the following sections, all computational experiments were performed on larger 200-customer instances. In Section 5.2, we provide the deterministic market equilibrium that considers the average number of CDs. Finally, in Section 5.3, we find the market equilibrium when the supply of CDs fluctuates from the average.

5.1 Performance

Table 3 compares the branch-and-price algorithm, i.e., B&P, and the column generation algorithm, i.e., C-Gen, proposed by Torres et al. (2021a) with the LNS 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 in Torres et al. (2021a), 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 LNS heuristic that we implemented. Since LNS 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 LNS with respect to the solution values of C-Gen.

Table 3: LNS performance

		B&P		**C-Gen**			**LNS**			
Ins	N	**LB**	**T(s)**	**Sol**	**T(s)**	**Gap(%)**	**Sol**	**T(s)**	**Gap(%)**	**Dev(%)**
	25	434.85	355	435.92	8	0.27	440.0	40	1.17	0.94
C1	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
	25	652.0	102	652.76	8	0.14	663.02	45	1.69	1.57
R1	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**
	25	618.12	32	618.8	3	0.01	636.23	45	2.93	2.4
RC1	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 LNS is much faster than C-Gen and it provides better average solution values for the larger 100-customer instances. On average, LNS 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 3 clearly show that LNS 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, LNS still provides solutions that are less than 5% from these weak bounds.

5.2 Deterministic Market Equilibrium

Here, we solve larger instances with 200 customers and a pool of 1000 CDs, i.e., $M = 1000$. The same parameters are used for the computations as in Torres et al. (2021a). We assume a deterministic setting where the platform has access to all vehicles in the pool with 100% certainty, i.e., $\xi \sim B(1000, 1000)$. We assume that historical data and information of wages are available for a given city, which allows us to derive a supply curve of the average number of CDs that will participate at each compensation level. The compensation is expressed as the ratio of the compensation of PV. For example, a 0.5 compensation means that $F' = 50, \beta' = 0.5$. We do not change the costs of PVs.

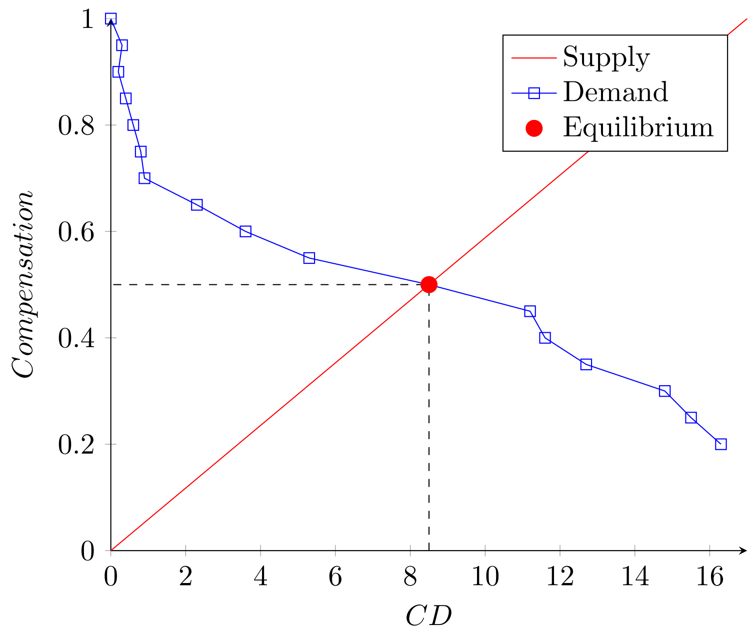

Figure 3: Market equilibrium for C1 200-customer instances

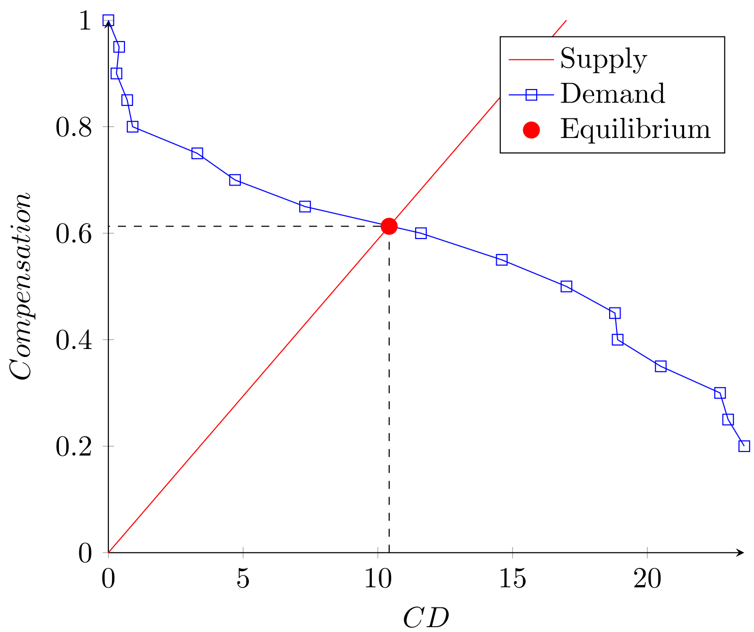

Figure 4: Market equilibrium for R1 200-customer instances

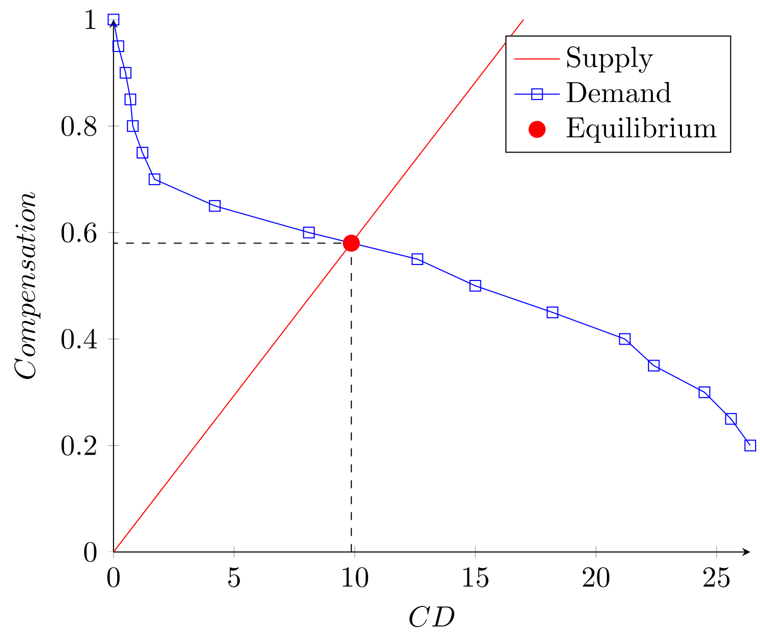

Figure 5: Market equilibrium for RC1 200-customer instances

Figures 3, 4 and 5 show the deterministic market equilibrium for the three instance classes, i.e., C1, R1 and RC1. The demand for CDs is the total number of CDs that we would like to use at the current compensation value, if they were available. The supply curve shows the average number of CDs that are willing to participate at the current compensation level. The supply curve intersects the demand at a value that gives us the deterministic market equilibrium. The compensation at equilibrium for C1 instances is 0.5 for an average number of CDs at 8.5. For R1 instances, the compensation at equilibrium is 0.61 for an average number of CDs at 10.4. For RC1 instances, the equilibrium is found at 0.58 compensation and 9.9 average CDs.

Recall that CDs are smaller than PVs and they are considered to have one half the capacity of PVs. When the compensation level is high, we can see that few CDs are required in the solution, even when we have an unlimited supply. However, once the compensation level is at around 0.7, we can see a rapid change in the slope of the demand curve as more CDs are used in the solution. The use of CDs continues to increase as the compensation decreases until a compensation level of around 0.25. At this point, the slope of the supply curve rapidly changes once more. Even if CDs were free, the platform can only use as many CDs as is necessary to fulfill all delivery requests. Hence, a reasonable compensation level should be in the range between 0.7 and 0.2.

5.3 Stochastic Market Equilibrium

In this subsection, we consider the fluctuations from the average supply of CDs to find an equilibrium. The size of the pool of CDs is the same as in Section 5.2, i.e., $M = 1000$. We consider that the average number of CDs available will follow the same supply curve plotted in Figures 3, 4, and 5. The binomial distribution is adjusted so that the average supply of CDs in the supply curve matches the average value in the binomial distribution, i.e., select h in $B(h, 1000)$ such that $h \times 1000$ intersects the supply curve. In this setting, random variations of the supply of CDs are explicitly considered. Moreover, in this case, the expected cost also includes the penalties associated with the recourse actions of having additional PVs to fulfill the delivery requests.

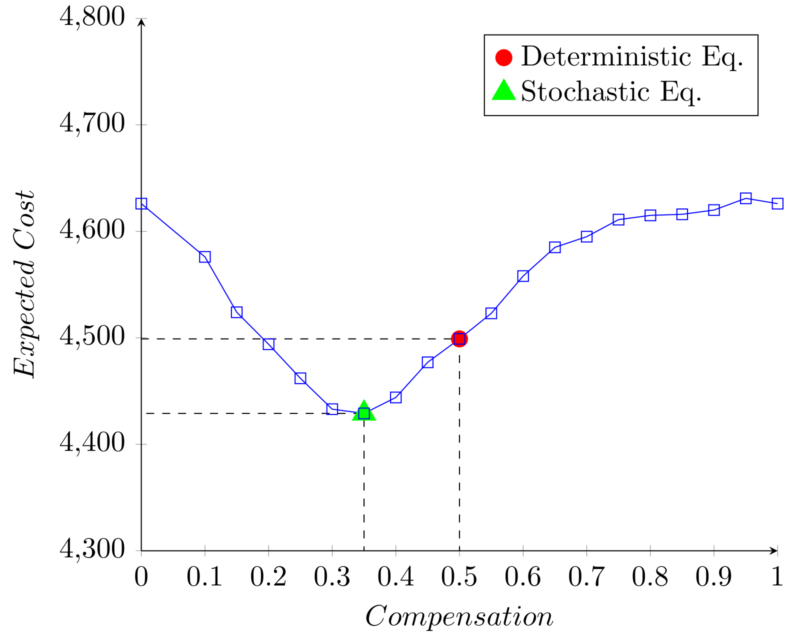

Figure 6: Best compensation for C1 200-customer instances

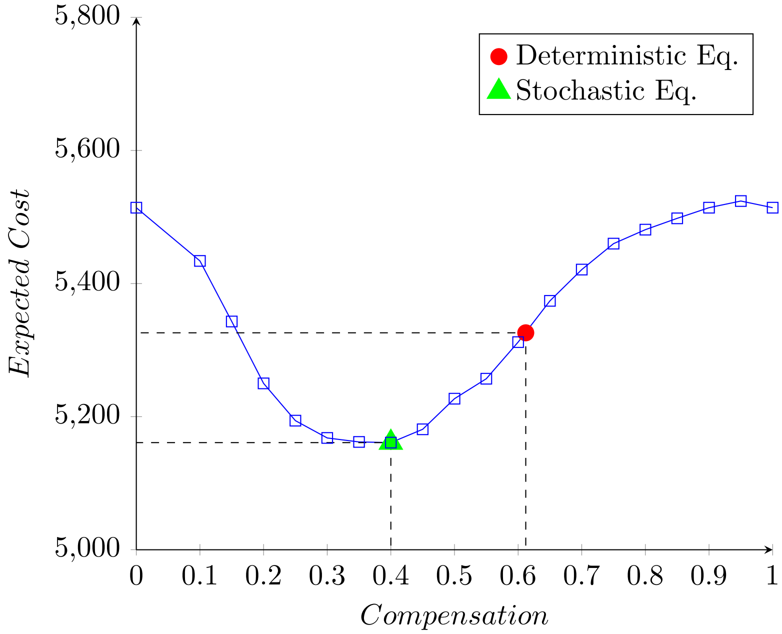

Figure 7: Best compensation for R1 200-customer instances

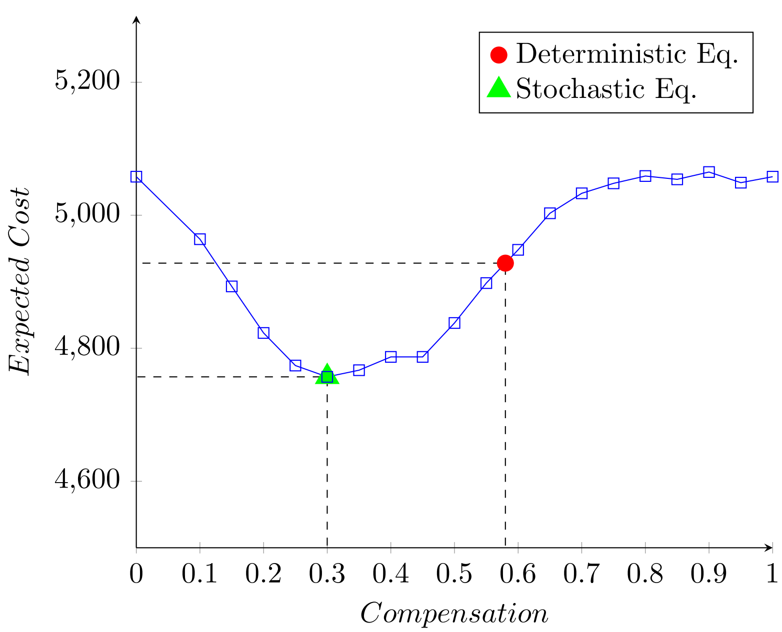

Figure 8: Best compensation for RC1 200-customer instances

Figures 6, 7 and 8 show the deterministic and stochastic market equilibrium for all three instance classes, i.e., C1, R1 and RC1. The platform will adjust the compensation to minimize the expected cost. When the compensation decreases, so does the participation. If the participation drops too much, the expected cost starts to rise. On the other hand, if the compensation is too high, the platform prioritizes the use of larger PVs to fulfill deliveries. We can see that the difference between the deterministic and stochastic market equilibrium is about 6% of the expected cost of the platform. Notice this difference is not the cost reduction of using CDs. Rather, it is the cost reduction when we consider the random fluctuations of the supply.

5.4 CD-Participation

In previous sections, it is assumed that the crowd-shipping platform wants to minimize the total expected costs. However, if the crowd-shipping platform has a strategic objective that seeks to increase participation in the platform, then it is better to establish the compensation level higher to increase participation. Figure 9, shows the maximum participation can be achieved at a compensation level equal to 0.5. When the compensation is low, few CDs want to participate, when the compensation is high they become too expensive for the crowd-shipping platform and less CDs are employed in the solution.

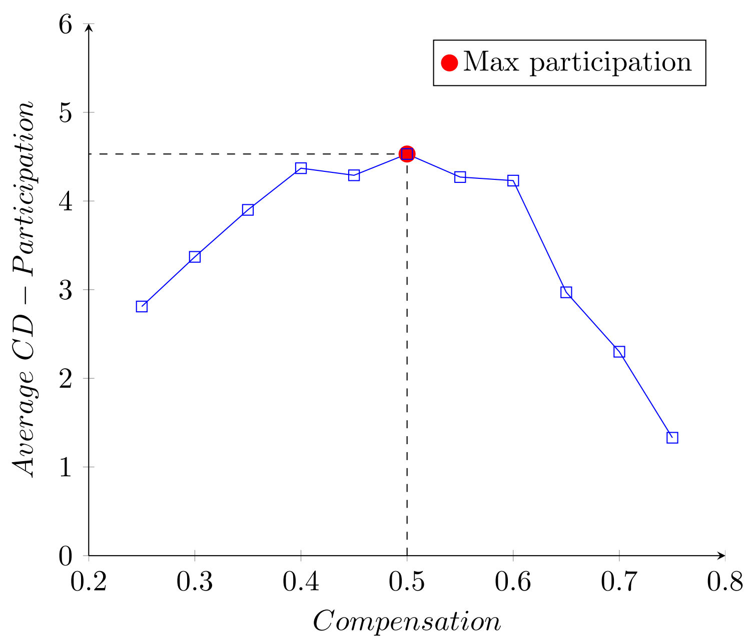

Figure 9: Average CD-Participation

In Figure 10, we report the compensation level that maximizes the average total compensation paid to CDs for all instances. We can see that the compensation level at 0.6 maximizes the total compensation given to CDs. If the strategic objective is to maximize participation, this could be a good way of doing it, i.e., by maximizing the compensation CDs receive. The overall participation does not decrease significantly at a compensation level of 0.6, however, after that level CDs become too expensive in comparison to the much larger and available PVs.

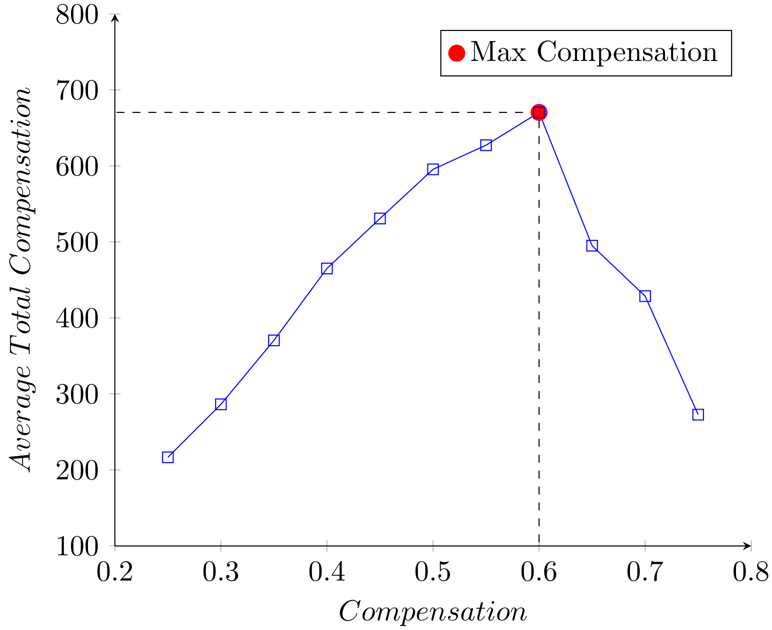

Figure 10: Average total Compensation given to CDs

6 Conclusions and future research

In this paper, we developed a LNS algorithm that outperforms the column generation method proposed by Torres et al. (2021a). A new procedure was introduced that orders routes based on the preference ordering of CDs and computes the total expected cost of the solution based on the CD-preference for each route. We showed that the compensation determined by the deterministic market equilibrium is not the compensation that minimizes the expected shipping costs of the platform when we consider the stochastic route acceptance of CDs. We determined the compensation that minimizes the total expected cost in a setting where the supply of CDs randomly fluctuates. We also indicated how different strategic objectives, e.g., maximizing crowd-driver participation, could be achieved by increasing the compensation level from the level at the market equilibrium.

There are different variants of crowd-shipping that exist, some consider the planned destinations of CDs while others consider pick-up and delivery problems, etc. Considering that the compensation strategy that is used is one of the main drivers that motivates the participation of CDs, finding the optimal strategies for various crowd-shipping settings is certainly a worthy avenue of research to pursue. Also, few stochastic variants of crowd-shipping exist. However, stochasticity is one of the main characteristics of crowd-shipping. Variants that consider the stochastic properties of CDs are still needed in the scientific literature, in particular, variants that consider the probability of route acceptance as dependent on the set of offered routes are scarce. If crowd-drivers are truly free to choose, then the probability of route acceptance must be dependent on the set of offered routes.

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.

References

D. Abramson, M. Krishnamoorthy, H. Dang, et al. Simulated annealing cooling schedules for the school timetabling problem. *Asia Pacific Journal of Operational Research*, 16:1–22, 1999.

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.

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

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.

B. Asdecker and F. Zirkelbach. What drives the drivers? a qualitative perspective on what motivates the crowd delivery workforce. In *Proceedings of the 53rd Hawaii International Conference on System Sciences*, 2020.

A. Bhatti, H. Akram, H. M. Basit, A. U. Khan, S. M. Raza, and M. B. Naqvi. E-commerce trends during covid-19 pandemic. *International Journal of Future Generation Communication and Networking*, 13(2):1449–1452, 2020.

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.

B. He, V. Gupta, and P. Mirchandani. Online selling through o2o platform or on your own? strategic implications for local brick-and-mortar stores. *Omega*, 103:102424, 2021.

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

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.

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.

S. Mancini and M. Gansterer. Bundle generation for last-mile delivery with occasional drivers. *Omega*, page 102582, 2021.

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

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.

A. Sampaio, M. Savelsbergh, L. Veelenturf, and T. Van Woensel. Crowd-based city logistics. In *Sustainable Transportation and Smart Logistics*, pages 381–400. Elsevier, 2019.

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.

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.

C.-J. Ting and C.-H. Chen. Combination of multiple ant colony system and simulated annealing for the multidepot vehicle-routing problem with time windows. *Transportation Research Record*, 2089(1):85–92, 2008.

F. Torres, M. Gendreau, and W. Rei. Vehicle routing with stochastic supply of crowd vehicles and time windows. *Transportation Science*, 2021a.

F. Torres, M. Gendreau, and W. Rei. Crowd-shipping: An open vrp variant with stochastic destinations. *CIRRELT*, 2021b.

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

A Appendix

Table 4: Average results for all instances

		Compensation			Cost		Vehicles	
Comp.	**Ins.**	**Fix**	**Var**	**Total**	**Recourse**	**Total**	**CD**	**Total**
0.25	C1	68.50	82.92	151.42	89.59	4462.46	2.74	20.26
	R1	67.50	178.47	245.97	145.88	5194.60	2.70	19.70
	RC1	75.0	177.51	252.51	141.79	4774.85	3.0	20.0
	Average	70.33	146.30	216.63	125.75	4810.64	2.81	19.98
0.30	C1	96.0	107.89	203.89	75.30	4441.75	3.20	20.30
	R1	114.0	234.91	348.91	153.76	5168.14	3.80	20.20
	RC1	93.0	213.37	306.37	89.03	4757.60	3.10	20.0
	Average	101.0	185.39	286.39	106.03	4789.16	3.37	20.17
0.35	C1	136.50	139.66	276.16	70.95	4429.07	3.90	20.40
	R1	150.50	306.64	457.14	135.17	5162.08	4.3	20.40
	RC1	122.50	255.62	378.12	77.32	4764.68	3.50	20.20
	Average	136.50	233.97	370.47	94.48	4785.28	3.90	20.33
0.40	C1	160.0	162.37	322.37	44.75	4444.12	4.0	20.50
	R1	180.0	364.46	544.46	110.78	5161.98	4.50	20.40
	RC1	184.0	344.47	528.47	95.57	4787.46	4.60	20.80
	Average	174.67	290.43	465.10	83.70	4797.85	4.37	20.57
0.45	C1	165.0	180.82	345.82	20.26	4477.51	3.67	20.33
	R1	207.0	414.55	621.55	71.84	5181.69	4.60	20.50
	RC1	207.0	418.18	625.18	59.29	4787.53	4.60	20.70
	Average	193	337.85	530.85	50.47	4815.58	4.29	20.51
0.50	C1	185.0	183.29	368.29	11.53	4499.23	3.70	20.40
	R1	250.0	475.02	725.02	61.59	5227.20	5.0	20.60
	RC1	245.0	448.34	693.34	36.98	4838.86	4.90	20.90
	Average	226.67	368.89	595.55	36.70	4855.10	4.53	20.63
0.55	C1	181.50	191.19	372.69	3.90	4523.47	3.30	20.2
	R1	280.50	530.95	811.45	46.22	5257.60	5.10	20.60
	RC1	242.0	455.79	697.79	16.02	4898.13	4.40	20.70
	Average	234.67	392.64	627.31	22.05	4893.07	4.27	20.50
0.60	C1	204.00	210.19	414.19	2.07	4558.37	3.40	20.30
	R1	294.0	561.63	855.63	37.96	5312.47	4.90	20.60
	RC1	264.0	477.67	741.67	10.96	4948.0	4.40	20.70
	Average	254.00	416.49	670.49	16.99	4939.62	4.23	20.53
0.65	C1	110.50	106.65	217.15	0.12	4585.61	1.70	19.70
	R1	286.00	492.40	778.39	17.06	5373.97	4.40	20.30
	RC1	182.00	307.67	489.67	4.34	5002.79	2.80	19.80
	Average	192.83	302.23	495.07	7.17	4987.46	2.97	19.93
0.70	C1	91.00	81.43	172.43	0.05	4595.23	1.30	19.50
	R1	238.00	462.03	700.03	9.92	5421.82	3.40	19.90
	RC1	154.00	259.69	413.69	0.74	5033.89	2.20	19.40
	Average	161.00	267.72	428.72	3.57	5016.98	2.30	19.60
0.75	C1	45.00	40.22	85.22	0.01	4611.53	0.60	19.40
	R1	172.50	324.97	497.47	3.32	5460.11	2.30	19.30
	RC1	82.50	153.16	235.66	0.19	5048.84	1.10	18.90
	Average	100.00	172.78	272.78	1.17	5040.16	1.33	19.20