CIRRELT

CIRRELT-2020-05

Vehicle Routing with Stochastic Supply of Crowd Vehicles and Time Windows

Fabian Torres
Michel Gendreau
Walter Rei

February 2020

Bureau de Montréal
Université de Montréal
C.P. 6128, 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 Palais-Prince, local 2415
Québec (Québec) G1V 0A6
Tél. : 1-418-656-2073
Télécopie : 1-418-656-2624

Vehicle Routing with Stochastic Supply of Crowd Vehicles and Time Windows

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, P.O. Box 6079, Station Centre-Ville, Montréal, Canada H3C 3A7

[3] Department of Management and Technology, Université du Québec à Montréal, P.O. Box 8888, Station Centre-Ville, Montréal, Canada H3C 3P8

Abstract. The growth of e-commerce has increased demand for last-mile deliveries, increasing the level of congestion in the existing transportation infrastructure in urban areas. Crowdsourcing deliveries can provide the additional capacity needed to meet the growing demand in a cost-effective way. We introduce a setting where delivery requests are fulfilled from a single depot by a fleet of professional drivers and a pool of crowd drivers. The uncertainty of crowd-driver availability is described by a binomial distribution. We formulate a two-stage stochastic model and propose a branch and price algorithm to solve the problem exactly. We further develop an analytical method to calculate upper bounds on the supply of vehicles that are feasible and an innovative cohesive pricing problem to generate columns for the pool of crowd drivers. Computational experiments are carried out on modified Solomon instances with a pool of 100 crowd vehicles. The algorithm is able to solve instances of up to 100 customers and solves all instances of 25 customers. We show that the value of the stochastic solution can be as high as 17% when compared with the solution obtained from a deterministic simplification of the model. Significant cost reductions of 35 % are achieved by our framework. Finally, we provide interesting insights on the compensation of crowd-sourced divers compared with professional drivers.

Keywords: Crowd-shipping, crowd-logistics, crowd drivers, occasional drivers, city logistics, stochastic programming, dynamic programming, column generation.

Acknowledgements. This research was partially funded by the Natural Sciences and Engineering 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 and Polytechnique Montréal.

---

* Corresponding author: fabian.torres@polymtl.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2020

1 Introduction

The growth of on-line retailing has caused significant disruptions to traditional brick and mortar retailers, and increased the total load on existing transportation infrastructures. At the same time a new phenomenon called the sharing economy Sundararajan (2016); Ertz et al. (2016), is allowing individuals in society to use their own vehicles to transport people or perform deliveries. Ride-sharing platforms like Uber and Lyft, or delivery platforms like Zipments, Deliv, AmazonFlex, among others, have started utilizing the colossal vehicle fleet made available by “the crowd” to fulfill transportation requests, a strategy that is often referred to as “crowd-shipping” when delivering packages. In a recent survey by Sampaio et al. (2019), the authors note that there is no commonly agreed definition of “the crowd” in the literature. Consequently what we mean by a “crowd-sourced driver” (CD), in general terms, is an independent individual that has access to a vehicle and is willing to exchange her time to perform a delivery task for a monetary compensation, without contracting any obligation to perform any other future task. The term “delivery task”, or “shipping”, is less contentious, it generally means a requirement to transport an object from one place to another, by any means (e.g., through pick up and delivery operations, the implementation of an open or closed route, etc.).

Crowd-sourced drivers decide when to work delivering packages and are not bound to any company, they can work one day for Uber and another at Deliv or AmazonFlex, or take a long vacation. They might even have a favorite platform which they prefer over the other ones, even if the compensation is lower. All the various types of platforms now have to compete for the supply of crowd drivers that are willing to fulfill delivery requests. This phenomenon represents a big challenge and opportunity, for logistic companies. When a company has its own fleet of vehicles, vehicle availability is generally considered to be deterministic,(an exception is vehicle breakdowns Mu et al. (2011)), but with a fleet of crowd-sourced vehicles (CV), the availability is not a decision variable that the company can control directly. Vehicle availability depends on the CDs’ behaviour and willingness to participate in any of the available programs that they have at their disposal, placing uncertainty (and thus risk of failure) in the entire delivery process (Sampaio et al. (2019)). Conversely by using CVs to perform delivery tasks the environmental impact could be reduced, as well as providing flexibility to increase the fleet at peak times throughout the year and decrease it at low times without having to buy or sell vehicles or hiring more professional drivers, which entails important additional fixed costs.

In this paper we introduce a setting where a crowd-shipping platform (CSP) has a program for individual car owners that allows them to become CD if they meet some basic requirements. More specifically, besides some security requirements the CSP requires that CVs have at least a standard capacity of Q', similar to the AmazonFlex program. The CSP program not only allows the capacity of CVs to be establish in advance but it also allows us to have probabilistic information that describes the supply of CVs during specific days. Drivers have different preferences, some might be less likely to deliver packages on rainy days or weekends. In any case, the supply of CVs on a given day could be described by a binomial distribution where each trial represents the probability of a CV becoming available during the day. The CSP has to solve the problem of routing a mixed fleet of company owned professional vehicles (PVs) and a pool of stochastic crowd drivers to perform closed routes. By returning to the depot, undelivered packages can be easily rescheduled for future delivery.

1.1 Related work

The concept of crow-shipping proposed by Walmart, i.e., of customers delivering packages (Barr and Wohl (2013)), has spawned a series of needed quantitative studies and mathematical models. In a study done by Archetti et al. (2016), a static deterministic model is introduced called the vehicle routing problem with occasional drivers (VRPOD). In this variant of the vehicle routing problem (VRP), the company owns a fleet of professionally driven vehicles (PV) that can only perform closed routes, and a set of static and deterministic occasional drivers (ODs) that can only visit one customer before they head to their intended destination, from where they do not return to the depot. Since then other variants of the VRPOD have been explored. We believe that the branch of research that includes ODs and a fleet of regular vehicles can be classified as a variant of the heterogeneous VRP (HVRP), if the fleet size is known, or the fleet size and mix VRP (FSM), if the fleet size is unknown. In a recent survey dedicated to the HVRP by Koç et al. (2016), it is noted, due to the difficulty of the HVRP, that most of the work in this field has been done with metaheuristics, and very limited exact methods exist. Many interesting variants are mentioned that are similar to the extensions of the homogeneous VRP, (e.g., time windows for customer visits, pickup and delivery problems, multi depot HVRP, HVRP with transshipment nodes, etc.). An extension similar to the VRPOD is the open HVRP (OHVRP) Li et al. (2012), where a heterogeneous vehicle fleet does not have to return to the depot, this problem is different from Archetti et al. (2016) in important ways. The OHVRP considers that all vehicles have open routes that can end at any customer, and the only characteristic that is considered to be different is the capacity. Conversely, in the VRPOD, the regular fleet has to return to the depot completing closed routes and OD can perform open routes all the way to their destination. Essentially the difference between vehicles goes beyond just the capacity, and even the compensation can be different. Consequently this leads to a whole new family of HVRP where various characteristics of heterogeneous vehicles are different, for example compensation, or specific restrictions for each vehicle type. More concisely the capacity is only one dimension that can make a fleet heterogeneous.

We divide the literature that extends the work done by Archetti et al. (2016) into two main categories, studies that consider the dynamic aspects of OD and research that introduce static deterministic variants. In the latter group Macrina et al. (2020) consider ODs that can deliver packages to transshipment nodes and regular vehicles can be used to complete deliveries, a mixed integer programming model is presented and a variable neighborhood search heuristic is develop to analyse the benefits of a logistical system with ODs, professional drivers and transshipment nodes. Another deterministic and static variant is the green VRP with OD, presented by Macrina and Guerriero (2018), where a green VRP is complemented by access to ODs, an integer programming formulation is developed and the benefits of OD are evaluated.

The dynamic aspects of the problem were first considered by Arslan et al. (2019), where a pick-up and delivery problem with ad-hoc drivers is introduced. A platform has access to a fleet of backup vehicles and a set of ad-hoc drivers that can only deviate a certain amount form their destination. An event-based rolling horizon framework is developed that repeatedly solves the problem of matching ad-hoc drivers with pickup and delivery tasks. In a similar extension Dahle et al. (2019) present a setting where pick-up and delivery can be done by OD as well as regular vehicles from a central depot. A 3-index formulation and symmetry breaking constraints are used to solve instances of up to 70 customers and 50 ODs, different compensation schemes are considered. Another interesting dynamic variant was presented by Dayarian and Savelsbergh (2017) where the dynamic aspects of both customers and OD are introduced. A sample scenario planning approach with tabu search is used to derive results for compensation and service quality. The uncertainty presented by ODs is also noted by Dahle et al. (2017), in the VRP with dynamic ODs. The authors present a two-stage stochastic model to represent the uncertainty of up to 3 ODs and optimally solve instances with up to 10 delivery requests utilizing a commercial solver. Each dynamic OD has a fixed destination and a probability of being available during the planing horizon. A penalty per customer is added to the objective if the customer is not visited.

Naturally crowd-sourcing a delivery task under any *modus operandi* would be highly stochastic due to the uncertainty originating from the drivers' behaviours and preferences, which can shift from one day to the next based on different events e.g., rain, requests on other platforms, etc. (Sampaio et al. (2019)).

The survey by Koç et al. (2016) report many variants of the HVRP, but none with a heterogeneous fleet composed of vehicles with deterministic and stochastic supply. The stochastic variants of the HVRP mentioned just consider the stochastic demandsTeodorovic et al. (1995).

In classical variants of stochastic VRPs, the uncertainty generally originates from the customers. In a survey by Psaraftis et al. (2016), the nature of uncertainty identified are primarily, stochastic demands Salavati-Khoshghalb et al. (2019), stochastic customer presence Gendreau et al. (2014), stochastic travel times Laporte et al. (1992), or stochastic service times Errico et al. (2018).

An interesting study that considers demand uncertainty, is the stochastic fleet composition problem, presented by Loxton et al. (2012), a company has to decide the quantity of vehicles of different types it needs in order to satisfy future uncertain demands represented by a binomial distribution. At each stage the demand for each vehicle type becomes known. There are only two cases, the chosen fleet is adequate to meet the demand or there is a shortage of vehicles, in the latter case a penalty for each vehicle type that is missing is added to the cost, the penalty represents the cost of having to hire an additional vehicle at a later stage. While Loxton et al. (2012) consider a stochastic "demand" for vehicles, in this paper we consider a stochastic "supply" of vehicles. It might seem like a trivial difference, but it actually represents a major paradigm shift in city-logistics caused by the sharing economy. We represent the stochastic supply by a binomial distribution while the demand is described by a set of delivery requests that must be fulfilled by CVs, or PVs. The recourse action taken if the supply is less than the demand, is to utilize the deterministic fleet of PVs to complete the CV routes. A penalty is considered for having to hire or have access to additional deterministic PVs at the second stage to fulfill the demand. The penalty is proportional to the length of the routes that are not fulfilled by CVs as well as a fixed cost.

The Vehicle Routing Problem with Stochastic Crowd-Vehicles and Time Windows (VRPSVTW) is the problem of routing two fleets of vehicles, one deterministic and the other stochastic and assessing if there is value in the stochastic fleet. The deterministic version of the problem resembles a heterogeneous VRP with time windows, two vehicle types, fixed and variable costs.

1.2 Our contributions

The main contributions of this paper are the following:

1. We introduce the vehicle routing problem with stochastic supply of crowd-vehicles and time windows.

2. We formulate a two-stage stochastic set partitioning model and further improve the model by developing an optimal policy to dynamically assign routes to vehicles and using additional variables that make the second stage problem linear.

3. We derive upper bounds on the maximum number of crowd-vehicles that can be used in the optimal solution, and utilize such bounds to bring valuable insights to the compensation of crowd-drivers.

4. We develop a branch and price algorithm, and create an innovative cohesive sub-problem that is capable of generating all the columns needed for the master problem.

5. Finally, we perform extensive computational experiments on modified Solomon instances, solving problems with 100 delivery requests and show the value of the stochastic solution and the impact of uncertainty when routing a fleet of stochastic vehicles.

The remainder of this paper is organized as follows. In section 2 we formally describe the problem and present a set partitioning formulation. In section 3 we present the solution approach. In section 4 we show extensive computational results and in section 5 we conclude and present some future research directions.

2 Vehicle Routing with Stochastic Supply of Crowd-Vehicles and Time Windows

In this section we describe and formulate The Vehicle Routing Problem with Stochastic Supply of Crowd-vehicles and Time Windows (VRPSVTW) as a two-stage stochastic model.

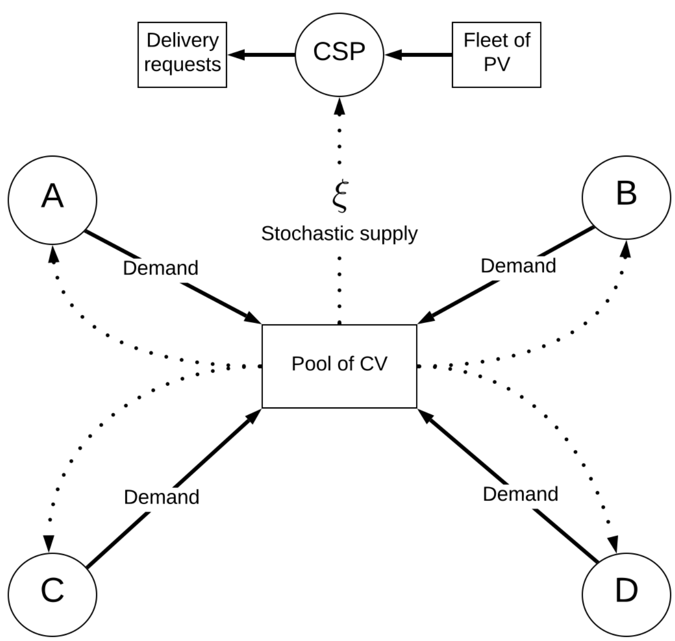

Figure 1: Stochastic supply of CV

Consider the CSP (Crowd-Shipping Platform), briefly described in the introduction, that has started a crowd shipping program and standardized the capacity of acceptable crowd vehicles to Q'. The CSP has to complete a series of deliveries, with access to its own vehicles (PVs) and the pool of accepted drivers that can easily switch to another platform based on demand and the preference of drivers on any specific day. Figure 1 represents the crowd-driver market with the pool of CDs at the center describing the supply and a set of platforms A,B,C, and D, that have a demand for the same pool, although not all vehicles will participate in all platforms. Some drivers might sign up in some platforms and not others, therefore we consider that there is some overlap between the pool of drivers accessible to all platforms. We assume that there is no monopsony or collusion in the crowd-sourcing market, and all platforms are competitors so that they are not cooperating with each other e.g., by sharing information. If a single company controlled all the demand for CVs, crowd drivers would have unfair wages because they would have no other option besides the only available company. We also assume that there is no labor union or any organized structure of crowd drivers either, so that the CDs decide what they will do independently of each other based on their own preferences, without knowing what other CDs are doing.

From the perspective of the CSP, the demand for CVs on other platforms and the quantity of CDs that will decide to participate at any platform, on any given day, are both unknown. Resulting in the stochastic supply of crowd vehicles represented by the random parameter ξ. We assume probabilistic information describing ξ, will be available thanks to the historical data showing the supply of CVs on each day. The CSP must commit PVs before knowing the total supply of CVs that will become available throughout the planning horizon. We consider a two stage decision process described by figure 2.

Figure 2: Two-stage decision process

First the CSP must plan two sets of routes, one set for their own fleet of PVs, and another for CVs, without knowing the supply of CVs. At the first stage PVs start the routes planned for them. At the second stage the supply of CV becomes known such that the remaining routes are assigned to the available CVs. Since the supply is stochastic, sometimes there will be an excess of CVs in which case all routes are completed without any problems, but conversely, if there are fewer CVs than required, expensive recourse actions will need to be taken to complete all delivery requests. Specifically, we will consider the cost of a PVs times a penalty to complete the routes that are not able to be completed due to a lack of CVs. This recourse action represents the additional cost to complete a route at the second stage, it could be for example, overtime paid to company drivers to complete the remaining unfulfilled routes.

2.1 Model

We consider an unlimited fleet of professional vehicles (PVs) with capacity Q, fixed cost F and variable cost parameter β that multiplies the distance traveled. And a pool of crowd-drivers K', with vehicles that have a capacity of at least $Q' < Q$, fixed costs $F' < F$, and variable cost equal to the distance. Information about the random variable ξ is provided by a binomial distribution with parameters p and M. The size of the pool of CVs, $M = |K'|$ is the number of Bernoulli trials, and p is the probability of success of each trial. The recourse cost in the second stage is the cost of a PV times a penalty α, for each additional PV that has to be used to complete unfulfilled routes.

The problem is described by a complete directed graph $G(A, V)$. Each customer $i \in N = V \setminus \{0\}$ has a delivery request with demand q_i that must be meet within a time window $[t_i^e, t_i^l]$, for earliest and latest arrival times respectively. The set of vertexes V represents the customers and a central depot and A is the set of arcs connecting each vertex.

2.1.1 Notation

Sets	
V	Set of all vertices;
$N = V \setminus \{0\}$	Set of customers;
A	Set of arcs connecting all customers;
K'	Set of all CVs in the pool;
Ω	Set of all feasible routes for PVs;
Ω'	Set of all feasible routes for CVs;
$M(\lambda) \subseteq \Omega'$	Set of CV routes constructed in the first stage solution λ;
Deterministic Parameters	
p	Probability of success of each Bernoulli trial in the binomial distribution;
$M = \|K'\|$	Total size of the pool of CVs;
F and F'	Fixed cost of PVs and CVs respectively;
Q and Q'	Maximum capacity of PVs and CVs respectively;
$\alpha > 1$	Penalty for utilizing a PV in the second stage;
$\beta > 1$	Factor that multiplies distance to give the variable cost of a PV;
z_r	Recourse cost of route $r \in M(\lambda)$;
q_i	Demand at customer $i \in N$;
t_i^l	Latest arrival time at $i \in N$;
t_i^e	Earliest arrival time at $i \in N$;
t_i	Service time at $i \in N$;
d_{ij}	Distance between vertices $i, j \in A$;
c_r	Total variable and fixed cost of a route $r \in \Omega$;
a_{ir}	Equal to 1 if customer i is visited in route r;
Stochastic Parameters	
$\xi \sim B(M, p)$	Stochastic variable representing the supply of CVs;
Variables	
λ_r	Binary variable equal to 1 if route $r \in \Omega \cup \Omega'$ is part of the optimal solution;
y_r	Second stage binary decision variables $r \in M(\lambda)$;

Table 1: Notation

2.1.2 Set-partitioning

The set-partitioning formulation is a well known model based on the Dantzig-Wolf decomposition method that has been successfully applied to solve different types of VRPs, combined with a branch and price solution approach.

Let Ω be the set of all feasible routes for PVs, and Ω' the set of all feasible routes of CVs. The fixed and variable cost of a route $r \in \Omega \cup \Omega'$, is equal to c^r and the binary variable λ^r is equal to 1 only if route r is chosen in the solution. Let the parameters a_i^r be equal to 1 if customer $i \in N$ is visited by route r, and M is the size of the pool of CVs. Then the set-partitioning formulation for the VRPSVTW is the following model:

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

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

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

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

Constraints (2) ensure that each customer is visited exactly once by only one vehicle. Constraint (3) restricts the total number of routes that can be assigned to CVs, e.g., the size of the pool M. The objective (1), is to minimize the total cost of routing plus the expected recourse cost.

2.1.3 Second stage problem

Once first stage decisions are made and the supply of CVs represented by the stochastic parameter ξ becomes known, the CSP has to make optimal second stage decisions to minimize the recourse cost.

Let $M(\lambda) \subseteq \Omega'$ be a set containing all routes constructed in the first stage for CVs for a solution λ and for all $r \in M(\lambda)$ there is a total distance of the route equal to d_r and the recourse cost is equal to the penalty α times the fixed and variable cost of a PV i.e., $\alpha(F+\beta d_r)$. The cost of route $r \in M(\lambda)$ if a CV is available at the second stage is already considered at the first stage objective (1) by the coefficient c_r with a value equal to the fixed cost of a CV and variable cost i.e., $F'+d$.

Let z_r be a second stage coefficient that has to be considered if there is a lack of supply of CVs such that route r has to be completed by an additional PVs at an extra cost. The second stage recourse cost coefficient that considers only the additional cost if a PVs completes route $r \in M(\lambda)$ is defined as follows:

$$z_r = \alpha F + \alpha\beta d_r - F' - d_r$$

We assume that the CSP decides the assignment of routes to available CVs and that their corresponding drivers are willing to complete the routes that are assigned to them by the CSP. This is a reasonable assumption since the most expensive route will have an even higher recourse cost and will also be the most profitable for CDs. Let y_r be binary second stage decision variables that equal 1 if route r requires a recourse action and 0 otherwise. The problem that the CSP has to solve to minimize the cost is the following.

$$Z(\lambda,\xi) = \min \sum_{r}^{M(\lambda)} z_r y_r \tag{4}$$

$$\text{s.t.} \sum_{r}^{M(\lambda)} y_r \geq |M(\lambda)| - \xi \tag{5}$$

$$y_r \in \{0,1\}, \forall r \in M(\lambda)$$

Constrain (5) bounds the variables below by the amount of routes that have to fail if there is a difference between routes and available vehicles. The objective (4) is to minimize recourse cost.

Note that ξ and λ are now known parameters since the supply of vehicles becomes known in the second stage and a set of planned routes that visit all customers were established in the first stage.

When the stochastic variable ξ follows a binomial distribution with probability of success of each Bernoulli trial equal to p and the quantity of trials equals the size of the pool M then the recourse function $\mathscr{Q}(\lambda)$, present in the objective of (MP) is equal to the following equation:

$$\mathscr{Q}(\lambda) = \mathbb{E}_\xi Z(\lambda, \xi) = \sum_{\xi=0}^{M} Z(\lambda, \xi) \binom{M}{\xi} p^\xi (1-p)^{M-\xi} \tag{6}$$

Therefore, equation (6) computes the expected recourse cost associated with the decisions of replacing CV routes with PV ones whenever the available supply of CV is insufficient to implement the planned routes.

3 Solution approach

In this section we develop a branch and price algorithm to solve the problem described in the previous section. Firstly in section 3.1 we show the optimal policy to assign routes to drivers in the second stage. In section 3.2 we describe the master problem. In Section 3.3 we derive upper bounds on the total number of CV routes that can be planned in the first stage, in section 3.4 we describe the pricing problem to generate columns to the master problem and finally in section 3.5 we show the branching strategy used to divide a fractional solution into two disjoint problems.

3.1 Optimal policy

In order to find an optimal solution for the second stage problem (4)-(5) given a set of routes $M(\lambda)$, it is imperative to develop a policy to assign routes to vehicles and find the routes that need a recourse action for a realization of the parameter ξ.

Let $H(\lambda) = (r_1, \dots, r_{|M(\lambda)|})$ be an ordered set of routes for a first stage solution λ that describes a sequence of routes $r \in M(\lambda)$ such that the recourse cost decreases monotonically i.e., $z_{r_i} \geq z_{r_{i+1}}$.

Proposition 1. *For a given realisation of the supply ξ, the solution to problem (4)-(5) is obtained by assigning the available CVs to the first elements of $H(\lambda)$.*

Proof. Suppose there is a second stage problem $Z(\lambda, \xi^*)$ for which there exists a better solution for a realization of the random variable ξ^* that does not follow the policy described by $H(\lambda)$. Let $y(\lambda, \xi^*)$ be the optimal solution of problem $Z(\lambda, \xi^*)$. There are at least two routes $y_{r_s}(\lambda, \xi^*) = 1$, $y_{r_i}(\lambda, \xi^*) = 0$ where $s < i$ and the recourse costs are $z_{r_s} > z_{r_i}$. Exchanging these routes reduces the objective value and leads to a contradiction. □

Therefore, the optimal policy is to assign the route with the largest recourse cost to the first available CV.

The two-stage decision process shown in figure 2 which we consider in this paper, is a good approximation of the dynamic aspects of the supply of CVs. In practice CVs will not necessarily show up simultaneously, instead they will arrive throughout the day, one by one at different intervals. It is important to note that the policy of assigning the first unfulfilled route in the ordered set $H(\lambda)$ to the first available CV is a practical implementation of the policy for the dynamic case i.e., as CVs become available one by one.

3.2 Master problem (MP)

With the policy described by proposition 1, it is easy to see that the expected recourse cost of any route is going to depend on the order in the list $H(\lambda)$ that it is given. If there are other routes that have a higher priority in the list then those will be done first. If each route is given a priority in the first stage then the expected recourse cost can be easily computed.

Let c_s^r be the cost of route $r \in \Omega'$, if its priority is $s = \{1, \ldots, M\}$, then route r would be assigned to a CV only after $(s-1)$ more expensive routes have been performed. Let $P(s > \xi)$ be the probability that the CV supply ξ is strictly lower than the priority s, then the expected cost can be computed as follows:

$$c_s^r = c^r + z_r P(s > \xi) \tag{7}$$

Binary variables λ_0^r and λ_s^r are introduced in the model to formulate the selection of regular and stochastic vehicles respectively. They are equal to 1 if route r is selected with priority s and 0 otherwise. Then the master problem for the VRPSVTW is defined as follows:

$$MP = \min \sum_{r \in \Omega} c_0^r \lambda_0^r + \sum_{s=1}^{M} \sum_{r \in \Omega'} c_s^r \lambda_s^r \tag{8}$$

$$\text{s.t.} \quad \sum_{s=0}^{M} \sum_{r \in \Omega \cup \Omega'} a_{ir} \lambda_s^r \geq 1 \qquad \forall i \in N \tag{9}$$

$$\sum_{r \in \Omega'} \lambda_s^r \leq 1 \qquad \forall s \in \{1, \ldots, M\} \tag{10}$$

$$\lambda_r^s \in \{0, 1\} \qquad \forall s \in \{0, \ldots, M\}, r \in \Omega \cup \Omega'$$

The objective (8) is to minimize the total cost, which includes the expected costs associated with the routes that have a specific priority. Constrains (9) guaranty that all customers will be visited at least once. This set of constraints are part of the well-known set covering formulation where it is assumed that it will always be cheaper to visit a customer only once and therefore relaxing the equality (2) still converges to the same solution. Constraints (10) restricts the total amount of routes that can be designated to a priority to at most one. This ensures that solutions with multiple routes that have the same priority are excluded.

The solution approach that we use to solve model (8) - (10) is the following:

1. We first improve the model by deriving an upper bound on the total supply of CVs and reducing the number of constrains and variables.

2. We relax the integrality constraint and create a restricted version of the model with an initial feasible solution consisting of all PV routes that visit a single customer.

3. The columns are generated by a heuristic DP algorithm. If the heuristic fails to find a single column with a negative reduced cost then an exact cohesive labeling algorithm is used to find any NG-routes that have a negative reduced cost if one exists.

4. If the solution is not integer we branch into two disjoint problems that exclude the fractional solution.

5. We select the active problem with the best bound, and go back to step 2.

3.3 Upper bound

In equation (10) we can see that there are as many priorities for routes as there are vehicles in the pool of CVs. Deriving an upper bound on the number of CVs will reduce the number of variables and constraints, making the model (8) - (10) more computationally tractable. At some point planning another route for an additional CV will be riskier and thus more expensive than utilizing a PV in the first stage. This is due to the monotonically increasing probability of failure with more CVs. When a route has the priority $s = 1$, it can only fail if the supply of CVs is 0, otherwise under all other realizations of ξ a route with first priority will not fail. But as the priority of a route decreases with respect to other routes the probability of failure increases.

Let d_{r^*} be the total distance traveled in a route $r^* \in \Omega'$ with priority s^*, and let $P(s^* \leq \xi)$ be the probability that the supply of CVs will be greater than or equal to s^*.

$$F + \beta d_{r^*} \leq P(s^* \leq \xi)(F' + d_{r*}) + \alpha(1 - P(s^* \leq \xi))(F + \beta d_{r^*})) \tag{11}$$

The lhs of inequality (11) is the first stage cost of route r^* if a PV is assigned the route. The rhs is the expected second stage cost of route r^* if it has priority s^*, this cost includes the probability that the route will need recourse actions.

Lemma 1. *If inequality (11) holds, then route r^* should not be left for the second stage.*

Proof. The lhs of this inequality is the first stage cost of a PV, and the rhs is the cost of a CV with priority s^*. It should be clear that if this was not true then we would reduce the cost by using a PV in the first stage for route r_{s^*} . □

By lemma 1 we can see that if a route does not meet the condition of inequality (11) then it should not be assigned to a CVs with a priority of $s*$, because of the risk involved with recourse actions that will be too expensive in comparison to the other option of simply using a PV to complete the route with no risk. Naturally, this does not mean that the same route with a higher priority (i.e., with a smaller s index) will not be worth considering for the second stage, since the risk is smaller if the route is prioritized as establish in the policy described in proposition 1. It follows that for any route there is a threshold index $\bar{s}$, such that the route should not be considered at the second stage unless the index is less than or equal to the threshold i.e., $s \leq \bar{s}$. We can establish a threshold probability from inequality (11) with $\bar{s}$.

Let $\bar{P}$ be the threshold probability equal to $P(\bar{s} \leq \xi) = \bar{P}$ for some route $r \in \Omega'$. By introducing $\bar{P}$ in inequality (11) we obtain the following equation:

$$\bar{P} = \frac{\alpha - 1}{\alpha - \frac{F'+d}{F+\beta d}} \tag{12}$$

The threshold probability $\bar{P}$ can be expressed as a function of the distance of any route and the fixed costs of the vehicles.

Lemma 2. *Given a route $r^* \in \Omega'$ with distance d^* and priority s^*. If $P(s^* \leq \xi)$ is strictly smaller than $\bar{P}$, then route r^* should not be considered for the second stage.*

Proof. It follows by moving the probability to the lhs of inequality (11) and from lemma 1 we obtain the result:

$$P(s \leq \xi) \leq \frac{(\alpha - 1)(F + \beta d)}{\alpha(F + \beta d) - (F' + d)}$$

$$P(s \leq \xi) \leq \frac{\alpha - 1}{\alpha - \frac{F'+d}{F+\beta d}}$$

□

Lemma 2 links the probability with the distance of any given route. The only element in equation (12) that is going to vary from one route to another is the distance which depends on the routing decisions that are made in the first stage. When routes have a longer distance the variable cost is more important and influences the threshold probability more. Conversely if the distance is small, the fixed cost has more weight in the cost structure and the threshold probability is determined by the fixed cost. It follows that we can establish bounds on $\bar{P}$ with the purpose of establishing a threshold that will be applicable to all routes. Consider the following ratio:

$$\bar{P} = \frac{\alpha - 1}{\alpha - \frac{1}{\beta}} \tag{13}$$

Lemma 3. *When the distance tends to infinity $d \to \infty$, $\bar{P}$ from equation (12) converges monotonically to (13).*

Proof. This result is easily obtained by observing the following fraction in equation (12):

$$\frac{F' + d}{F + \beta d}$$

□

Lemma 3 states that when the distance of a route increases to a sufficiently large value, $\bar{P}$ tends to 13. Consider the following ratio:

$$\bar{P} = \frac{\alpha - 1}{\alpha - \frac{F'}{F}} \tag{14}$$

Lemma 4. *When the distance tends to zero, $d \to 0$, $\bar{P}$ from equation (12) converges monotonically to (13).*

Proof. It follows from the proof of lemma 3. □

The ratio in equation (13) provides the bound needed for threshold probability when the distance is sufficiently large. Conversely the ratio in equation (14) provides the bound when the distance is equal to zero.

Figures 3 and 4 plot $\bar{P}$ as the distance grows, for different parameter values. We can also visualize both ratios (13) and (14). In figure 3, ratio (13) provides a lower bound for $\bar{P}$ and ratio (14) provides an upper bound. Conversely, in figure 4, ratio (14) provides a lower bound for $\bar{P}$ and ratio (13) provides an upper bound. The smallest ratio is the least restrictive, and bounds $\bar{P}$ below. Note that the ratio that determines the lower bound depends on the parameters for each instance.

Figures 5 and 6 plot the cumulative probability $P(s \leq \xi)$ and the priority s for a pool of 100 CVs. We can see on both figures that creating routes with priority greater than or equal to 5 is not going to be optimal, regardless of the distance of routes.

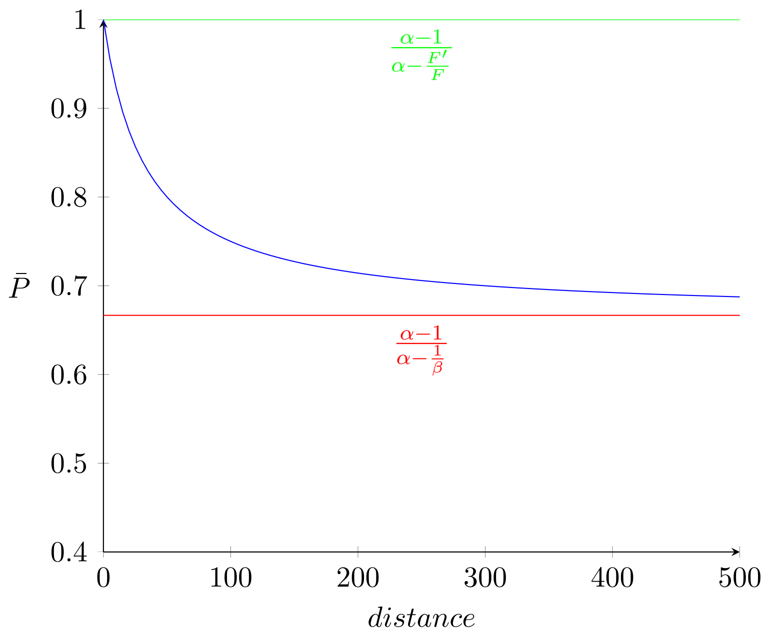

Figure 3: Threshold probability for $\frac{F'}{F} = 1$, $\frac{1}{\beta} = 0.5$ and $\alpha = 2$

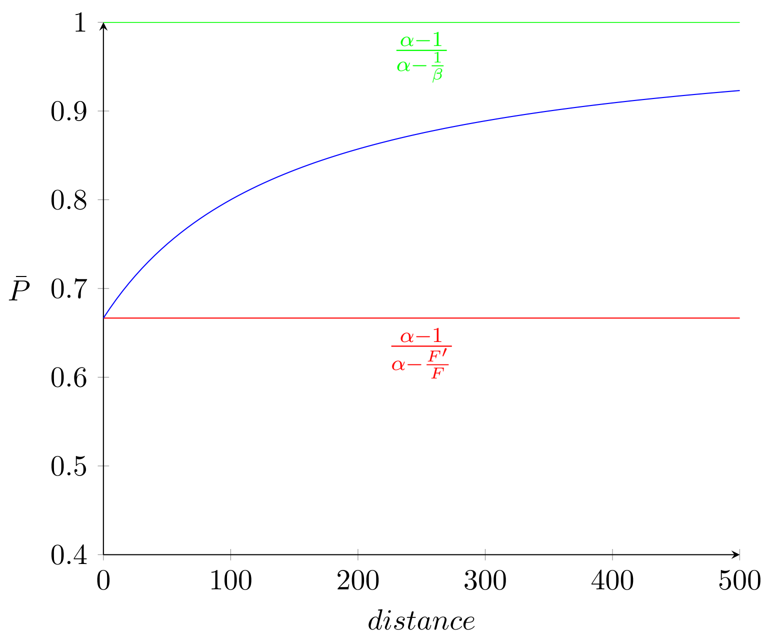

Figure 4: Threshold probability for $\frac{F'}{F} = 0.5$, $\frac{1}{\beta} = 1$ and $\alpha = 2$

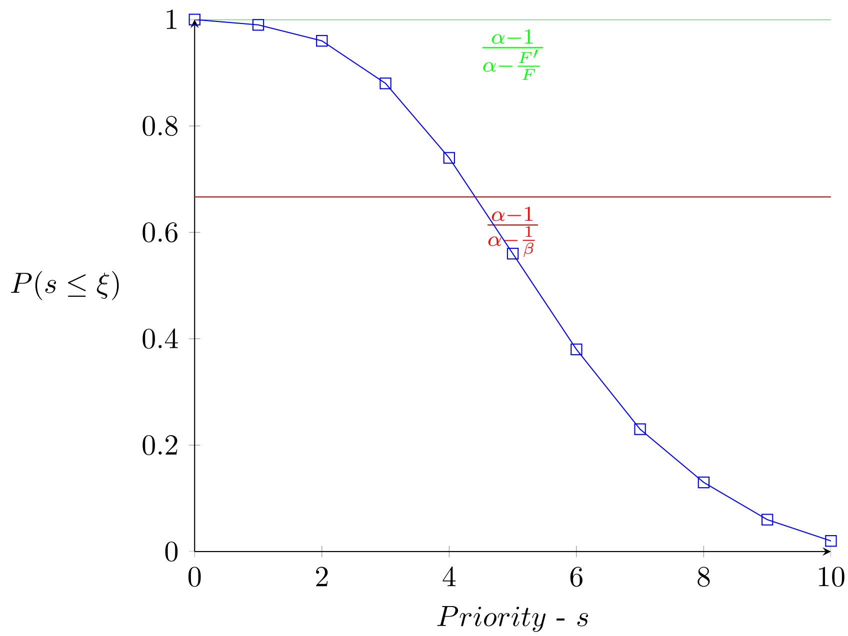

Figure 5: Cumulative binomial probability for each priority s with $\frac{F'}{F} = 1$, $\frac{1}{\beta} = 0.5$, $\alpha = 2$, $p = 0.05$, $M = 100$

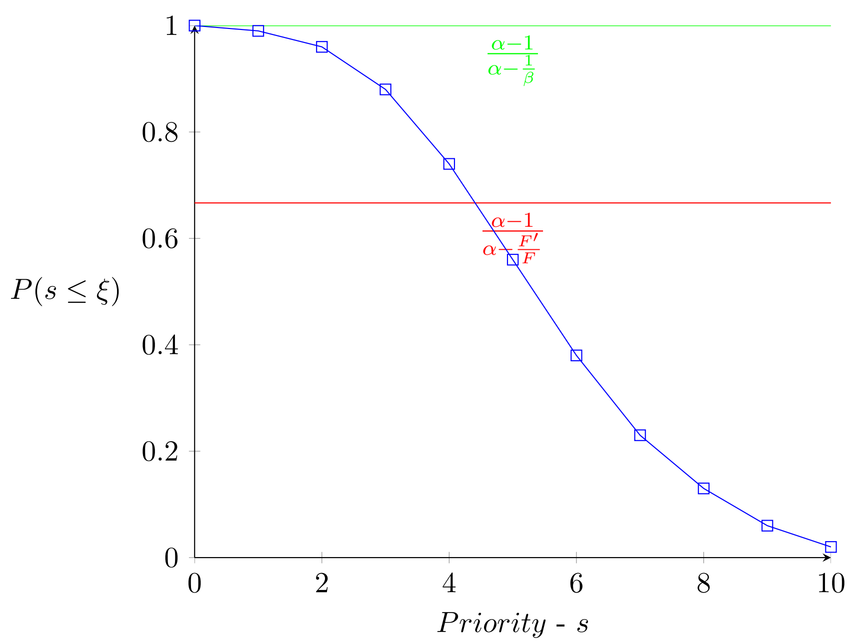

Figure 6: Cumulative binomial probability for each priority s with $\frac{F'}{F} = 0.5$, $\frac{1}{\beta} = 1$, $\alpha = 2$, $p = 0.05$, $M = 100$

Let $P^{-1}(x)$ be the discrete quantile function of $P(s \leq \xi)$. Let $\bar{M}$ be defined by:

$$\bar{M} = max\left[P^{-1}\left(\frac{\alpha-1}{\alpha-\frac{1}{\beta}}\right), P^{-1}\left(\frac{\alpha-1}{\alpha-\frac{F'}{F}}\right)\right] \tag{15}$$

Proposition 2. *$\bar{M}$ is an upper bound of the total CVs in the optimal solution.*

Proof. From lemma 3, and lemma 4, we see that the threshold probability converges either to (13) or (14), depending on the given parameters. The least restrictive of these two values will provide an upper bound. □

The upper bound $\bar{M}$ can replace M in the MP, reducing the amount of variables and constrains, thus making the pricing problem computationally less difficult.

It is imperative for the CSP to establish a threshold compensation, i.e., a point at which paying CDs more would lead to loses and not provide a cost reduction. The fleet of PVs has a known fixed and variable cost as well as a known penalty α, for hiring a PV in the future. The compensation that has to be decided upon is the fixed and variable cost for CVs. In any realistic representation of the problem the upper bound for CVs has to be greater than or equal to one, i.e., $\bar{M} \geq 1$. Otherwise it should be clear that CVs should not be used. It can be concluded from equations (13), (14) and (15), that β and F' should be fixed in a way such that $\bar{M} \geq 1$.

3.4 Pricing

Generating columns for the linear relaxation of the restricted master problem requires the solution of a set of sub-problems for each variable of type s in order to find variables with a negative reduced cost. If no variable with negative reduced cost is found and the solution is integer then the solution is optimal for the MP.

Let π and μ be the dual variables with respect to constraints (9) and (10) respectively. Then the pricing problems are the following:

$$\min_r \hat{c}_s^r = \{c_s^r - \sum a_i^r \pi_i + \mu_s : r \in \Omega \cup \Omega'\} \qquad \forall s \in \{0, \dots, \bar{M}\} \tag{16}$$

Where c_s^r equals the fixed and variable cost of route r as defined in the MP and the index $s = 0$ corresponds to the PV routes. From (16) we can see that there are $\bar{M} + 1$ problems that are required to find a negative reduced cost variable. In section 3.4.1 we present a simple exact dynamic programming algorithm that given the priority $s = \{0, \dots, \bar{M}\}$ generates columns with negative reduced cost. In section 3.4.2 we describe an exact cohesive labeling algorithm that generates columns for any priority $s = \{0, \dots, \bar{M}\}$, if one exists. In section 3.4.3 we illustrate how to implement Ng-routes and decremental state space relaxations to both exact algorithms. Finally, we outline a heuristic DP algorithm that creates elementary routes.

3.4.1 Dynamic programming

All problems in (16) are elementary shortest path problems with resource constraints (ESPPRC). Well-known dynamic programming methods called labeling algorithms exist to solve shortest path problems (Martinelli et al. (2014)). Labels represent partial paths that can return to the depot or continue visiting additional customers.

Let $\mathcal{L}_p^s = (i_p, w_p, t_p, d_p, \bar{\pi}_p, \mathscr{V}_p)$ be a label for variable of type s that describes path p, with current customer, load, time, distance, cumulative dual variables, and the set of unreachable customers respectively. For all possible loads, new labels are created by extending existing labels to feasible customers such that:

1. The customer is not contained in the set of unreachable customers $\mathscr{V}_p$.
2. Time windows are not violated.
3. Capacity of s is not exceeded, i.e., $w \leq Q'$ for $s = \{1, \ldots, \bar{M}\}$, and $w \leq Q$ otherwise.

The resource vector is then updated for the newly created label by extending the resources from its parent label. Columns for the MP can be created by extending the labels back to the depot. To facilitate the calculation of the reduced cost we define the following parameters that represent the variable and fixed cost for each sub-problem:

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

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

The reduced cost can be constructed later, when extracting solutions, by extending each label from its current customer to the depot and adding the corresponding dual variables. The reduced cost for a path p represented by a label L_p^s is equal to:

$$(d_p + c_{i_p 0})v_s - \bar{\pi}_p + \mu_s + F_s \tag{17}$$

The parameter v_s multiplies the distance, and the parameter F_s is the fixed cost depending on the priority that the path is assigned. The fixed costs in equation (17), i.e., μ_s and F_s do not alter the path structure in any way, and can be evaluated when extracting solutions from the labeling algorithm.

Dominance rules are used to reduce the number of labels such that labels that utilize less resources and have a smaller cost, dominate the labels that do not. More precisely for each $s = \{0, \ldots, \bar{M}\}$:

Dominance rules 3.1. *Label $\mathcal{L}_1^s$ dominates label $\mathcal{L}_2^s$ if the following rules are true:*

a) $i_1 = i_2$

b) $w_1 \leq w_2$

c) $t_1 \leq t_2$

d) $\mathscr{V}_1 \subseteq \mathscr{V}_2$

e) $d_1 v_s - \bar{\pi}_1 \leq d_2 v_s - \bar{\pi}_2$

Condition a) confirms that the current customer is the same for both labels and the remaining conditions guaranty that all customers that can be reached by label L_2^s can also be reached with a smaller cost by the dominant label L_1^s. The dominated label $\mathcal{L}_2^s$ is then removed form the algorithm and it is no longer considered. Dominance rules improve the performance of the labeling algorithm by significantly reducing the amount of labels.

Algorithm 1 describes the exact method to find the cheapest label w.r.t., equation (17) that follows dominance rules 3.1 for all s from 0 to the upper bound of crowd drivers. It builds a matrix $\mathcal{M}(w,i)$ that contains all labels in customer $i \in N$ with load w for all the possible loads. The first step is to create labels that extend to each customer from the depot. The procedures $feasible$ and $Dominance$ are simple procedures that verify the feasibility of an extension and the dominance rules respectively. The algorithm terminates by returning the cheapest column by iterating trough all labels in $\mathcal{M}$ that were not dominated and checking the cost.

Algorithm 1: Exact S-DP

```
Input  : π_i, μ_s, s   ∀i ∈ N,
Output: Column with smallest reduced cost ℒ_s
M(w,i) ← ∅,   ∀i ∈ N, w ∈ {0,...,Q_s} ;
M(q_i,i) ← {(i, q_i, max(t_i^e, d_0i) + t_i, d_0i, π_i, {i})},   ∀i ∈ N : feasible(0,i) ;
for w = 1 to Q_s do
    for i ∈ N do
        if w − q_i > 0 then
            for j ∈ N \ {i} do
                for l ∈ M(w − q_i, j) do
                    if feasible(j,i) then
                        if i ∉ V_l then
                            Create new label:
                            g ← (i, w, max(t_i^e, d_ji + d_l) + t_i, d_l + d_ji, π̄_l + π_i, {i} ∪ V_l);
                            dominated ← false;
                            Dominance ← rules 3.1;
                            for h ∈ M(w,i) do
                                Check dominance:
                                if Dominance(h,g) then
                                    dominated ← true;
                                    Break;
                                else if Dominance(g,h) then
                                    Delete dominated label:
                                    M(w,i) \ {h};
                                end
                            end
                            if not dominated then
                                M(w,i) ← M(w,i) ∪ {g};
                            end
                        end
                    end
                end
            end
        end
    end
end
return Cheapest label from M
```

3.4.2 Cohesive pricing

Solving the sub-problems separately can be time consuming. Furthermore, most labels that are created are the same, the dominance rules 3.1 vary for each sub-problem only for rule e), i.e., the variable cost, but all others are the same.

Dominance rules 3.2. *Label $\mathcal{L}_1$ dominates label $\mathcal{L}_2$ if the following rules are true:*

a) - d)

f) $d_1 v_s - \bar{\pi}_1 \leq d_2 v_s - \bar{\pi}_2 \quad \forall s \in \{0, \ldots, \bar{M}\}$

Dominance rules 3.2 guaranty that the label $\mathcal{L}_2$ is eliminated only if label $\mathcal{L}_1$ is dominant for all possible values of the parameter v_s for each sub-problem. Therefore if a label is created in one of the sub-problems then it will also be created in the cohesive pricing problem. Let L_c be the set containing all labels created in the cohesive sub-problem following dominance rules 3.2, and L_s be the set of labels created in sub-problem $\forall s \in \{0, \ldots, \bar{M}\}$ applying rules 3.1 then it is easy to see that:

$$L_c = \bigcup_{s=0}^{\bar{M}} L_s$$

In fact the parameters v_s can be ordered from smallest to largest. Let $\underline{v}$ and $\bar{v}$ be the smallest and largest values of parameters v_s respectively.

Proposition 3. *If condition $d_1\bar{v} - \bar{\pi}_1 \leq d_2\bar{v} - \bar{\pi}_2$, and condition $d_1\underline{v} - \bar{\pi}_1 \leq d_2\underline{v} - \bar{\pi}_2$ are true for any two labels $\mathcal{L}_1$ and $\mathcal{L}_2$, then all following conditions are also true:*

$$d_1 v_s - \bar{\pi}_1 \leq d_2 v_s - \bar{\pi}_2 \quad \forall s \in \{0, \ldots, \bar{M}\}.$$

Proof. By contradiction suppose there exist some $s^* \in \{0, \ldots, \bar{M}\}$ for which:

$$d_1 v_{s^*} - \bar{\pi}_1 > d_2 v_{s^*} - \bar{\pi}_2$$

The cases where $v_{s^*} = \bar{v}$ or $v_{s^*} = \underline{v}$ are trivial since this is stated to be true in the proposition. Therefore the value v_{s^*} is strictly larger than $\underline{v}$ and strictly smaller than $\bar{v}$ because by definition they are the smallest and the largest value in the vector v_s respectively. So there exists two values $f^1 > 0$ and $f^2 > 0$ such that $v_{s^*} = \bar{v} - f^1$ and $v_{s^*} = \underline{v} + f^2$
$(d_1\bar{v} - \bar{\pi}_1) - (d_2\bar{v} - \bar{\pi}_2) > (d_1 - d_2)f^1 \Rightarrow\Leftarrow (d_1\underline{v} - \bar{\pi}_1) - (d_2\underline{v} - \bar{\pi}_2) > (d_2 - d_1)f^2$ □

By proposition 3 we can now describe a new set of dominance rules that are simpler than rules 3.2, by reducing the the number of rules. Furthermore, they allows the solution of all sub-problems in a single one.

Dominance rules 3.3. *A label $\mathcal{L}_1$ dominates $\mathcal{L}_2$ if:*

a) - d)

g) $d_1\underline{v} - \bar{\pi}_1 \leq d_2\underline{v} - \bar{\pi}_2$

h) $d_1\bar{v} - \bar{\pi}_1 \leq d_2\bar{v} - \bar{\pi}_2$

Algorithm 2 shows the exact method for the cohesive pricing strategy. This simple algorithm is able to generate all the columns for the master problem by applying the improved dominance rules. When the load is higher than the CV capacity Q', it is no longer necessary to use dominance rules 3.3 because the only possible vehicle is a PV. Dominance rules 3.1 are then applied only for a PV and all labels that are dominated for $s = 0$ are removed from future extensions.

Algorithm 2: Exact C-DP

```
Input  : π_i, μ_s, s   ∀i ∈ N,
Output: Column with smallest reduced cost L_s
M(w, i) ← ∅,   ∀i ∈ N, w ∈ {0, ..., Q_s} ;
M(q_i, i) ← {(i, q_i, max(t_i^e, d_0i) + t_i, d_0i, π_i, {i})},   ∀i ∈ N : feasible(0, i) ;
for w = 1 to Q_s do
    for i ∈ N do
        if w − q_i > 0 then
            for j ∈ N \ {i} do
                for l ∈ M(w − q_i, j) do
                    if feasible(j,i) then
                        if i ∉ V_l then
                            Create new label:
                            g ← (i, w, max(t_i^e, d_ji + d_l) + t_i, d_l + d_ji, π̄_l + π_i, {i} ∪ V_l);
                            if w > Q' then
                                Dominance ← rules 3.1 for s = 0;
                            else
                                Dominance ← rules 3.3;
                            end
                            dominated ← false;
                            for h ∈ M(w, i) do
                                Check dominance:
                                if Dominance(h, g) then
                                    dominated ← true;
                                    Break;
                                else if Dominance(g, h) then
                                    Delete dominated label:
                                    M(w, i) \ {h};
                                end
                            end
                            if not dominated then
                                M(w, i) ← M(w, i) ∪ {g};
                            end
                        end
                    end
                end
            end
        end
    end
end
return Cheapest label from M
```

3.4.3 Ng-routes and decremental state-space relaxation (DSSR)

Dominance rule *d*) imposes the elementary constraints on the considered paths, which make the subproblems considerably more complex to solve. A common relaxation is to drop rule *d*) and allow cycles, where customers can be present multiple times on the paths obtained. Columns with cycles are always more expensive and are eliminated in the optimal solution. Therefore, the relaxation improves the time complexity of the pricing problems but weakens the bound of the master problem. A compromise between the two is achieved by ng-routes relaxation proposed by Baldacci et al. (2011). For each customer an ng-set of nearest customers including the current customer, $N_i \subseteq N$ for all $i \in N$, is kept. Dominance rule *d*) is relaxed by only applying it to the customers that are contained in the ng-sets.

Let $\mathcal{L}_p$ be a label of path $p = (0, i_1, \ldots, i_p)$ that starts at the depot and visits a sequence of customers and ends at the current customer i_p, let $\mathcal{V}_p$ be the set of prohibited customers and i_{p+1}

be a feasible customer that can be reached form label $\mathcal{L}_p$ such that $i_{p+1} \notin \mathcal{V}_p$. To extend label $\mathcal{L}_p$ to customer i_{p+1} and create label $\mathcal{L}_{p+1}$ the set of prohibited customers are updated in the following way:

$$\mathcal{V}_{p+1} = \left\{\mathcal{V}_p \cap N_{p+1}\right\} \cup i_{p+1}$$

The initial set of unreachable customers is empty since from the depot all customers are feasible. It is clear that if the ng-set contains all customers then we end up with elementary routes. As the ng-sets become smaller non necessarily elementary routes can be formed as long as they are not in the ng-sets. In our implementation we use the 10 nearest customers from each customer such that the set $|N_i| = 10$ for all $i \in N$. Decremental state space relaxations (DSSR), further improve the performance of the pricing algorithms by starting with empty ng-sets and adding customers as needed until the cheapest route has no cycles that violate the ng restrictions. For a more detailed description of the implementation of ng-routes and DSSR the reader is referred to Baldacci et al. (2011), and Martinelli et al. (2014).

3.4.4 Heuristic DP

At every iteration, it is not necessary to find the best column to add to the restricted master problem. The search can be sped up by applying a heuristic method to find a column with a negative reduced cost. The heuristic implementation of S-DP, and C-DP is done by dropping the difficult rule d completely while only allowing elementary paths to be formed. Labels are extended to unvisited customers and can be deleted by following dominance rules $a) - c)$, and $f)$ so that only a few paths remain. This simple heuristic creates most columns for the algorithm and only when this heuristic fails to find a negative column the exact method is used.

3.5 Branching

The branching strategy chosen in this implementation is the most fractional value first. Branching is done primarily in two ways:

1. Total number of PV and CV;

2. The arcs that connect each note can either be 1 or 0.

When the total number of vehicles is fractional we branch up and down so that the fractional solution is excluded. If the number of vehicles is integer then we branch on the variables that traverse through an arc. The sum of variables that pass through an arc (i, j), in any feasible solution, must be either 1 or 0. If the sum of variables is fractional then we branch on the sum of such variables.

The DP algorithm can be easily modified for both branching strategies. The first branching strategy does not affect the paths. The second branching strategy requires that we check at each extension whether it is possible. If the current problem in the branching tree has the condition that the arc $(i, j) = 0$, or at any parent node, then the extensions in the pricing problem that go from i to j are not allowed, otherwise the extension is allowed.

4 Computational results

To evaluate the impact of a stochastic fleet on traditionally deterministic problems, we modify the well-known Solomon instances C1, R1, and RC1 with 25, 50 and 100 customers by including an additional fleet of CVs.

Parameters	Description	Value
Q	Capacity of PV	200
Q'	Capacity of CV	100
F	PV fixed cost	100
F'	CV fixed cost	50
α	Second stage penalty	2.0
$\frac{1}{\beta}$	CV variable cost ratio	0.5
p	Probability of success in binomial distribution	0.05
M	Size of the pool of CVs	100
$p \times M$	Average number of CVs	5
$\bar{M}$	Upper bound on CVs	4

Table 2: Parameters for the base case

Table 2 shows the values for the parameters in the base case. PVs have the same capacity for the Solomon instances of 200, but we added a fixed cost of 100 and a variable cost $\beta = 2.0$. The pool of crowd vehicles has a total size of 100, with an average value of 5 vehicles. The fixed and variable costs ($F' = 50$, $\frac{1}{\beta} = 0.5$) for CVs are half of the cost for PVs to be consistent with the capacity of 100. Finally the upper bound $\bar{M} = 4$ is calculated by replacing the values for the parameters in equation (15).

In the remainder of this section, we present the computational results of the branch and price algorithm. In section 4.1, we evaluate the value of the stochastic solution for our model, compared with a deterministic model that simplified the stochastic elements by replacing the stochastic parameter ξ with its expected value. In section 4.2 we report the cost savings that can be achieved by implementing CVs. In section 4.3 we compare the improvements by cohesive pricing compared to the exact DP algorithm that solves each subproblem independently. Furthermore we present result for all instances with up to 100 customers and report times and cost obtained. Finally, in section 4.4 we show how the bound obtained in section 3.3 can be used to derive interesting insights on the compensation of CDs. The algorithms were all implemented in Java SE 1.8.0 and executed in a Linux-CentOS 7 system with an Intel core E5-2683 at 2.1GHz, and 16GB of ram. The solver CPLEX 12.9 was used for the linear problems.

4.1 Value of the stochastic solution (VSS)

A naive solution approach to problems with uncertainty is to simply get the expected value of the random variable and solve the deterministic model, i.e., assuming that the random variable is always equal to its expected value; this problem is called the expected value problem (EVP). In reality variations occur in the random variable and expensive recourse actions need to be taken. The problem that considers the variations of the random variable based on probabilistic information is the recourse problem (RP), which minimizes the routing cost and the cost of recourse actions. To construct the EVP for the base case, we simply set the number of crowd vehicles to 5, which is the expected supply of CV. The expected value of the objective function for the EVP solution is then evaluated by considering the variations of the random variable and the recourse cost. The value for the RP solution is obtained by simply solving model 8-10. The value of the stochastic solution (VSS), which is what is gained by solving the stochastic problem, is simply the difference of these two values:

$$VSS = \mathbb{E}_{\xi}[EVP] - RP$$

In table 3 we assess the value of the stochastic model (RP) compared to the EVP by performing a sensitivity analysis on the parameters. We solved all instances with 25 customers and report the average values in each row. The first column of the table show the parameter change from the base case. For the EVP and the RP we present on each column the total average cost, excluding the recourse cost, next column shows the average recourse cost $\mathcal{Q}(\lambda)$ for the solution obtained. Next column reports the total average cost with the cost of recourse actions, and the last two columns shows the average CVs and PVs used in all instances, for EVP and RP.

The last three columns of table 3 report the average VSS, the average percentage of the VSS and the standard deviation of the average percentage of the VSS for all instances respectively. We can see that the VSS is more significant as the solution for the EVP utilizes more CVs and it becomes less important when few CVs are utilized in the EVP. If the CSP plans on replacing the deterministic fleet with the pool of stochastic vehicles then it is imperative to solve the stochastic model. The VSS is a non trivial amount for all cases, but the largest changes are caused by variations in the variable cost and the recourse penalty.

Table 3: Value of the stochastic solution for 25-customer instances

	EVP					RP					VSS		
Ins	Cost	$\mathcal{Q}(\lambda)$	Total c.	CV	PV	Cost	$\mathcal{Q}(\lambda)$	Total c.	CV	PV	Value	%	σ
Base	715.75	213.93	929.7	4.45	0.45	763.81	85.68	849.5	2.8	1.3	80.2	11.33	11.0
$\alpha = 1.5$	715.75	142.63	858.37	4.45	0.45	738.11	78.66	816.78	3.31	1.0	41.59	6.26	6.88
$\alpha = 2.5$	715.75	285.25	1001.0	4.44	0.44	821.23	54.32	875.54	2.31	1.76	125.46	17.03	15.18
$\frac{1}{\beta} = 0.25$	787.72	402.14	1189.86	4.52	0.45	893.02	215.29	1108.35	3.48	0.97	81.51	8.57	8.84
$\frac{1}{\beta} = 0.75$	668.25	78.55	746.8	2.51	1.41	689.92	44.9	734.81	2.0	1.76	12.0	1.37	2.29
$\frac{F'}{F} = 0.25$	604.31	232.69	837.0	4.48	0.45	673.6	104.15	777.74	3.21	1.14	59.26	9.41	9.55
$\frac{F'}{F} = 0.75$	813.98	124.29	938.26	3.07	1.14	834.56	72.77	907.33	2.24	1.62	30.93	3.03	4.39
$Q' = 75$	804.23	159.12	963.35	3.0	1.41	859.14	45.55	904.69	1.86	2.03	58.66	6.77	7.29
$Q' = 125$	655.47	230.52	885.99	4.41	0.17	693.52	126.17	819.69	3.69	0.59	66.31	7.53	11.11

4.2 Value of crowd vehicles (VCV)

To determine whether implementing CVs produces any cost savings, we compare the objective value of the solution obtained by our model, with the solution value of the problem without CVs (PVP), i.e., PV only. When PVs are used exclusively, there is no risk considered, hence no recourse actions, and no additional cost. Conversely, when CVs are included in the set of options that the CSP has, risk of not having enough supply is considered leading to additional cost. To evaluate how much cost savings is achieved by implementing CVs in the routing problem, we compare the solution value of a problem with PVs only, with the solution value of our framework.

The value of crowd vehicles is defined as the difference between the solution value of the PVP problem, and the solution value of the solution of RP as defined in section 4.1.

In table 4, we solve the same instances of table 3, described in section 4.1, with the two relevant problems, i.e., PVP and RP. The last three columns report respectively, the average VCV for the instances, the average percentage of the VCV w.r.t. the PVP value, and the standard deviation of the percentage. For the PVP problem, we report the average total cost (Total c.) and the average PV used in the solution. For the RP problem, we report the total average cost, average CV, and average PV.

Table 4: Value of CVs for 25-customer instances

	PVP		RP			VCV		
Ins	Total c.	PV	Total c.	CV	PV	Value	%	σ
Base	1082.57	3.86	849.5	2.8	1.3	233.08	19	13
$\alpha = 1.5$	1082.57	3.86	816.78	3.31	1.0	265.8	21.44	14.6
$\alpha = 2.5$	1082.57	3.86	875.54	2.31	1.76	207.02	16.75	11.5
$\frac{1}{\beta} = 0.25$	1778.94	3.86	1108.35	3.48	0.97	670.62	35.12	14.4
$\frac{1}{\beta} = 0.75$	849.29	3.86	734.81	2.0	1.76	114.76	11.86	9.3
$\frac{F'}{F} = 0.25$	1082.57	3.86	777.74	3.21	1.14	304.82	25.90	12.8
$\frac{F'}{F} = 0.75$	1082.57	3.86	907.33	2.24	1.62	175.24	13.62	11.9
$Q' = 75$	1082.57	3.86	904.69	1.86	2.03	177.88	13.39	12.8
$Q' = 125$	1082.57	3.86	819.69	3.69	0.59	262.89	22.22	10.6

The parameters that have the greatest impact on the VCV are the variable cost of CV, fixed costs of CV, and the capacity CV. Naturally, as CV have more capacity available, they can perform longer routes, and hence are more efficient.

4.3 Performance

We solved all instances with the branch and price algorithm and tables 3-5 show the results for instances with 25, 50 and 100 customers respectively. All 29 instances were solved for 25 customers, 19 out of 29 for 50 customer instances, and 5 were solved optimally for instances with 100 customers.

We can see that CVs are a part of all solutions, some requiring only 1 CV and other instances the maximum number allowed. In smaller instances PVs are replaced completely since only a few vehicles are needed to complete the delivery requests. Larger instances require more vehicles to complete all deliveries and the upper bound becomes more restrictive.

Table 5: Results for 25-customer instances

Instance	Cost	Time (s)	CV	PV	Total
C101	617.18	10.7	3	1	4
C102	615.03	422.55	3	1	4
C103	611.12	282.87	1	2	3
C104	606.29	1402.25	1	2	3
C105	617.18	16.17	3	1	4
C106	617.18	11.29	3	1	4
C107	604.55	3.5	1	2	3
C108	604.55	9.63	1	2	3
C109	604.55	20.97	1	2	3
R101	1622.97	1.3	4	4	8
R102	1396.0	16.8	4	3	7
R103	884.12	61.1	4	0	4
R104	794.14	35.6	4	0	4
R105	1134.28	3.26	4	1	5
R106	993.09	40.4	4	1	5
R107	802.75	51.19	4	0	4
R108	781.71	240.3	4	0	4
R109	850.89	4.61	4	0	4
R110	835.08	41.55	4	0	4
R111	815.67	65.51	4	0	4
R112	778.83	442.77	4	0	4
RC101	1038.62	4.2	2	2	4
RC102	947.16	17.64	2	2	4
RC103	924.54	30.44	2	2	4
RC104	875.52	17.0	2	2	4
RC105	964.10	4.72	2	2	4
RC106	942.34	22.14	2	2	4
RC107	881.70	41.76	2	2	4
RC108	874.25	71.12	2	2	4

Table 6: Results for 50-customer instances

Instance	Cost	Time (s)	CV	PV	Total
C101	1174.76	408.65	3	3	6
C102	1174.54	58788.3	3	3	6
C105	1174.76	1641.46	3	3	6
C106	1174.76	925.42	3	3	6
C107	1172.0	1966.2	3	3	6
R101	2815.28	35.96	4	7	11
R102	2384.16	221.15	4	6	10
R103	1920.78	1547.54	4	4	8
R105	2235.7	708.19	4	5	9
R106	1879.46	980.6	4	3	7
R107	1639.01	6375.52	4	2	6
R109	1839.67	6251.07	4	3	7
R110	1682.27	4169.51	4	3	7
RC102	1883.5	26295.2	4	3	7
RC103	1745.9	2891.95	2	4	6
RC104	1555.68	495.23	2	4	6
RC106	1741.59	2649.08	2	4	6
RC107	1613.1	3624.5	2	4	6
RC108	1551.1	16914.6	2	4	6

Table 7: Results for 100-customer instances

Instance	**Cost**	**Time (s)**	**CV**	**PV**	**Total**
C101	2626.65	7157.37	3	8	11
C105	2617.93	16851.2	3	8	11
C107	2617.58	44567.4	3	8	11
R101	4723.86	152.21	4	15	19
R102	4203.5	272.12	4	13	17

Instances with 25 customers were tested with both pricing algorithms S-DP and C-DP. Table 8 shows a comparison between the two and reports the number of columns (col.) and time at the root node (T(s)), and the total number of columns and total time to find the optimal solution. The last two columns show the difference in total time and the percentage of improvement in time respectively when using C-DP. Both algorithms create roughly the same number of columns, but C-DP is much faster since it does not create labels redundantly.

Table 8: Comparison of DP algorithms for 25-customer instances

	S-DP				C-DP					
	Root		Total		Root		Total			
Ins	col.	T(s)	col.	T(s)	col.	T(s)	col.	T(s)	ΔT(s)	%T
C101	444	4.13	1525	18.4	374	0.83	1475	4.57	13.83	75.16
C102	512	13.59	11841	674.76	453	2.49	10083	277.57	397.19	58.86
C103	477	26.82	4704	347.39	479	6.96	4108	129.59	217.8	62.70
C104	500	43.98	14965	2926.87	499	16.7	14624	1455.51	1471.36	50.27
C105	436	4.07	2036	34.63	399	0.8	2114	8.4	26.23	75.74
C106	471	4.75	1442	18.92	370	0.78	1319	4.6	14.32	75.69
C107	475	5.2	706	10.48	374	0.87	648	1.56	8.92	85.11
C108	596	12.5	1062	25.50	492	1.76	966	4.82	20.68	81.10
C109	498	14.61	1272	35.11	427	3.15	1155	11.64	23.47	66.85
R101	246	2.66	246	2.7	219	0.54	219	0.55	2.15	79.63
R102	310	17.04	419	39.07	293	1.53	435	5.29	33.78	86.46
R103	455	58.36	1396	187.61	461	5.19	1199	33.95	153.66	81.90
R104	396	71.1	444	96.0	434	8.46	560	16.21	79.8	83.12
R105	369	6.01	541	11.14	303	0.75	418	1.28	9.86	88.51
R106	359	28.66	1493	88.01	357	3.73	1086	16.97	71.04	80.72
R107	485	68.26	1011	150.61	452	10.54	827	24.54	126.07	83.71
R108	458	65.8	3461	894.5	481	13.37	3448	184.76	709.74	79.34
R109	346	17.42	346	17.44	422	2.29	422	2.3	15.14	86.81
R110	326	32.68	1139	108.15	432	5.6	933	18.12	90.03	83.25
R111	400	49.35	864	105.97	427	8.08	799	23.91	82.06	77.44
R112	428	66.69	4606	1291.1	425	14.88	5044	370.19	920.91	71.33
RC101	392	3.66	992	10.23	332	0.75	1222	4.1	6.13	59.92
RC102	417	9.05	934	30.78	359	1.47	991	6.76	24.02	78.04
RC103	471	16.93	1798	91.76	411	2.73	1692	20.15	71.61	78.04
RC104	455	19.94	643	32.55	415	4.77	624	9.52	23.03	70.75
RC105	387	6.88	596	10.24	389	1.47	512	2.42	7.82	76.37
RC106	424	7.45	1611	37.92	364	1.45	1260	7.82	30.1	79.38
RC107	423	21.94	1995	93.11	338	3.92	1396	21.61	71.5	76.79
RC108	375	33.83	1732	147.75	327	7.98	960	41.74	106.01	71.75

4.4 Compensation

The compensation can have a significant impact in the supply of vehicles. Based on the well-known laws of supply and demand, the more the CSP compensates drivers, the more likely it is they will be available. Conversely, the less CV are paid, the smaller the supply and the more uncertain the delivery process will be. Table 9 shows the maximum number of CV (i.e., $\bar{M}$) that is feasible for any delivery problem as the compensation changes represented as a ratio of the cost of a PV. Columns show different values for the probability of success in a binomial distribution with 100 CVs. The parameter $\alpha = 2.0$ and equation (15) are used to compute the upper bound. We can see that the more expensive CVs are, the less likely they will be used in a solution. This table can provide insights to the compensation of CV when probabilistic information is known as well as the sensitivity of the supply and compensation. For example, when the compensation of a CV is equal to 50% of the PV cost and $p = 0.05$, the CSP can plan on using only 5 CVs before incurring in a loss. However, if the CSP increases the compensation to 60% and as a consequence the supply shifts to $p = 0.07$, then the CSP can feasibly plan to use up to 7 CVs.

$\frac{CV\,cost}{PV\,cost}$	Probability of success for a pool of 100 CVs								
	0.02	0.03	0.04	0.05	0.06	0.07	0.08	0.09	0.1
0.9	1	1	2	3	4	5	5	6	7
0.8	1	2	3	4	5	5	6	7	8
0.7	1	2	3	4	5	6	7	8	9
0.6	2	3	4	5	6	7	7	8	9
0.5	2	3	4	5	6	7	8	9	10
0.4	2	3	4	5	6	7	8	9	10
0.3	2	3	4	5	6	7	8	9	10
0.2	2	3	4	5	7	8	9	10	11
0.1	2	3	5	6	7	8	9	10	11
0	2	4	5	6	7	8	9	10	11

Table 9: Upper bound on the number of CVs for different compensation levels

5 Conclusions and future research

Crowd-shipping is a new phenomenon that is still at a conceptual stage requiring quantitative studies to help evaluate the feasibility of different approaches. In this paper, we presented the problem of routing a mixed fleet of deterministic company vehicles and stochastic crowd vehicles and formulated a set partitioning model improved by adding variables that enable the solution of the second-stage problem. We developed a policy to dynamically allocate routes to crowd drivers as they become available throughout the day. We derived upper bounds that strengthen the formulation and gave interesting insights into the compensation of drivers. We presented a branch and price algorithm that is able to solve large instances with 100 customers and a pool of 100 CVs. The cohesive pricing dynamic programming algorithm was compared with the traditional DP algorithm and it was shown to be much faster because it avoids creating redundant labels at every extension. We computed the VSS and showed that it can be quite significant for small problems with 25 customers, it can be as large as 17% of the total delivery cost. Finally, we showed that there can be significant cost savings of 35 % by implementing crowd drivers.

Future research is needed to gain additional insights in the effect that uncertain vehicle properties have in routing problems in contrast to traditional models where uncertainty originates from customers or road conditions. In this paper we have studied the uncertain supply of vehicles, but many other problems remain unsolved. We assumed that CDs accept the assignment of routes by the CSP, since the most expensive and thus more profitable routes are assigned to the first available drivers. Nevertheless not all drivers are solely motivated by profit, some might have specific preferences such as smaller routes, duration of routes or even the neighborhood in which routes are could influence the drivers choices of routes. Accounting for these preferences is an interesting avenue of research to consider. There are many opportunities to extend this study and explore different vehicle properties as well, e.g., the capacity of vehicles could be stochastic, drivers might also have stochastic destinations and so on. As the sharing economy becomes more ubiquitous in society, mathematical models that provide quantitative results will be more important to evaluate different delivery concepts.

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 and Polytechnique Montréal.

References

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.

A. M. Arslan, N. Agatz, L. Kroon, and R. Zuidwijk. Crowdsourced delivery: A dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1):222–235, 2019.

R. Baldacci, A. Mingozzi, and R. Roberti. New route relaxation and pricing strategies for the vehicle routing problem. *Operations Research*, 59(5):1269–1283, 2011.

A. Barr and J. Wohl. Exclusive: Walmart may get customers to deliver packages to online buyers. *REUTERS–Business Week*, 2013.

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. *Optimization Online*, pages 07–6142, 2017.

F. Errico, G. Desaulniers, M. Gendreau, W. Rei, and L.-M. Rousseau. The vehicle routing problem with hard time windows and stochastic service times. *EURO Journal on Transportation and Logistics*, 7(3):223–251, 2018.

M. Ertz, F. Durif, and M. Arcand. Collaborative consumption: Conceptual snapshot at a buzzword. *Journal of Entrepreneurship Education*, 19(2):1–23, 2016.

M. Gendreau, O. Jabali, and W. Rei. Chapter 8: Stochastic vehicle routing problems. In *Vehicle Routing: Problems, Methods, and Applications, Second Edition*, pages 213–239. SIAM, 2014.

Ç. Koç, T. Bektaş, O. Jabali, and G. Laporte. Thirty years of heterogeneous vehicle routing. *European Journal of Operational Research*, 249(1):1–21, 2016.

G. Laporte, F. Louveaux, and H. Mercure. The vehicle routing problem with stochastic travel times. *Transportation science*, 26(3):161–170, 1992.

X. Li, S. C. Leung, and P. Tian. A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem. *Expert Systems with Applications*, 39 (1):365–374, 2012.

R. Loxton, Q. Lin, and K. L. Teo. A stochastic fleet composition problem. *Computers & Operations Research*, 39(12):3177–3184, 2012.

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 G. Laporte. Crowd-shipping with time windows and transshipment nodes. *Computers & Operations Research*, 113:104806, 2020.

R. Martinelli, D. Pecin, and M. Poggi. Efficient elementary and restricted non-elementary route pricing. *European Journal of Operational Research*, 239(1):102–111, 2014.

Q. Mu, Z. Fu, J. Lysgaard, and R. Eglese. Disruption management of the vehicle routing problem with vehicle breakdown. *Journal of the Operational Research Society*, 62(4):742–749, 2011.

H. N. Psaraftis, M. Wen, and C. A. Kontovas. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31, 2016.

M. Salavati-Khoshghalb, M. Gendreau, O. Jabali, and W. Rei. An exact algorithm to solve the vehicle routing problem with stochastic demands under an optimal restocking policy. *European Journal of Operational Research*, 273(1):175–189, 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.

A. Sundararajan. *The sharing economy: The end of employment and the rise of crowd-based capitalism.* Mit Press, 2016.

D. Teodorovic, E. Krcmar-Nozic, and G. Pavkovic. The mixed fleet stochastic vehicle routing problem. *Transportation Planning and Technology*, 19(1):31–43, 1995.