Crowd-shipping under Uncertainty: Models and Solution Approaches

M. Gendreau[a,b,*], F.A. Torres Durán[a,b], W. Rei[a,c]

[a] CIRRELT, Montreal, Canada
michel.gendreau@cirrelt.net, fabiantodu@gmail.com
[b] Département de mathématiques et de génie industriel
Polytechnique Montréal, Montreal, Canada
[c] Département d'Analytique, Opérations et Technologies de l'Information
École des Sciences de la Gestion, Université du Québec à Montréal, Montreal, Canada
rei.walter@uqam.ca
[*] Corresponding author

Extended abstract submitted for presentation at the 11th Triennial Symposium on Transportation Analysis conference (TRISTAN XI)
June 19-25, 2022, Mauritius Island

January 15, 2022

Keywords: Crowd-shipping; crowd drivers; stochastic programming; dynamic programming; column generation

1 INTRODUCTION

The extensive development of on-line retailing and of collaborative consumption systems, such as Uber, has led large retailers to consider new ways of making deliveries to their on-line consumers. One of them is *crowd-shipping*, which consists in having goods bought on-line delivered to final customers by other customers or other *crowd drivers* (CD). In either case, individuals who are not employed by the retailer or one of its logistics subcontractors are used to make deliveries. This tendency to move towards e-commerce and crowd-shipping has been exacerbated by the COVID-19 pandemic and the associated risks when visiting brick-and-mortar retail stores.

The most successful crowd-shipping platform (CSP) is Amazon Flex that now operates in the US and Canada, and in over 100 cities (AmazonFlex, 2021). Amazon has solved some of the challenges by creating a program where individuals can sign up a priori only if they meet some basic requirements. Walmart recently started its own CSP called Walmart Spark Delivery. For a review of the different CSPs and the scientific literature on crowd-shipping, see Alnaggar *et al.* (2021).

After crowd-shipping was conceptualized by Amazon and other companies, a first quantitative study of crowd-shipping was presented by Archetti *et al.* (2016). The authors introduce the Vehicle Routing Problem with Occasional Drivers (VRPOD). They formulate a deterministic and static model where a set of delivery requests have to be fulfilled from a central depot either by an unlimited fleet of vehicles driven by professional drivers (employed either directly by the CSP or through a logistics partner), in the following we refer to these vehicles as *Professionnal Vehicles* (PV), which complete closed routes, or by a set of CDs that are willing to deliver some packages while they travel towards their specific destination. The compensation of CDs could be based on different strategies. A mixed integer program (MIP) is introduced to solve small instances, and a multi-start heuristic is proposed to provide solutions to larger instances.

It is shown that significant cost reductions can be achieved by employing CDs compared to conventional delivery means.

A main question that arises from Archetti *et al.* (2016) is how can CDs be applied in a dynamic and/or stochastic setting? In practice, CDs could arrive dynamically throughout the day; thus, the total number of CDs is unknown until late in the day. Furthermore, CDs could reject or accept routes. In Gdowska *et al.* (2018), the authors consider the possible rejection of delivery tasks by ODs. The probability of rejecting a delivery request by a CD is viewed as independent from other delivery requests.

In this paper, we consider a general crowd-shipping situation, in which the participation of CDs and the acceptance of CD routes is uncertain. We derive a general stochastic model that can be applied to different variants of this basic situation, and we develop effective algorithms that can be used to tackle variants of this model.

2 PROBLEM DESCRIPTION

We consider a CSP that allows individuals to sign up if they are interested in delivering parcels with their personal vehicles. This provides the CSP with a pool of CDs. We assume that each vehicle has at least a minimum pre-specified capacity Q' available. While CDs could arrive at various times, we assume that they all show up at the depot early enough to handle the delivery tasks at hand. Therefore, in our basic situation, the only source of stochasticity that we consider is the availability of CDs for the current planning period, which is given by a random variable ξ, which follows a binomial distribution on the interval $\{0, \ldots, M\}$, with given probability p.

The CSP has a known list of delivery requests by online customers. Each request has a specific demand or quantity and must be delivered within a specific time window. To complete the delivery tasks, the CSP can use CDs or its own fleet of PVs with capacity $Q > Q'$. Routes must be planned for both PVs and the vehicles of CDs.

The compensation scheme used to pay CDs is flexible and consists of a fixed cost that every driver gets by delivering at least one parcel, a variable cost associated with the total distance traveled on the route performed, and a reward that is paid for each parcel delivered. These three types of compensations can be applied in any proportion to incentivize different types of behaviors in CDs. Similarly, there are three costs associated with PV routes, a (larger) fixed cost, a (larger) variable cost, and a (similar) cost to service each customer.

In a second variant of the problem, we consider a setting in which the territory served by the CSP is divided into geographical sectors. In this variant, CDs are characterized by their destination sector. They perform open routes from the depot to their destination sector, subject to capacity and route duration constraints. CDs' availability is also defined by sector.

3 MODELS

The basic structure of the models for both problem variants described above is that of a two-stage stochastic programming problem with recourse. We now focus on the first variant in which we consider the complete territory served by the CSP. The main decisions are all taken in the first-stage problem, in which we create a set of routes serving all customer requests. The requests are split into two subsets: one to be served by CDs and the remainder by PVs, and routes are created for both PVs and CDs. The cost of routes for PVs is deterministic and easy to compute. Conversely, routes planned for CDs are not deterministic and their cost depends on future events at the second stage, i.e., the realization of the random parameter ξ. However, since we assume that the probability distribution of ξ is available, we can derive the expected cost of each route. Let r be a route created for a CD. If a CD is available to complete route r, then the route is completed, and the compensation is paid to the CD. Conversely, if there is no CD to complete the route a recourse action has to be taken. The recourse cost of a route in the second stage is the incremental cost of serving this route with a PV instead of a CD. It is computed as the cost of performing this route with a PV times a penalty α (to represent the additional inconvenience associated with the late assignment of the route to a PV) less the cost of having the route served by a CD.

The second-stage problem requires assigning, for any given value of ξ, the available CDs to the constructed routes. We assume that the CSP decides the assignment of routes to available CDs to minimize its own costs, which amounts to assigning CDs in priority to the routes requiring expensive recourse actions. We also assume that CDs complete the routes that are assigned to them. Although this assumption might seem limiting, routes requiring more expensive recourse actions are also better compensated for CDs and should be more attractive to them. We showed that, if the reward per delivery is trivial, this assumption is equivalent to assuming that CDs will choose better compensated routes.

The overall formulation of the problem thus corresponds to a fairly standard set partitioning formulation of the VRPTW with two types of vehicles (PVs and CDs' vehicles), with one major difference: the addition of an expected recourse term that accounts for the additional costs incurred for routes originally designed for CDs that will have to be performed by PVs, if the participation of CDs is not large enough. The computation of this expected recourse term relies on additional binary variables that allow us to choose the priority of routes assigned to CDs and thus the probability that they will be actually performed by a CD.

In the second variant of the problem, the set partitioning formulation also accounts for the various sectors that are served.

4 SOLUTION METHODS

To solve the proposed models, we first develop exact solution methods based on column generation and Branch-and-Price. Considering the issues relative to the priorities assigned to the routes, one could expect that the column (route) generation procedure would require defining several subproblems to allow for proper pricing of routes: one for PVs and one for each possible route priority for CDs. Even though it is possible to derive an analytical method to calculate an *upper bound on the supply of vehicles* (i.e., a bound on the largest number of routes that one would like to construct for CDs based on expected costs arguments) and thus on the range of priorities to consider, this could become intractable for problems with a large pool of CDs. To circumvent this difficulty, we propose an innovative *cohesive pricing* problem that allows to consider simultaneously all CD route priorities when pricing routes. This problem is solved using extensions of standard dynamic programming (DP) labeling algorithms for elementary shortest path problems with resource constraints (ESPPRC). To speed up the solution process, we resort to a heuristic DP procedure for solving the pricing problem, whenever possible.

The branching scheme for the Branch-and-Price procedure considers in priority the total number of PVs and CDs and then flow values between customers. We branch on the most fractional variables first.

We also propose a column generation heuristic (C-Gen) to solve larger instances quickly and effectively. In this procedure, the branching process is interrupted and the restricted master problem is sent to CPLEX to find a feasible integer solution. In our implementation, the pricing problems are solved with the DP heuristic mentioned above.

Finally, we develop a Large Neighborhood Search Algorithm (LNS) heuristic. The removal and insertion operators used in this procedure are adapted or inspired by operators used in other works, such as in Pisinger & Ropke (2007): Random Removal, Random Route Removal, First Customers Removal, and Last Customers Removal; Greedy insertion, Greedy insertion with noise, PV first insertion, and CD first insertion. A distinctive feature of our LNS is the use of a Route assignment procedure, which finds the optimal assignment of the routes within a given set with the corresponding preference of CDs.

5 COMPUTATIONAL RESULTS

Extensive computational experiments were conducted for the first model. To obtain the set of instances for these experiments, we modified the well-known Solomon instances C1, R1, and RC1 with 25, 50 and 100 customers by including an additional fleet of CDs. The discrete probability function used for the availability of CDs in the instances is the binomial distribution $B(p; M)$, where p is the probability of each trial and M is the total size of the pool of CDs.

We solved all instances with the exact branch-and-price algorithm, the column generation heuristic, and the LNS heuristic. The B&P algorithm was executed for 3 hours. If we consider all instances with a gap of less than 1% solved, then all 29 instances were solved for 25 customers, 20 out of 29 for 50-customer instances, and 6 were solved optimally for instances with 100 customers. The B&P performs well for smaller instances with 25 customers, however, as the number of delivery requests increases to 100, the algorithm fails to converge to a solution in under 3 hours for most instances.

While B&P takes hours to execute, the column generation heuristic takes minutes and finds good feasible solutions for all instances. It performs especially well for clustered instances where the gap is only 1.18% for large instances with 100 customers.

We observed that LNS is much faster than C-Gen and that it provides better average solution values for the larger 100-customer instances. On average, LNS terminates within a minute while can C-Gen take up to 41 minutes on average for the R1 instances. However, the smaller 25-customer instances are solved in about the same time with C-Gen and the gaps are better. In practice, platforms can have hundreds or thousands of delivery requests that need to be fulfilled, a method that can solve large instances quickly is thus required.

We also ran LNS on larger instances with 200 customers and a pool of 1,000 CDs. This allowed us to perform detailed sensitivity analyses of the parameters used for computing the compensation of CDs and to derive managerial insights regarding the optimal level of CDs' compensation.

Computational experiments were also performed for the second model (with sectors). In that case, we focused on the Solomon instances C1 with 25, 50 and 100 customers. The C1 set of instances with 100 customers has 10 clusters that we use to represent the sectors in a city $u \in U$, i.e., $|U| = 10$ when we consider all 100 customers. The discrete probability function used for of ξ_u in all instances is the binomial distribution $B_u(p_u; M_u)$ for each sector$u \in U$, where p_u is the probability of success of each trial and M_u is the size of the pool of ODs in sector u. Variants of the base instances focusing on sectors further or nearer to the depot were also considered.

The B&P exact algorithm and three versions of the column generation heuristic (stopping after exploring 1, 10 or 100 nodes in the branch-and-bound tree) were applied to these instances. C-Gen-100, i.e., the version that stops after exploring 100 nodes of the tree, provides the best balance between solution quality and running times.

References

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

AmazonFlex. 2021. AmazonFlex.

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

Gdowska, K., Viana, A., & Pedroso, J.P. 2018. Stochastic last-mile delivery with crowdshipping. *Transportation Research Procedia*, **30**, 90–100.

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