A Branch-Price-And-Cut Algorithm for Stochastic Crowd Shipping Last-Mile Delivery with Correlated Marginals

Marco Silva[1] ✉
INESC TEC, Porto, Portugal

João Pedro Pedroso ✉
INESC TEC, Porto, Portugal
University of Porto, Portugal

Ana Viana ✉
INESC TEC, Porto, Portugal
Politechnic of Porto, Portugal

Xenia Klimentova ✉
INESC TEC, Porto, Portugal

Abstract

We study last-mile delivery with the option of crowd shipping, where a company makes use of occasional drivers to complement its vehicle's fleet in the activity of delivering products to its customers. We model it as a data-driven distributionally robust optimization approach to the capacitated vehicle routing problem. We assume the marginals of the defined uncertainty vector are known, but the joint distribution is difficult to estimate. The presence of customers and available occasional drivers can be random. We adopt a strategic planning perspective, where an optimal a priori solution is calculated before the uncertainty is revealed. Therefore, without the need for online resolution performance, we can experiment with exact solutions. Solving the problem defined above is challenging: not only the first-stage problem is already NP-Hard, but also the uncertainty and potentially the second-stage decisions are binary of high dimension, leading to non-convex optimization formulations that are complex to solve. We propose a branch-price-and-cut algorithm taking into consideration measures that exploit the intrinsic characteristics of our problem and reduce the complexity to solve it.

2012 ACM Subject Classification Applied computing → Transportation; Mathematics of computing → Mathematical optimization

Keywords and phrases Last-mile delivery, Stochastic Vehicle Routing Problem, Crowd shipping, Distributionally Robust Optimization, Data-driven Optimization

Digital Object Identifier 10.4230/OASIcs.ATMOS.2021.12

Funding The authors are funded by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness) and by National Funds through the Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project "POCI-01-0145-FEDER-028611".

1 Introduction

Last-mile delivery is defined as the movement of goods from a transportation depot to the final delivery destination, which is typically a personal residence. Due to its importance and competitive value, last-mile delivery has prompted many companies to seek creative and innovative solutions.

[1] corresponding author

In this paper, we consider a setting in which a company not only has a fleet of capacitated vehicles and drivers available to make deliveries, but may also use the services of occasional drivers (ODs) who are willing to make deliveries using their vehicle in return for a small compensation. Under such business model, a.k.a crowd shipping, the company seeks to make all the deliveries at the minimum total cost, i.e., the cost associated with their vehicles and drivers plus the compensation paid to the ODs.

The advantages of crowd-shipping are numerous and are not only related to economic issues, since the compensation for the ODs is generally less than the cost associated with delivering using its own capacitated vehicles. If relying on the idea of individuals sharing their potentially under-utilized property, sharing vehicles can lead to a reduction in polluting emissions, energy consumption, noise and traffic congestion.

The application of crowd shipping alluded to above gives rise to new and interesting variants of the routing problem. It has been addressed as an extension of the classical vehicle routing problem (VRP) or the traveling salesman problem, being modeled under different deterministic, stochastic and/or dynamic optimization approaches.

In this work we adopt a data-driven stochastic approach where we model uncertainty as the probability of each customer to be delivered by an OD, a.k.a outsourced, or to be absent. We name them skipped customers. This probability, modeled as a Bernoulli distribution, should be easy to compute from historical data. Different from other crowd shipping last-mile delivery works in the literature, we do not assume that the uncertain events are independent ([8, 10, 13, 22]). Furthermore, because estimating correlations from potentially high dimensional uncertainty historical data can be very difficult (as is our case with many customers), we propose a worst-case probability approach where the joint probability of customers uncertainty is not known. We are interested in analyzing the effect of this assumption in the results when compared to the independent uncertainty assumption.

We consider a two-stage model with recourse. In the first stage, only the ordering in which the customers will be visited is defined. The company's vehicle routes are set only in the second stage after the uncertainty is revealed. Furthermore, we assume that each company's vehicle can serve a limited number of customers. A route is defined by starting at the depot, then following the order defined in the first stage, but skipping outsourced or absent customers and returning to the depot if the maximum number of customers have been delivered or if there are no more customers to be delivered. A new route is started from the depot going to the next not outsourced customer and following the same scheme as in the previous route. Potentially, many vehicle routes are set.

We are also interested in analyzing the potential cost savings associated with this recourse when compared to the case of reoptimization, when a different optimal decision is made for each scenario of uncertainty.

The main contributions and results of this work are:

- A novel data-driven worst-case probability paradigm for crowd shipping last-mile delivery, advancing the state-of-the-art in this topic. We model uncertainty in a way that it can capture customers that are absent or outsourced to ODs.
- A mixed-integer linear optimization formulation based on a distributionally robust formulation solved with a branch-price-and-cut algorithm approach, where we can capture characteristics of the problem to reduce the complexity to solve it.
- Computational evidence on the capability of the proposed model, that reflects a more realistic assumption of correlated marginals, to obtain solutions that can improve those provided using more simplified assumptions.

In what follows, in Section 2 we review relevant approaches to solve variants of the problem and contextualize our approach. In Section 3 we elaborate on the literature on distributionally robust optimization that we leverage to formulate our problem. In Section 4 we formally present our problem, the model and the formulation we have defined, exploiting the problem's characteristics to reduce the complexity of the algorithms proposed to solve it. Section 5 details the algorithm developed to be able to solve larger instances. Next, in Section 6 we present and discuss the computational results. Finally, in Section 7 we present the conclusion of the work done.

2 Literature review

Here we focus on the literature most relevant to compare to our approach. We are interested not only in crowd shipped last-mile related publications, but also in works that deal with similar problems under the concept of customer uncertainty.

2.1 Crowd shipping routing

A seminal work on last-mile delivery with crowd shipping is proposed in [4]. The authors study a deterministic approach where the customers' locations and the ODs parameters are input data. The model proposed is a combination of an assignment problem, where ODs are assigned to customers based on pre-defined assignment rules, with a capacitated VRP where routes are defined for vehicles passing through customers not served by ODs. For each OD and customer combination, a compensation fee to be paid for the outsourcing is also defined. Furthermore, each OD always accepts deliveries assigned to her/him. Under these assumptions, a customer is only outsourced to an OD if the overall solution is optimal. The pricing mechanism, meaning how compensation fees are defined, undertakes a critical part of the algorithm and is discussed in more detail by the authors. The authors develop a multi-start heuristic to handle instances with more than 25 customers.

Differently, in [5] the authors develop a dynamic solution alternative, where the solution is adjusted every time new information is available. They consider a service platform that automatically creates matches between parcel delivery tasks and ODs. The matching of tasks, drivers, and dedicated vehicles in real-time gives rise to a new variant of the dynamic pickup and delivery problem. They propose a rolling horizon framework and develop an exact solution approach to solve the matching problem each time new information becomes available.

The authors in [10] introduce a dynamic and stochastic routing problem in which the demand, arrives over time, as also does part of the delivery capacity, in the form of in-store customers willing to make deliveries. They develop two rolling horizon dispatching approaches to the problem: one that considers only the state of the system when making decisions, and one that also incorporates probabilistic information about future online orders and in-store customer arrivals.

In [9], the authors consider stochastic ODs and define routes for the company vehicles and the ODs based on their destination. They consider time windows when the ODs may appear and use a two-stage model in which partial routes of the company vehicles are defined in the first stage and, after the ODs are revealed, they adjust deliveries in the second stage. A penalty is paid for non served customers. They develop a Mixed Integer Linear formulation for the problem and special techniques to expedite the resolution. The stochastic solution is based on a scenario approach and they assume a uniform distribution of scenarios. Results are reported on instances with up to 20 customers and 3 ODs.

In [13] the authors consider that customers can be offered or not to potential ODs and that there is a known probability of them being accepted. They develop a heuristic to identify which customers will be offered to ODs and what will be the exact expected value of the associated solution by scenario enumeration. The probabilities of acceptance are considered independent. Computational experiments are conducted on randomly generated instances of 15 customers.

2.2 Routing with customer uncertainty

One of the first works addressing routing with customer uncertainty was presented in [17] that defines a problem of routing through a set of customers where only a random subset of them needs to be visited: The Probabilistic Traveling Salesman Problem. Assuming that the probability distribution is known and that it is equal to all customers and independent, the authors derive closed-form expressions for computing efficiently the expected length of any given tour.

In [6] the authors extend the previous work by considering a probabilistic variant of the classical VRP, in which demands and/or customer presence are stochastic. They introduce a recourse strategy where, in the second stage, not only absent customers are skipped, but also the route is broken and a detour happens every time the capacity of the vehicle is reached. Another contribution of the work is to elaborate on the need that many times arises of looking for strategic planning solutions, where an a priori sequence among all customers of minimal expected length is calculated, rather than solving the problem only when the demand becomes known. Assuming that the probability distribution is known, different to each customer and independent, they find closed-form expressions and algorithms to compute the expected length of an a priori sequence.

To solve the two previous models, integer L-Shaped branch-and-cut algorithms were proposed in [19] and in [14]. The authors could solve instances with up to 9 uncertain customers.

A specialized branch-and-bound algorithm is presented in [2] for the probabilistic traveling salesman problem under the a priori strategy. They adapt existing algorithms for the deterministic traveling salesman problem using the closed expected value evaluation expression defined in [17] and present numerical results for instances of up to 18 customers. The same authors present in [3] another branch-and-bound approach, this time using parallelization techniques, solving instances of up to 30 customers.

An approximation algorithm is presented in [18], for the VRP with probabilistic customers. They propose a two-stage stochastic optimization set-partitioning formulation where, in the first stage, a dispatcher determines a set of vehicle routes serving all potential customer locations, before actual requests for service realize. In the second stage, vehicles are dispatched after observing the subset of customers requiring service; a customer not requiring service is skipped from its planned route at execution. A column generation framework that allows for solving the problem to a given optimality tolerance is proposed. For a time limit of six hours, instances of up to 40 customers were solved.

The works presented so far assume that uncertain variables are independent. Nevertheless, in many planning problems, the correlations among individual events contain crucial information. The underlying correlations, possibly caused by some common trigger factors (e.g., weather, holidays, geographic location), are often difficult to predict or analyze, which makes the planning problem complicated. Estimating the correlations is hard, particularly when this includes the huge sample size required to characterize joint distribution since they are potentially high-dimensional. This can be our case, even when the estimation of their one-dimensional marginals is rather accurate.

Focusing on this issue from a general perspective, the authors in [1] study the possible loss incurred by ignoring these correlations, and propose a new concept called Price of Correlations (POC) to quantify that loss. They show that the POC has a small upper bound for a wide class of cost functions, including uncapacitated facility location, Steiner tree and submodular functions, suggesting that the intuitive approach of assuming independent distribution may work well for these stochastic optimization problems. On the other hand, they demonstrate that for some cost functions, POC can be particularly large.

Alternatives to the VRP with the assumption of independent uncertainty can be found in the works of [12] and [15], where the authors model using concepts from distributionally robust optimization (DRO), where it is assumed that probability distributions are not completely known and a worst-case probability distribution formulation is optimized.

3 Distributionally robust optimization (DRO)

Distributionally robust optimization is a robust formulation for stochastic programming problems and dates back to the work of [23], exploiting the concept of a worst-case probability distribution (see, e.g., [7, 11, 16]).

In this modeling approach, after defining a set $\mathcal{P}$ of feasible probability distributions that is assumed to include the true distribution $\mathbb{P}$, the objective function is reformulated with respect to the worst-case expected cost over the choice of a distribution in this set. This leads to solving the Distributionally Robust Optimization Problem

$$\min_{z \in Z} \max_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}}[h(z, \xi)], \tag{DROP}$$

where $h(z, \xi)$ is a cost function in z that depends on some vector of random parameters ξ, and $\mathbb{E}_{\mathbb{P}}$ is the expectation taken with respect to the random vector ξ given that it follows the probability distribution $\mathbb{P}$. The set $\mathcal{P}$ is called the ambiguity set.

The ambiguity set $\mathcal{P}$ is a key ingredient of any distributionally robust optimization model. It is a natural alternative when the question of how should one make decisions in the presence of a large amount of uncertain data arises and the correlations are not known. Since an ambiguity set only characterizes certain properties of the unknown true probability distribution, its estimation requires fewer data and can often be done using historical records, being suitable for data-driven approaches.

Since the introduction of distributionally robust optimization, several ambiguity sets have been proposed (e.g., [11, 21, 24]). It is shown that under specific assumptions over these ambiguity sets, many problems can be reformulated as convex optimization problems that can be efficiently solved by commercial solvers.

4 Stochastic crowd shipping last-mile delivery with correlated marginals

The two-stage approach defined in Section 1 is suitable under an a priori strategic planning process. The first stage decision will minimize the average total cost considering all scenarios under a worst-case probability paradigm. The total cost is given not only by the vehicle's routes cost but also by the total compensation fee paid to ODs.

A vital modeling decision of our approach is that uncertainty is customer-related. We can express not only the customer absence, but also uncertainty related to outsourcing the delivery service to an OD. It is different from the current crowd shipping last-mile delivery models, where uncertainty is related to the OD (e.g. [9, 10]). It is suitable for planning purposes and has the advantage that we can reduce the complexity of the problem to be solved by not having to introduce explicit OD's constraints, such as their quantity, capacity and routes, in the problem formulation. In our model, this reflects intrinsically in the customer's Bernoulli probability distribution that can be estimated from available historical data.

We define a compensation fee to be paid to the OD for each customer. In our model, it pays for only a small detour around each customer. It is equivalent to the idea that the customer will only be crowd shipped if there is an OD located very near him. It is compatible with the case where a delivery company would utilize crowd shipping with an emphasis on reducing environmental impacts, like traffic and gas emissions, and not on transforming it into an opportunity for professional services. Potential ODs are offered to outsource customers against the defined compensation fee. If they are available, and therefore accept, in the second stage the compensation fee is paid and the outsourcing is done.

A typical setting would be the use of in-store shoppers, who are willing to drop off packages for online customers on their route back home. In return, these in-store shoppers are offered a small compensation to reimburse their travel costs partially. As the participants are usually free to use any means of transportation to perform the delivery, we refer to them using ODs.

4.1 Problem formulation

Let $G = (V, A)$ be a directed graph, where $V = \{0, \ldots, N\}$ is the set of vertices and $A = \{(i, j)|i, j \in V\}$ is the set of arcs. Set V consists of a depot (vertex 0) and a subset C of customers' represented by locations ($C = \{1, \ldots, N\}$). We assume that the graph is symmetric, meaning that the cost or distance to transverse between two customers is the same regardless of the direction. Such feature is exploited in the algorithms developed to solve the problem. With each arc is associated a non-negative cost or distance c_{ij}. This cost or distance satisfies triangular inequalities. We also assume that the vehicles to be used as the company fleet are identical and can serve up to Q customers.

Vector $\xi = (\xi_1, \ldots, \xi_N)$ defines an uncertain scenario, $\xi_i = 1$ iff $i \in C$ is skipped, 0 otherwise. The support of the joint distribution, Ξ, includes all possible combinations of the scenario's components. We index scenarios using indicator $w \in W$. For each scenario with customer i being skipped there is a marginal probability, m_i, and a compensation fee, f_i, associated. As a remark, note that f_i is the compensation fee paid to the OD, weighted by the probability of the customer being outsourced. We assume that the uncertain components are not independent and the joint distribution is unknown.

We initially formulate our problem as in (DROP), where now $h(z, \xi)$ is the cost of delivery of the second stage routes and z defines the first-stage ordering.

To reduce complexity of the algorithm, we reformulate the problem exploiting some of its characteristics, as follows. We define our ambiguity set as

$$\mathcal{P} = \{\mathbb{P} \,|\, \mathbb{P}\{\xi \in \Xi\} = 1; \text{ marginals } m_i \text{ for } \xi_i = 1,\, i \in C\},$$

and since our uncertainty is binary, we reformulate DROP as in Proposition 1.

▶ **Proposition 1.** *Formulation DROP applied to our ambiguity set can be reformulated as*

$$\begin{aligned} \min_{z \in Z} \quad & s - \sum_{i \in C} m_i u_i \\ s.t. \quad & s - \sum_{i \in C} \xi_i^w u_i \geq h(z, w) \quad \forall w \in W \\ & s \geq 0,\, u_i \geq 0 \qquad\qquad\quad \forall i \in C \end{aligned} \tag{DROR}$$

where s, u_i, $i \in C$ are dual variables defined in our reformulation. We abuse notation and express the second-stage cost function now in terms of the first stage variables and the uncertainty index, $h(z, w)$.

Proof. We defer a step by step reformulation to Appendix A. ◀

The next step is to define the first and second stage formulations, including the cost function $h(z, \xi)$. The first stage is defined solely by a ordering for serving the customers. The following variables are used:

- First-stage main variable
  $z_{i,j} = 1$ iff customer i is served before customer j.
- First-stage auxiliary variables
  $z^1_{i,j,r} = 1$ iff customer r is served in between customers i and j
  $z^2_{i,j,r} = 1$ iff customer r is served before customers i and j
  $z^3_{i,j,r} = 1$ iff customer r is served after customers i and j

The second stage is defined in a way that we can calculate the cost of a route given the ordering of the first stage and the scenario to be considered. We define the following sets of main and auxiliary second-stage variables, where now we include the depot in the ordering as it will be always the first and last to be served in each route:

- Main variables
  $y_{w,i,j} = 1$ iff, for scenario ξ^w, depot or customer j is served right after depot or customer i. This means that all customers r in between i and j are outsourced in this scenario.
  $v_{w,i,j} = 1$ iff, for scenario ξ^w, vehicle capacity, Q, is reached at customer i and j is the next not skipped customer. This means that before customer i, in scenario ξ^w, there are $kQ - 1$ customers, where $k \in \{1, \dots, \left\lfloor \frac{|C|}{Q} \right\rfloor\}$ and that all customers r in between i and j are outsourced in this scenario.
- Auxiliary variables
  $y^1_{w,i,t} = 1$ iff, for scenario ξ^w and given a ordering of customers, there are t customers before i, $t \in \{0, \dots, |C| - 1\}$. It indicates the position of a customer for each scenario.

We can now define the cost function $h(z, \xi)$. The cost function sums up the cost of each arc transpassed considering all routes plus the cost of the outsourced customers. We have already stated that each variable $y_{w,i,j} = 1$ defines an arc that is transpassed and each variable $v_{w,i,j} = 1$ defines a detour to the depot. This way we define the cost function as

$$h(z, w) = \sum_{i \in C} f_i \xi_i^w + \sum_{\substack{i,j \in V \\ i \neq j}} c_{i,j} y_{w,i,j} + \sum_{\substack{i,j \in C \\ i \neq j}} (c_{i,0} + c_{0,j} - c_{i,j}) v_{w,i,j}, \tag{1}$$

where we index uncertainty with indicator w.

With all variables and cost function defined we reformulate DROR as in Proposition 2.

▶ **Proposition 2.** *With variables and cost function defined, Formulation DROR can be reformulated as*

$$
\begin{aligned}
\min \quad & s + \sum_{i \in C} m_i u_i && \text{(DROC)}\\
s.t. \quad & s + \sum_{i \in C} u_i \xi_i^w \geq \sum_{i \in C} f_i \xi_i^w + \sum_{i,j \in V} c_{i,j} y_{w,i,j} + \sum_{i,j \in C} (c_{i,0} + c_{0,j} - c_{i,j}) v_{w,i,j} \\
& z_{i,j} + z_{j,i} = 1 \\
& z_{i,j} + z_{j,r} + z_{r,i} \leq 2 \\
& z^1_{i,j,r} \geq z_{i,r} + z_{r,j} - 1 \\
& z^2_{i,j,r} \geq z_{r,i} + z_{r,j} - 1 \\
& z^3_{i,j,r} \geq z_{i,r} + z_{j,r} - 1 \\
& y_{w,i,j} \geq 1 - \xi_i^w + 1 - \xi_j^w + z_{i,j} + \sum_{r \in C} (\xi_r^w z^1_{i,j,r} + z^2_{i,j,r} + z^3_{i,j,r}) - |C| \\
& y_{w,0,i} \geq 1 - \xi_i^w + \sum_{j \in C} (\xi_j^w z_{j,i} + z_{i,j}) - |C| + 1 \\
& y_{w,i,0} \geq 1 - \xi_i^w + \sum_{j \in C} (\xi_j^w z_{i,j} + z_{j,i}) - |C| + 1 \\
& v_{w,i,j} \geq y_{w,i,j} + \sum_{k \in \{1, \dots, \lfloor \frac{|C|}{Q} \rfloor\}} y^1_{w,i,kQ-1} - 1 \\
& \sum_{t \in \{0, \dots, |C|-1\}} y^1_{w,i,t} = 1 - \xi_i^w \\
& \sum_{t \in \{0, \dots, |C|-1\}} t y^1_{w,i,t} \leq \sum_{j \in C} (1 - \xi_j^w) z_{j,i} \\
& \sum_{i \in C} y^1_{w,i,t} \leq 1 \\
& s \geq 0,\ u_i \leq 0 \\
& z^1_{i,j,r},\ z^2_{i,j,r},\ z^3_{i,j,r} \in [0,1],\ z_{i,j} \in \{0,1\} \\
& y^1_{w,i,t}, y_{w,i,j},\ y_{w,0,i},\ , y_{w,i,0},\ v_{w,i,j} \in [0,1]
\end{aligned}
$$

where the constraints and variables are valid $\forall w \in W$, $\forall i, j, r \in C$, $i \neq j \neq r$, *and* $\forall t \in \{0, \dots, |C| - 1\}$, *when not stated otherwise.*

Proof. We defer a step by step reformulation to Appendix B. ◀

5 Algorithm

Formulation (DROC) is challenging to solve. Not only it englobes an NP-Hard linear ordering problem based on binary $z_{i,j}$ variables with a weak linear relaxation, as evidence by our experiments, but also, it is defined by an exponential number of constraints and variables indexed by uncertain scenarios. To solve it we propose a branch-price-and-cut algorithm (BPC). Algorithm 1 summarizes the main steps undertaken to perform BPC. The directives of the implementation of the algorithm are:

- A customized branching rule based on the incremental ordering of the sequence of the visit of the customers. This branching rule permits that we fix many binary variables simultaneously to their lower or upper bounds at a node while producing feasible regions of equitable sizes after branching.

- A symmetry breaking strategy to limit the number of branchings. This is a way to eliminate partial orderings of customers that will not contribute to arriving at an optimal solution and therefore gain greater computational efficiency by eliminating nodes of our branching tree.
- At each node solve a relaxed restricted version of the formulation. The restricted version is composed of a finite number of scenarios.
- Initial tests indicate that the node relaxation is weak and may consume significant time. On the other hand, the independent marginal distribution version of the formulation provides a lower bound that is easy to calculate at each node. We then use this alternative as a lower bound to prune the nodes before proceeding with the calculation of the relaxed restricted version of our problem.
- Each node is solved to optimality and is pruned by its lower bound.
- Each node's integer solution is validated against new scenarios. A separation subproblem with a column and row generation approach is used to separate invalid integer solutions.
- New scenarios inserted re-initiate the process of solving the node relaxed problem.
- Valid integer solutions are tested against the incumbent solution and the correspondent node is pruned afterwards.
- Fractional solutions are branched.
- The algorithm runs until no more nodes are available to test or when a time limit is reached

Algorithm 1 Branch-price-and-cut (*BPC*) algorithm.

```
Input                                                    ▷ Q, set C, vectors c, f, m
Initialize
//Nodes list ← root node, Incumbent solution ← Heuristic, Lower bound ← −∞
while There are still nodes to be branched in the Nodes list do
    Node Select                                  ▷ Select node based on search criteria
    Initialize scenarios                           ▷ Add scenarios from parents node
    Prune                                            ▷ by Independent lower bound
    while There are still scenarios to be added do
        Solve
        Prune                                       ▷ by Node solution-lower bound
        Scenario Separation subproblem                                ▷ If integer
    end while
    Update if new Incumbent solution                       ▷ Prune if better value
    Branch node
    Prune                                                          ▷ by symmetry
    Update Nodes List
end while
Return optimal solution - order of customers to visit and expected cost
```

Appendix C details the implementation of each feature of the algorithm.

6 Experiments and Computational Results

For this Section, the objective of our experiments is two-fold: we want to analyze the effect of considering dependent marginals from a solution perspective and we are interested in analyzing the effect of the recourse strategy defined for our problem. To pursue this objective, we implement additional algorithms to compare the solution of the different approaches. All algorithms are coded in Julia ([20]) using JuMP package and Cplex 12.7 and run in an Intel Xeon Cluster. A limit of 25200 seconds (7 hours) of computing time is given for each instance.

6.1 Instances

As test instances, we adapt the ones in [22], generated from instances in the TSPLIB by truncating them to the first $n+1$ vertices (one depot and n customers) for different values of n and assigning values to m_i and f_i according to different criteria. The number of customers used is $|C| \in \{6, 10, 14, 18\}$. Five instances for each number of customers are generated.

The compensation fee f_i for each customer i is set to a fixed small value, to avoid zero compensation fees, plus a value proportional to the minimal detour considering all pairs of customers $r, j \in C, i \neq j \neq r$ and given by $\min_{j,r \in C} c_{j,i} + c_{i,r} - c_{j,r}$. We assume that the pairs (m_i, f_i) generated are coherent, meaning that the compensation paid will reflect the associated probability to skip customers.

The professional fleet vehicle capacity is defined by $Q = \left\lfloor \frac{|C|}{3} \right\rfloor$.

With the instances generated from TSPLIB, we create 4 different sets of instances based on specific probability assignment rules as described below, arriving at 80 instances. All results presented by the number of customers is an average of all of their respective instances.

Instance Set A- Probability m_i is linearly proportional to the vertex's distance from the depot, with $m_i = 0.95$ for the farthest delivery point.

Instance Set B- As in set A, but we assigned probabilities with inverse proportionality to their distance from the depot. The rationale is that, in real applications, far delivery points might be inaccessible and harder to crowdsource.

Instance Set C- Here we assume that all probabilities are equal, having $m_i = 0.3$.

Instance Set D- In this case, we select probabilities at random.

6.2 Additional algorithms

We present in Table 1 a general description of different variations of Algorithm BPC. These variants were developed to run exact solutions to similar problems found in the literature, but using the same algorithmic approach that we have established for BPC. We want to compare solutions and time performance of these different problems and algorithms.

Table 1 Algorithms variants.

Algorithm Code	Description
$INDPCAP$	Independent Marginals
$DETM$	Deterministic version
$REOPT$	Reoptimization strategy

6.3 Price of correlation

In this section, we analyze the effect of considering dependent marginals. For doing so, we run a set of instances against our algorithm but also against algorithm $INDPCAP$ that implements the same recourse but considers marginals independent. For a particular problem

Table 2 Price of Correlation.
Indep is the % savings average when compared to deterministic solution for *INDPCAP*.
Dep is the % savings average when compared to deterministic solution for *BPC*.
CG is the correlation gap average, as defined in Section 6.3.
We use the best solution provided by the algorithm under the time limit.

	Set A			Set B			Set C			Set D		
$\|C\|$	Dep	Indep	*CG*	Dep	Indep	*CG*	Dep	Indep	*CG*	Dep	Indep	*CG*
6	44.26	51.92	1.31	14.12	37.72	1.38	13.54	34.75	1.32	30.08	40.82	1.33
10	49.98	59.00	1.40	26.08	50.53	1.63	18.90	45.58	1.49	30.75	52.11	1.44
14	48.04	54.34	1.44	23.69	45.96	1.52	18.05	38.74	1.33	27.81	42.99	1.33
18	44.55	52.02	1.44	26.82	47.76	1.44	17.28	40.19	1.43	25.84	44.19	1.34

instance, let z_I be the optimal decision assuming independent marginals distribution. [1] define an indicator called correlation gap (CG) as an upper bound to the price of correlation (POC), that is given by

$$CG = \frac{\mathbb{E}_{\mathbb{P}^{D(z_I)}}[h(z_I, \xi)]}{\mathbb{E}_{\mathbb{P}^I}[h(z_I, \xi)]},$$

where $\mathbb{P}^I$ is the independent Bernoulli distribution with marginals m_i, and $\mathbb{P}^{D(z_I)}$ is the worst-case distribution for decision z_I.

We use the same indicator as a measure of the effectiveness of using a worst-case distribution formulation. A small CG indicates that the decision-maker can take the independent marginal distribution solution as an approximation of the worst-case distribution without involving much risk.

Table 2 presents, for each set of instances, the percentage of savings achieved by algorithms *INDPCAP* (Indep) and *BPC* (Dep) solutions. It also shows the correlation gap (CG) calculated for these solutions. We note that the absolute saving values of each algorithm are not as important - as that depends strongly on the compensation fees - as the relationship between them. We can see that the correlation gap (CG) indicates variations in the range of 31 % up to 74%. There is not a determinant difference between the CG indicator for different sets of instances. For many applications, this gap can be already beyond what would be acceptable as an approximation. We can observe that savings of the *Indep* solution are always larger than savings of the *Dep* solution which is coherent with the fact that the independent marginals solution is a lower bound to the correlated marginals solution. We can observe also that the savings associated with Set A are always greater than the savings for all the other sets of instances. Set A is constructed in a way that the probability of outsourcing for customers that are distant from the Depot is higher.

6.4 Quality of recourse solution

In Table 3 we compare the solution of our recourse strategy, *BPC*, to solutions provided by the algorithm that implements reoptimization strategy, *REOPT*. The two solutions are given as a percentage of savings when compared to the deterministic approach, *DETM*, and were run for small instances only to be able to calculate exact reoptimization solutions.

For the instances that were run, the gaps between *BPC* and *REOPT* solutions are very small. There is even no gap for the very small instances. We observe gaps larger than zero for the larger instance. Intuitively, we can see that for larger instances there is even more flexibility to rearrange the ordering of customers in a reoptimization strategy which can result in larger gaps.

Table 3 Quality of recourse solution.
$SolREOPT$ is the % savings average comparing to deterministic solution for $REOPT$.
$SolBPC$ is the % savings average comparing to deterministic solution for BPC.

	Set A		Set B		Set C		Set D	
$\|C\|$	$SolREOPT$	$SolBPC$	$SolREOPT$	$SolBPC$	$SolREOPT$	$SolBPC$	$SolREOPT$	$SolBPC$
6	44.26	44.26	14.12	14.12	13.54	13.54	30.08	30.08
10	49.98	49.98	26.08	26.08	19.37	18.09	30.97	30.75
14	50.89	48.04	26.72	23.69	21.25	18.58	28.66	27.81

Based on the instances run, we conclude that our recourse strategy works as a good alternative to the more flexible reoptimization strategy.

7 Conclusion

We present a novel exact solution approach for the stochastic crowd shipping last-mile delivery problem where marginals are correlated, advancing the current state-of-the-art in this topic. In our approach, it is possible to capture customers that are absent or outsourced to ODs, providing a good tool to be used for a priori strategy planning solutions. We consider a worst-case joint uncertainty distribution.

We have analyzed under what conditions this approach can be relevant using the concept of the price of correlation and show that, in many cases of the instances, studied, the defined correlation gap is higher than what would be tolerated as an approximation of the problem.

Overall, we compare the solutions of the developed algorithm BPC against different exact solution algorithms using the same branch-and-bound method (e.g., one algorithm assuming independent marginals and another with an uncapacitated one vehicle with only one route). This comparison shows that the obtained solutions improve over the others, where more simplified assumptions are considered, and can help decision-makers in their work to obtain more competitive solutions.

References

1. Shipra Agrawal, Yichuan Ding, Amin Saberi, and Yinyu Ye. Price of correlations in stochastic optimization. *Operations Research*, 60(1):150–162, 2012.
2. Mohamed Abdellahi Amar, Walid Khaznaji, and Monia Bellalouna. An exact resolution for the probabilistic traveling salesman problem under the a priori strategy. *Procedia Computer Science*, 108:1414–1423, 2017. International Conference on Computational Science, ICCS 2017, 12-14 June 2017, Zurich, Switzerland.
3. Mohamed Abdellahi Amar, Walid Khaznaji, and Monia Bellalouna. A parallel branch and bound algorithm for the probabilistic tsp. In Jaideep Vaidya and Jin Li, editors, *Algorithms and Architectures for Parallel Processing*, pages 437–448, Cham, 2018. Springer International Publishing.
4. Claudia Archetti, Martin W. P. Savelsbergh, and M. Grazia Speranza. The vehicle routing problem with occasional drivers. *European Journal of Operational Research*, 254(2):472–480, 2016.
5. Alp M. Arslan, Niels Agatz, Leo Kroon, and Rob Zuidwijk. Crowdsourced delivery—a dynamic pickup and delivery problem with ad hoc drivers. *Transportation Science*, 53(1):222–235, 2019.
6. Dimitris J. Bertsimas. A vehicle routing problem with stochastic demand. *Operations Research*, 40(3):574–585, 1992.
7. Xin Chen, Melvyn Sim, and Peng Sun. A robust optimization perspective on stochastic programming. *Operations Research*, 55(6):1058–1071, 2007.

8 Lars Dahle, Henrik Andersson, and Marielle Christiansen. The vehicle routing problem with dynamic occasional drivers. In Tolga Bektaş, Stefano Coniglio, Antonio Martinez-Sykora, and Stefan Voß, editors, *Computational Logistics*, pages 49–63, Cham, 2017. Springer International Publishing.

9 Lars Dahle, Henrik Andersson, Marielle Christiansen, and M. Grazia Speranza. The pickup and delivery problem with time windows and occasional drivers. *Computers & OR*, 109:122–133, 2019.

10 Iman Dayarian and Martin Savelsbergh. Crowdshipping and same-day delivery: Employing in-store customers to deliver online orders. *available in Optimization Online*, 2017.

11 Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

12 Thai Dinh, Ricardo Fukasawa, and James Luedtke. Exact algorithms for the chance-constrained vehicle routing problem. *Mathematical Programming*, 172(1):105–138, November 2018.

13 Katarzyna Gdowska, Ana Viana, and João Pedro Pedroso. Stochastic last-mile delivery with crowdshipping. *Transportation Research Procedia*, 30:90–100, 2018.

14 Michel Gendreau, Gilbert Laporte, and René Séguin. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, 29(2):143–155, 1995.

15 Sudipta Kumar Ghosal and W. Wiesemann. The distributionally robust chance constrained vehicle routing problem. *available at* `http://www.optimization-online.org/DB_FILE/2018/08/6759.pdf`, 2018.

16 Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4-part-1):902–917, 2010.

17 Patrick Jaillet. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, 36:929–936, December 1988.

18 Felipe Lagos, Mathias Klapp, and Alejandro Toriello. Branch-and-price for probabilistic vehicle routing. *available at* `http://www.optimization-online.org/DB_HTML/2017/12/6364.html`, December 2017.

19 Gilbert Laporte, François V. Louveaux, and Hélène Mercure. A priori optimization of the probabilistic traveling salesman problem. *Operations Research*, 42(3):543–549, 1994.

20 Miles Lubin and Iain Dunning. Computing in Operations Research using Julia. *CoRR*, abs 1312.1431, 2013.

21 Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Mathematical Programming*, 171(1):115–166, September 2018.

22 Alberto Santini, Ana Viana, Xenia Klimentova, and João Pedro Pedroso. Exact, heuristic and machine learning approaches to the probabilistic travelling salesman problem with crowdsourcing. *available at* `https://santini.in/files/papers/santini-viana-klimentova-pedroso-2020.pdf`, 2020.

23 Herbert Scarf. *A Min-Max Solution of an Inventory Problem*, pages 201–209. Stanford University Press, 1958.

24 Bart P. G. Van Parys, Paul J. Goulart, and Manfred Morari. Distributionally robust expectation inequalities for structured distributions. *Mathematical Programming*, December 2017. `doi:10.1007/s10107-017-1220-x`.

25 Christoph Weiler, Benjamin Biesinger, Bin Hu, and Günther R. Raidl. Heuristic approaches for the probabilistic traveling salesman problem. In Roberto Moreno-Díaz, Franz Pichler, and Alexis Quesada-Arencibia, editors, *Computer Aided Systems Theory – EUROCAST 2015*, pages 342–349, Cham, 2015. Springer International Publishing.

26 Bo Zeng and Long Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461, 2013.

A Proof of Proposition 1

Our objective is to reformulate the inner maximization problem of our initial formulation

$$\min_{z \in Z} \max_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}}[h(z, \xi)], \tag{2}$$

given that our ambiguity set is defined as

$$\mathcal{P} = \{\mathbb{P} \,|\, \mathbb{P}\{\xi \in \Xi\} = 1; \text{ marginals } m_i \text{ for } \xi_i = 1, \, i \in C\}, \tag{3}$$

Based on the definitions (2) and (3) we can formulate the inner maximization problem of (2) as

$$\begin{aligned} \max \quad & \mathbb{E}_{\mathbb{P}}[h(z, \xi)] & (4)\\ s.t. \quad & \mathbb{P}\{\xi \in \Xi\} = 1 & (5)\\ & \mathbb{P} \text{ has marginals } m_i \text{ for components } \xi_i = 1, \, i \in C & (6) \end{aligned}$$

We now index the uncertainties ξ with indicators w and introduce variable $p_w \geq 0$ as the probability associated with each scenario in $\mathbb{P}$. We can then reformulate our inner problem as

$$\begin{aligned} \max \quad & \sum_{w \in W} p_w h(z, w) & & & (7)\\ s.t. \quad & \sum_{w \in W} p_w = 1 & & (s) & (8)\\ & \sum_{w \in W} \xi_i^w p_w \geq m_i & \forall i \in C & (u_i) & (9) \end{aligned}$$

where we introduce dual variables s and u_i. Note that we use sign $\geq$, instead of $=$, for constraints (9). It can be done since the solution to our problem will satisfy these constraints at equality. We then restrict to non positive dual variables u_i.

Since the formulation above is always feasible (the independent marginals joint distribution will always be a possible solution to this problem), we dualize and arrive to

$$\begin{aligned} \min \quad & s + \sum_{i \in C} m_i u_i & & (10)\\ s.t. \quad & s + \sum_{i \in C} \xi_i^w u_i \geq h(z, w) & \forall w \in W & (11)\\ & s \geq 0, \, u_i \leq 0, \, \forall i \in C & & (12) \end{aligned}$$

where we can restrict $s \geq 0$ because the right side of (11) is always non negative and variables u_i are non positive.

We then merge our inner reformulation to the outer minimization problem. We arrive to the formulation of Proposition 1:

$$\begin{aligned} \min_{z \in Z} \quad & s - \sum_{i \in C} m_i u_i & & (13)\\ s.t. \quad & s - \sum_{i \in C} \xi_i^w u_i \geq h(z, w) & \forall w \in W & (14)\\ & s \geq 0, \, u_i \geq 0, \, \forall i \in C & & (15) \end{aligned}$$

We note that one can easily verify that, at optimality, the duals of constraints (14) correspond to the worst-case probability associated to each scenario, since they reflect the probability of each scenario in the original formulation.

B Development of complete reformulation in DROC

With first-stage variables defined, the constraints associated with the first stage are,

$$z_{i,j} + z_{j,i} = 1 \quad (16)$$
$$z_{i,j} + z_{j,r} + z_{r,i} \leq 2 \quad (17)$$
$$z^1_{i,j,r} \leq z_{i,r} \quad (18)$$
$$z^1_{i,j,r} \leq z_{r,j} \quad (19)$$
$$z^1_{i,j,r} \geq z_{i,r} + z_{r,j} - 1 \quad (20)$$
$$z^2_{i,j,r} \leq z_{r,i} \quad (21)$$
$$z^2_{i,j,r} \leq z_{r,j} \quad (22)$$
$$z^2_{i,j,r} \geq z_{r,i} + z_{r,j} - 1 \quad (23)$$
$$z^3_{i,j,r} \leq z_{i,r} \quad (24)$$
$$z^3_{i,j,r} \leq z_{j,r} \quad (25)$$
$$z^3_{i,j,r} \geq z_{i,r} + z_{j,r} - 1 \quad (26)$$

where all constraints are valid $\forall i, j, r \in C$, $i \neq j \neq r$.

Constraints (16) and (17) define the ordering feasible region for the first-stage binary variables $z_{i,j}$. Constraints (16) state that either customer i is served before j or the contrary. Constraints (17) are the so called 3-dicycle inequalities. They state that if customer i is served before j, and j is served before r, r cannot be served before i. Constraints (18) to (26) position customer r with relation to customer i and j. For example, constraints (18) to (20) state that customer r will only be served in between i and j if it is served after i, before j and only if these two conditions happen simultaneously. The other constraints have analogous purpose. Since we are concerned with a minimization problem, constraints (18), (19),(21),(22),(24) and (25) are redundant and can be eliminated from the final formulation. Also, variables $z^1_{i,j,r}$, $z^2_{i,j,r}$ and $z^3_{i,j,r}$ are naturally integer and integrality requirements for these can be relaxed.

The second-stage is defined in a way that we can calculate the cost of a route given the ordering of the first stage and the scenario to be considered. Due to the format of the resultant feasible second-stage region, where uncertainty parameters appear not only at the right hand side of constraints, but also as bilinear coefficients with first stage variables, we opt for equivalently defining the second stage with first-stage variables indexed by the indicator $w \in W = \{1, \ldots, |\Xi|\}$, meaning there is one variable for each possible scenario.

With the sets of main and auxiliary second-stage variables defined, we first define constraints relative to the auxiliary variables, valid $\forall i \in C$, $\forall w \in W$ and $\forall t \in \{0, \ldots, |C|-1\}$, when not stated otherwise:

$$\sum_{t \in \{0,\ldots,|C|-1\}} y^1_{w,i,t} = 1 - \xi^w_i \quad (27)$$
$$\sum_{t \in \{0,\ldots,|C|-1\}} t y^1_{w,i,t} \leq \sum_{j \in C} (1 - \xi^w_j) z_{j,i} \quad (28)$$
$$\sum_{i \in C} y^1_{w,i,t} \leq 1 \quad (29)$$

Constraints (27) state that a customer i can only be associated to one and only position t, if customer i is not outsourced in the referenced scenario. Otherwise there is no assigned position. Constraints (28) assigns of a position to each customer i based on the expression

$\sum_{j\in C}(1-\xi_j^w)z_{j,i}$, that counts the number of customers before i in the referenced scenario (note the $\leq$ sign of the constraint to accommodate the case when i is outsourced). For this reason we add constraints (29) that assure a maximum of 1 customer for each position t and guarantee together with the other constraints a natural binary solution. We can therefore relax integrality requirement for $y^1_{w,i,t}$.

We next define constraints for the variables $y_{w,i,j}$. Note that $y_{w,i,j}=1$ means that there is an arc linking customers or depot i and j in scenario ξ^w and this arc is part of a route defined in the second stage. The constraints below are valid $\forall w \in W$ and $\forall i,j,r \in C$, $i \neq j \neq r$.

$$y_{w,i,j} \leq 1-\xi_i^w \tag{30}$$
$$y_{w,i,j} \leq 1-\xi_j^w \tag{31}$$
$$y_{w,i,j} \leq z_{i,j} \tag{32}$$
$$y_{w,i,j} \leq \xi_r^w z^1_{i,j,r} + z^2_{i,j,r} + z^3_{i,j,r} \tag{33}$$
$$y_{w,i,j} \geq (1-\xi_i^w) + (1-\xi_j^w) + z_{i,j} + \sum_r (\xi_r^w z^1_{i,j,r} + z^2_{i,j,r} + z^3_{i,j,r}) - |C| \tag{34}$$
$$y_{w,0,i} \leq 1-\xi_i^w \tag{35}$$
$$y_{w,0,i} \leq \xi_j^w z_{j,i} + z_{i,j} \tag{36}$$
$$y_{w,0,i} \geq 1-\xi_i^w + \sum_j (\xi_j^w z_{j,i} + z_{i,j}) - |C| + 1 \tag{37}$$
$$y_{w,i,0} \leq 1-\xi_i^w \tag{38}$$
$$y_{w,i,0} \leq \xi_j^w z_{i,j} + z_{j,i} \tag{39}$$
$$y_{w,i,0} \geq 1-\xi_i^w + \sum_j (\xi_j^w z_{i,j} + z_{j,i}) - |C| + 1 \tag{40}$$

Constraints (30) to (34) determine the condition for an arc (i,j) to exist in a second stage, if i and j are not the depot. Variable $y_{w,i,j}=1$ only if 1) i is not outsourced (30), 2) if j is not outsourced (31), 3) if i is served before j (32) and, 4) for all other customers r, r is positioned before i and j ($z^2_{i,j,r}=1$) or after i and j ($z^3_{i,j,r}=1$) or, if positioned in between i and j ($z^1_{i,j,r}=1$), it is outsourced ($\xi_r^w=1$). This is guaranteed by constraints (33). Constraints (34) guarantee that all these conditions have to happen simultaneously. Constraints (35) to (37) and constraints (38) to (40) work in an analogous form when one of the nodes of the arc is the depot (0). Because this is a minimization problem, constraints (30), (31), (32), (33), (35), (36), (38) and (39) are redundant and can be eliminated in the final formulation. Variables $y_{w,i,j}$ are naturally binary and the integrality requirement for these variables can be relaxed.

Constraints for variable $v_{w,i,j}$ are defined below. If variable $v_{w,i,j}=1$, it means that the capacity of a vehicle is reached at customer i and, so, a detour should be performed by returning to the depot and coming back to customer j. This way, variable $v_{w,i,j}$ defines when one vehicle route reaches its ends and another vehicle route should be initiated. The constraints below are valid $\forall w \in W$ and $\forall i,j \in C$, $i \neq j$.

$$v_{w,i,j} \leq \sum_{k \in \{1,\ldots,\left\lfloor \frac{|C|}{Q} \right\rfloor\}} y^1_{w,i,kQ-1} \tag{41}$$

$$v_{w,i,j} \leq y_{w,i,j} \tag{42}$$

$$v_{w,i,j} \geq y_{w,i,j} + \sum_{k \in \{1,\ldots,\left\lfloor \frac{|C|}{Q} \right\rfloor\}} y^1_{w,i,kQ-1} - 1 \tag{43}$$

Constraints (41) guarantee that capacity is reached at customer i only if it occupies special positions in the ordering of customers relative to the scenario in reference. These positions are given by $kQ - 1$, where $k \in \{1, \ldots, \left\lfloor \frac{|C|}{Q} \right\rfloor\}$. Constraints (42) determine that the return is made to the next not outsourced customer, if it exists. Constraints (43) determine that all conditions should happen simultaneously. Again, because this is a minimization problem, constraints (41) and (42) are redundant and can be eliminated in the final formulation. Variables $v_{w,i,j}$ are naturally binary and the integrality requirement of these variables can be relaxed.

C Detailed implementation of algorithm BPC

In the next subsections we detail the implementation of each feature of the Algorithm 1.

C.1 Branching

We create a search tree with no customers pre positioned at the root node. From the root node, $|C|$ branches lead to $|C|$ nodes on the first level, each of which corresponds to a particular customer being positioned in the first position. Generally, each node at level l in a tree corresponds to a set $J_l \subseteq \{1, \ldots, |C|\}$ filling the first l positions in a given order. By successively placing each customer j ($j \in C \backslash J_l$) in the $(|J_l| + 1)$-th position, $|C \backslash J_l|$ new nodes are created.

A node selection is done by use of a depth-first search strategy, i.e. the node selected is the one, among unprocessed nodes with maximum depth in the search tree. This way we navigate the tree prioritizing the search of new incumbent values. The scenarios accumulated in the solution of a parent node are transmitted to all downward children of the tree.

C.2 Independent marginals lower bound

The authors in [6, Theorem 1 Strategy b] present a closed expression to, given an ordered route, calculate the a priori expected cost under the recourse strategy we have defined for our problem, when marginals are independent. It can be calculated in polynomial time. Since an independent marginal distribution provides a lower bound to our case, we can use it as a means to prune the nodes of the branch-and-bound tree. Each node of our tree defines a partial ordering of the routes to undertake. To approximate the independent marginal expected cost from below we assume that all remaining customers not sequenced in the node ordering have same costs and probability, given by the best or minimum values among them. Since we run under a depth-first search strategy, each iteration of a same branch of the tree provides a better lower bound. Also, we do not have to recalculate the lower bound from the beginning at each node, and can reuse partially the lower bound calculations of the parent's node. The last level of the tree provides an exact independent marginal expected cost for the respective route.

Let r be a route defined by an ordered sequence of visited customers, C, and let $r(i)$ represent the i-th planned visit in r (with $i = 0$, $i = N + 1$ meaning the Depot and $m_{r(0)} = m_{r(N+1)} = 0$, $c_{r(0),r(N+1)} = 0$). For completeness, and adapting to our case with compensation fees to be payed to ODs, the expression for the a priori independent marginals expected cost for a given route, $E(r)$, is given by

$$\begin{aligned} E(r) = \quad & \sum_{i=1}^{N} f_i m_i + \sum_{i=0}^{N} \sum_{j=i+1}^{N+1} \left((1 - m_{r(i)})(1 - m_{r(j)}) \prod_{l=i+1}^{j-1} m_{r(l)} \right) c_{r(i),r(j)} \\ & + \sum_{i=1}^{N} \sum_{j=i+1}^{N} (c_{r(i),r(0)} + c_{r(0),r(j)} - c_{r(i),r(j)}) \gamma_{r(i)} (1 - m_{r(j)}) \prod_{l=i+1}^{j-1} m_{r(l)} \end{aligned} \tag{44}$$

where $\gamma_{r(i)} = 0$, $i \in \{1, \ldots, Q-1\}$, $\gamma_{r(i)} = (1 - m_{r(i)}) \sum_{k=1}^{\lfloor \frac{i}{Q} \rfloor} s(i-1, kQ-1)$, $i \geq Q$, and $s(b, r)$ expresses the probability of exactly r customers among the first b customers being not outsourced and is computed by recursion: For $b = 1, \ldots, N$, $r = 1, \ldots, b$, $s(b, r) = (1 - m_{r(b)}) s(b-1, r-1) + m_{r(b)} s(b-1, r)$, with initial conditions $s(b, b) = \prod_{i=1}^{b} (1 - m_{r(i)})$, $s(b, 0) = \prod_{i=1}^{b} m_{r(i)}$.

C.3 Scenario separation problem

Formulation (DROC) can be understood as a two-stage robust optimization problem with exponential number of scenarios and second stage variables. To solve it, we adopt the algorithm developed in [26] where the authors present a constraint-and-column generation algorithm to solve two-stage robust optimization problems. They argue that enumerating all the possible uncertain scenarios is not feasible, but that not all scenarios (and their corresponding variables and constraints) are necessary in defining the optimal value. Probably only a few important scenarios play the significant role in the formulation. The authors emphasize that it is different from the 2-stage stochastic optimization model where every single scenario in the scenario set actually contributes to the optimal value through its realization probability. They also show that the algorithm converges in a finite number of iterations.

Let $\hat{s}$, $\hat{u}$, $\hat{z}$, $\hat{z^1}$, $\hat{z^2}$, $\hat{z^3}$ represent the values of variables s, u, z, z^1, z^2, z^3, respectively, after solving a node restricted problem with integer solution for these variables. The separation problem is given by

$$
\begin{aligned}
\min_{\xi,y,v,y^1} \quad & \hat{s} + \sum_{i\in C}\hat{u}_i\xi_i - \sum_{i\in C} f_i\xi_i - \sum_{i,j\in V} c_{i,j}y_{i,j} - \sum_{i,j}(c_{i,0}+c_{0,j}-c_{i,j})v_{i,j}\\
& y_{i,j} \geq 1-\xi_i + 1 - \xi_j + \hat{z}_{i,j}\\
& \qquad + \sum_r (\xi_r \hat{z^1}_{i,j,r} + \hat{z^2}_{i,j,r} + \hat{z^3}_{i,j,r}) - |C|\\
& y_{i,j} \leq 1-\xi_i\\
& y_{i,j} \leq (1-\xi_j)\hat{z}_{i,j}\\
& y_{i,j} \leq \xi_r \hat{z^1}_{i,j,r} + \hat{z^2}_{i,j,r} + \hat{z^3}_{i,j,r}\\
& y_{0,i} \geq 1-\xi_i + \sum_j (\xi_j \hat{z}_{j,i} + \hat{z}_{i,j}) - |C| + 1\\
& y_{0,i} \leq 1-\xi_i\\
& y_{0,i} \leq \xi_j \hat{z}_{j,i} + \hat{z}_{i,j}\\
& y_{i,0} \geq 1-\xi_i + \sum_j (\xi_j \hat{z}_{i,j} + \hat{z}_{j,i}) - |C| + 1\\
& y_{i,0} \leq 1-\xi_i\\
& y_{i,0} \leq \xi_j \hat{z}_{i,j} + \hat{z}_{j,i}\\
& v_{i,j} \geq y_{i,j} + \sum_{k\in\{1,\dots,\lfloor \frac{|C|}{Q}\rfloor\}} y^1_{i,kQ-1} - 1\\
& v_{i,j} \leq \sum_{k\in\{1,\dots,\lfloor \frac{|C|}{Q}\rfloor\}} y^1_{i,kQ-1}\\
& v_{i,j} \leq y_{i,j}\\
& \sum_{t\in\{0,\dots,|C|-1\}} y^1_{i,t} = 1-\xi_i\\
& \sum_{t\in\{0,\dots,|C|-1\}} t y^1_{i,t} \leq \sum_{j\in C}(1-\xi_j)\hat{z}_{j,i}\\
& \sum_{i\in C} y^1_{i,t} \leq 1\\
& y_{i,j},\, y_{0,i},\, y_{i,0},\, v_{i,j} \in [0,1], \xi_i \in \{0,1\},
\end{aligned}
\tag{SEP}
$$

where the constraints below are valid $\forall i,j,r \in C$, $i \neq j \neq r$, when not stated otherwise.

If the objective value of problem (SEP) is greater than a given tolerance value we insert the respective value of the scenario solution into our restricted node formulation, together with the respective new variables y, y^1 and v and new associated constraints of problem (DROC), and restart the node solving step of the algorithm .

C.4 Symmetry breaking implementation

The recourse strategy is composed by two components. The first one is defined by skipping the absent customers. The second one is defined by adding detours when a vehicle achieves its capacity at a customer position.

For the first component, there is clearly symmetry since, for any scenario, traversing the route in one direction and skipping absent customers has the same cost as traversing the route in opposite direction. If the recourse is to be defined only by the first component, we can implement a symmetry breaking strategy by ordering the customers lexicographically and filtering all branch nodes where first and last customer in the node ordering cannot be crescent (or decrescent). Since this is valid for all scenarios, it can be used by both the independent marginals and dependent marginals cases.

For the second component there is no symmetry, since the order of traversing the route will define different nodes where the vehicle will achieve capacity and, therefore, different detour costs. There is a lexicographical order of the customers that will lead to a better solution, but identifying that order while solving each branch-and-bound node of our algorithm can be time costly. On the other hand, there are calculations that can be shared by both lexicographical orders during the execution of the algorithm. For instance, calculations for the first component of the recourse, skipping absent customers, can be made only once since this cost is the same for both lexigraphical orders. To profit from the time saving incurred by sharing these calculations we adopt the lexicographical ordering branching filter also for the recourse with the two components. We use the same node to calculate lower bounds for the two orderings, by sharing possible calculations, and consider the minimum lower bound or feasible solution as a result for this node. Note that this adds to the possibility of sharing calculations between a parent and child in the depth-first branching strategy.

C.5 Initial Incumbent solution

For an initial incumbent solution we leverage the work done on heuristics for the probabilistic traveling salesman problem. We refer to the work of [25] where the authors consider different heuristic approaches for this problem. In particular, we adapt the Almost Nearest Neighbor Heuristic ([25]) to our case. By doing this, we attempt to find a solution with a maximum lower bound. Considering independent marginals, we search for an ordering of customers where we append the customer with the lowest change of expected length from the last inserted customer to the tour. For a given set T of customers already inserted in a tour, the cost of inserting customer j can be computed as

$$\min_{j \in C \setminus T} \sum_{i=1}^{|T|} (1 - m_i)(1 - m_j) c_{i,j} \prod_{k=i+1}^{|T|} m_k,$$

We solve problem (DROC) using the heuristic solution above and use its value as our first incumbent.