CIRRELT

CIRRELT-2022-36

Two-Stage Stochastic One-to-Many Driver Matching for Ridesharing

Gabriel Homsi
Bernard Gendron
Sanjay Dominik Jena

November 2022

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 Palasis-Prince, local 2415
Québec (Québec) G1V 0A6
Tél : 1-418-656-2073
Télécopie : 1-418-656-2624

Two-Stage Stochastic One-to-Many Driver Matching for Ridesharing

Gabriel Homsi[1,2,*], Bernard Gendron[1,2], Sanjay Dominik Jena[1,3]

1. Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)
2. Department of Computer Science and Operations Research, Université de Montréal
3. Analytics, Operations, and Information Technologies Department, School of Management, Université du Québec à Montréal

Abstract. We introduce a modeling framework for stochastic rider-driver matching in many to-one ridesharing systems, in which drivers have to be selected before the exact rider demand is known. The modeling framework allows for the use of driver booking fees and penalties for unmatched drivers, therefore supporting different system operating modes. We model this problem as a two-stage stochastic set packing problem. To tackle the intractability of the stochastic problem, we introduce three model approximations and evaluate them on a large set of benchmark instances for three different system operating modes. Our computational experiments show the superiority of some model approximations over others, and provide valuable insights on the impact of penalties and booking fees on the system's profitability and user satisfaction.

Keywords: 2-stage stochastic programming, ridesharing matching, mixed-integer programming.

Acknowledgements. We would like to thank Netlift and Mathieu Lozeau for their thoughtful input, domain knowledge, and for providing us with access to data. We are also grateful to Compute Canada for providing computational resources. The work of the first author was supported by the Chaire en transformation du transport (Université de Montréal/Polytechnique Montréal), which is funded by the Ministère de l'Économie, de l'Innovation et de l'Énergie du Québec. The work of the third author was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2017-05224 and the Fonds de Recherche Nature et Technologies (FRQNT) under grant 70739202.

---

* Corresponding author: ga.homsi@umontreal.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2023

1 Introduction

Ridesharing agencies match drivers and riders to jointly fulfill their itineraries. As more than half of the world population lives in urban areas (UN, 2018), the potential benefits of ridesharing are numerous. To name a few, the reduction of road congestion and CO2 emissions, improved access to transportation in regions with limited mass transit, and the reduction of transportation costs. Individuals may participate in ridesharing for several reasons, such as environmental awareness, economical savings, or the lack of access to mass transit systems. Furthermore, cities may encourage ridesharing through the introduction of high-occupancy vehicle lanes (Giuliano et al., 1990), and governments may implement carbon-reduction regulations to promote ridesharing (Si et al., 2022).

The practice of ridesharing is not new, and dates as early as WWII and the 1970s oil crisis. Furthermore, the expansion of telecommunication systems, particularly the Internet, contributed to the development of numerous ridesharing systems (Chan and Shaheen, 2012). The potential scale of ridesharing markets is evidenced by surveys conducted by the US government on the occupancy rate of vehicles. These surveys show that the average occupancy rate of vehicles involved in work-related trips has been smaller than 1.4 since the 1970s (McGuckin and Fucci, 2018). Thus, provided the presence of a large user-base, ridesharing systems can promptly access a large fleet of vehicles with spare capacity for transportation. The presence of such user-base is therefore of key importance for the success of a ridesharing system.

Ridesharing systems are two-sided markets: the ridesharing platform serves as an intermediary agency that proposes rideshares (matches) to drivers and riders. These drivers and riders are free to accept or reject a rideshare. Therefore, focusing on user-friendliness is a priority for ridesharing companies. Ridesharing works best when participants have similar itineraries. For example, when commuting from residential to commercial areas or vice versa. Even if ridesharing systems limit themselves to rideshares that are geographically compatible, these rideshares may not be user-friendly enough to convince the participants to commit to the rideshares. Encouraging user participation can be achieved in several ways. One way of increasing user participation is, for example, to offer incentives such as service guarantees to participants. Furthermore, applying Operations Research tools to ridesharing may improve the quality of the generated rideshares and the overall performance of the system.

Given the importance of available drivers within the system that can be matched to ride requests, it is of utmost importance that drivers remain satisfied users of the system. It is therefore in the best interest of the operator to respond as quickly as possible to driver requests. The ridesharing matching problem here studied therefore assumes that the operator has to respond to driver requests in a timely manner. As such, drivers have to be selected, typically, before all compatible rider requests are known. The operator has two possibilities to reassure selected (i.e., booked) drivers: either by paying an a priori booking fee, or by paying a compensation in case a selected driver is not matched to any rideshare. The underlying planning problem is a two-stage many-to-one dynamic ridematching problem, in which drivers can be booked on different levels in the first stage, while the rider requests are still uncertain. In the second stage, the actual rider requests are observed and assigned to booked drivers. Assignment decisions between drivers and riders correspond to planned ridesharing trips that may take place in the near future. The goal of the problem is to book drivers and assign riders to booked drivers such that the planned rideshares maximize profit.

Contributions. The contributions of this paper are as follows: 1) we introduce a novel many-to-one ride-matching problem that allows for booking drivers on different contract levels under rider request uncertainty; 2) we propose a two-stage stochastic set-packing modeling framework, instantiating three different ridesharing system operating modes; such models provide a flexible starting point for operators that can be tailored to their specific context; 3) we propose models that approximate the second-stage value function of the two-stage stochastic model, specifically, a sample average approximation model and an expected value problem model; to further improve the computational tractability, we introduce a relaxation of the sample average approximation model where the integrality constraints of the second-stage problem are relaxed; finally, 4) we evaluate all models on a new set of benchmark instances that represent a wide range of ridesharing operations; specifically, we analyze the performance of the different models' solutions in practice and quantify the economic importance to explicitly consider the uncertainty in the problem; further, we analyze the impact of booking fees, penalties and the use of multiple booking levels, and provide managerial insights, allowing operators to identify system configurations that are both profitable and result in high user satisfaction.

Outline. This paper is structured as follows. In Section 2, we conduct a literature review and distinguish our work with respect to previous research in the field of ridesharing. In Section 3, we define the considered planning problem. In Section 4, we introduce mathematical formulations to solve the problem, namely, a deterministic and a two-stage stochastic formulation. We also provide approximations for the two-stage stochastic formulation. In Section 5, we conduct computational studies where we evaluate the two-stage stochastic approximations on instances based on data from an industrial partner. In Section 6, we present the conclusions of our study and identify potential future research directions.

2 Related Work

Ridesharing planning is usually modeled either as one-to-one bipartite matching (Agatz et al., 2011) or as vehicle routing (Baldacci et al., 2004). The bipartite matching formulation allows for a simple modeling of the decision problem, and its network structure allows it to be solved in polynomial time. Different characteristics of ridesharing operations were studied before. To name a few, Baldacci et al. (2004) study a same-destination dial-a-ride problem for carpooling with multiple passengers per vehicle. Wang et al. (2018) and Peng et al. (2022) study ridesharing and taxi-sharing problems with stability constraints. The motivation for studying matching stability is that users would be less likely to arrange rideshares outside a rideshare platform that provides stable matches to their users. When considering multiple riders per vehicle, both studies employ a set-packing or set-partitioning formulation (which are known to be NP-Hard) similar to our problem, but without rider stochasticity, booking fees and penalties for not finding a match. In particular, Peng et al. (2022) proposes a branch-and-price algorithm to solve a many-to-one stable matching problem. To find feasible rideshares, the authors design the pricing subproblem as several knapsack problems, one for each driver. As the authors consider routes with at most two riders, this pricing subproblem can be solved efficiently. For surveys on ridesharing and related problems, we refer the reader to Agatz et al. (2012), Furuhata et al. (2013), Mourad et al. (2019), and Martins et al. (2021).

The issue of information flow and uncertainty concerning user requests has received mixed attention in the ridesharing literature. Several studies (e.g. Agatz et al., 2011; Wang et al., 2018) assume that information is released dynamically within a rolling horizon framework. In other words, in ridesharing contexts, information is rarely fully known in advance. However, tackling this uncertainty has often been overlooked in the literature of ridesharing and related fields. Bertsimas et al. (2019) study an online taxi dispatching problem, and propose a myopic matching policy that has no access to predictions on future demand. To the best of our knowledge, Homsi et al. (2020, 2021) are the only studies that address demand uncertainty in a dynamic context. Homsi et al. (2020) introduce a two-stage matching model with rematches for ridesharing. In their problem, rider requests are stochastic. Homsi et al. (2021) evaluate this model within a rolling horizon framework. Our work is inspired by these works in the sense that we consider stochastic rider requests. We further extend these works by allowing for multiple riders in the same rideshare. However, we do not consider the possibility of rematching in our study. Instead, our work focuses on the interplay between the ridesharing decisions and the booking fees and penalties.

Furthermore, to the extent of our knowledge, there are no ridesharing studies that consider the possibility of different service level agreements between participants and the ridesharing system. We address this gap in the literature by allowing for different levels of driver booking, which may be associated with different booking fees and penalties, and impact the profitability and feasibility of potential rideshares. Finally, previous studies have suggested penalties based on the response time, i.e., the time of occurrence of requests and the time the match is generated (e.g. Riley et al., 2019; Homsi et al., 2020), or penalties for riders that were not serviced (Lee and Savelsbergh, 2015). However, to the best of our knowledge, no attention was given to penalties associated with the failure to provide a rideshare to drivers once an agreement (i.e., a booking) between the system and the drivers is established. Given that a ridesharing system depends on the presence of a large driver user-base to attract potential riders, our study explores the use of penalties for unmatched drivers. These penalties promote a higher level of service stability to users, as the rideshare operator has a monetary risk associated with decisions that are likely to leave drivers without a rideshare. Furthermore, prioritizing drivers may be an effective strategy to bootstrap a ridesharing system, as the presence of drivers is needed in order to attract a rider user-base.

From an Operations Research perspective, the here considered problem has connections to several classical optimization problems. As drivers have to pick-up riders at their origin and drop them off at their destinations, the problem resembles an extension of Pickup and Delivery Problems defined on a graph (see, e.g. Parragh et al., 2008a), in which the vehicles have to be selected before the commodities are known. From the matching perspective, the problem structurally can also be seen as a stochastic set-packing problem (see, e.g. Escudero et al., 2011), where the objective function coefficients are uncertain. In our case, the right-hand-side coefficients of some constraints are uncertain. Our problem also has connections to facility location problems, which can be modeled as a set-partitioning problem (Baldacci et al., 2002). In this case, driver bookings correspond to the activation of facilities, and the rideshares correspond to the assignment of customers to facilities.

3 Planning Problem and Operating Modes

We now define the planning problem and the variants considered in this paper. Section 3.1 introduces the general planning problem, based on which Section 3.2 then instantiates three

problem variants corresponding to different rideshare operating modes. Finally, Section 3.3 elaborates on the relation of the problem to its multi-stage counterpart.

3.1 Problem Definition

A ridesharing platform receives transportation requests for a specific time-window (also referred to as the planning period), indicating the willingness of drivers and riders to engage in ridesharing with other users on the platform. When creating requests, a user specifies its origin, destination, earliest departure time, latest required arrival time at the destination, and whether it wants to participate as a driver or a rider. Based on this information, the platform plans ridesharing groups such that participants can fulfill their itineraries while travelling together to save fuel. As such, a ridesharing group is defined as a planned trip composed of one driver and one or more riders.

Driver and rider requests. Drivers are defined as individuals that possess a vehicle and have a planned itinerary consisting of an origin, a destination, an earliest departure time and a latest arrival time at which the driver has to arrive at her destination. Drivers may participate in a rideshare with riders, i.e., a driver may pick-up riders at their origin locations and drop them off at their specified destinations, as long as the specified time windows are respected. In most ridesharing systems that involve commuting or intra-city transportation, driver requests must be made and selected in a timely manner before the actual rider requests are known. As such, the matching planning problem considers driver and rider requests with itineraries that fall into the window of a specific planning period, e.g., the morning commuting period of a weekday. Specifically, this planning period spans the time period starting at the earliest departure time among all drivers and ending at the latest arrival time among all drivers. Once the driver requests for the planning period are known, a subset of these requests has to be *booked* in advance (e.g., the evening before) before rider requests are known, such that these booked drivers can be assigned to rider requests once they have occurred.

The occurrence of specific rider requests is uncertain. It is assumed that the platform is able to characterize this uncertainty through probability distributions or historical observations (e.g., rider requests observed throughout the last weekdays). The planning problem therefore resolves into a two-stage structure: in the first stage, driver requests are released and must be booked; in the second stage, rider requests are known and must be assigned to booked drivers. As mentioned above, such two-stage structure may represent a context where driver requests are made and selected in the evening. Rider requests are then gradually released until the morning commuting period of the planning day and assigned to the selected drivers.

Contract levels. Given that drivers have to be selected before the actual rider requests are known, it may be possible that a booked driver is not assigned to any actual rideshare. In order to provide an incentive to drivers and encourage an expanding driver user-base, the system operator may therefore book a driver according to different *contract levels*, which can be seen as a type of service agreement or booking policy: either a predefined fee is paid to a selected driver, no matter whether the driver is actually assigned to a rideshare, or a penalty is paid to drivers that have been selected, but not assigned to a rideshare. These contract levels, along with their specified booking fees and penalty values, may be specified for different service characteristics. For example, the operator may decide to pay higher booking fees for drivers that are available during larger time-windows. While the costs to book drivers may increase, larger availability windows also offer higher flexibility to match to potential riders. As a result, the final choice of such contract levels

may impact several relevant performance measures considered by the system operator, such as the profitability of the rideshare, the satisfaction of drivers (reflected by the level of compensation when not matched) and the satisfaction of riders (reflected by the ability to match them).

While drivers are typically compensated for making available their vehicles, riders typically pay a fee in order to be matched with the prospect of saving due to the rideshare. Incentives to participate in a rideshare are therefore monetary. Riders pay for the service, because it is cheaper than driving alone, or because they do not have access to a car or other forms of transportation. The drivers pick up riders in exchange for a compensation because it is also cheaper than driving alone. Therefore, both drivers and riders are potentially saving money when ridesharing. We can characterize the benefit associated with a rideshare by, for example, the amount of distance savings that it generates for its participants. Specifically, such savings may be computed as the difference between the total rideshare distance and the individual travel distances in case each participant drives alone. However, the general problem, as well as our modeling framework are general enough to allow for benefits that are based on other performance measures. Note that we do not make any assumption on how the fees are distributed among participants. For further reading on this subject, we refer the reader to literature on, for example, Shapley values for ridesharing (Levinger et al., 2020).

The objective of the problem is to book drivers and then assign them to riders to build rideshares with maximal value. In the following, we will refer this problem to the 2-Stage Stochastic Driver Matching Problem (2S-SDMP). Next, we describe how this problem can represent different operating modes of ridesharing services.

3.2 Different Operating Modes

The above introduced notions of contract levels, booking fees and penalties allow for various ridesharing system configurations. In the following, we will focus on three specific problem variants, which represent realistic operating modes in practice and will allow us to explore the impact of the system parameters on the performance measures relevant to the operator:

- **Multiple contract levels with penalties** (PEN-K): Each driver can be booked at one of k different contract levels. Booking a driver is free of cost. However, a penalty is paid if a booked driver is not assigned to a rideshare. Each contract level specifies a different driver availability window and a penalty. Assigning rider itineraries to the route of a driver results in overall savings (both in terms of used vehicles and in terms of distance), which is shared among the participants. This variant corresponds to traditional ridesharing systems, with an additional penalty mechanism to attract drivers to its user-base. For $k = 1$, this variant represents ridesharing operations where the system only offers one type of contract to its users. For $k > 1$, it represents a more flexible system that offers drivers different types of contract.

- **Multiple contract levels with booking fees** (FEE-K): Booking a driver involves a fee, but there are no additional penalties in case drivers are not assigned to rideshares. Drivers can be booked at one of k contract levels. While drivers have a specific origin (i.e., a location where they will start their trip), they do not have a fixed destination. This variant therefore represents taxi-like sharing platforms, where drivers are independent contractors that are booked in advance to transport riders. After completing the trip, they are free to take new

trips or to leave the platform. Here, benefits from rideshares mostly come from the cost savings when several riders share the same taxi, as opposed to paying a separate taxi for each ride.

- **Penalties and booking fees** (PEN-FEE): Drivers can be booked on one of two contract levels: either without a booking fee, but with a penalty if the driver is not assigned to a rideshare, or vice-versa (i.e., with a booking fee, but without penalties). This variant may fit a traditional ridesharing system, which has a smaller driver user-base and would like to attract more riders by providing more driver resources. If a driver is willing to be booked on either contract level, this provides an additional opportunity to optimize profits and improve the overall service level.

As mentioned earlier, a ridesharing operator is not only concerned with the profitability of the system, but also with its long-term success, often reflected by its attractiveness to the driver and rider user-base (reflected by, e.g., booking fees, penalties, matching probability). The first problem variant, PEN-K, will allow us to explore the trade-off between driver availability windows, rideshare profitability and penalties. Large driver availability windows allow for more matching flexibility, but their attractiveness depend on the price. The second problem variant, FEE-K, will allow us to study how the system's profitability is affected by an increasing compensation of the drivers. Finally, studying the hybrid variant PEN-FEE may help to understand under which penalty values are acceptable compared to the booking fees.

3.3 A note on the multi-stage planning problem

The above presented planning problem assumes that the driver-selection is carried out independently for a predefined planning period, for which all driver requests are known, while rider requests are yet unknown. This is a realistic setting in a variety of contexts, for example, for the planning of work-related trips, where the planning takes place twice a day: morning trips from home to work, and afternoon return trips from work to home. We now consider a more dynamic context, where requests may be revealed gradually over time and driver-selection takes place several times throughout the day. Specifically, consider a k-stage planning problem, where drivers can be selected and riders can be matched at any of the k stages, and decisions at each stage are made for the upcoming planning period (which are therefore disjoint). We now elaborate on the suitability of the 2-stage approach presented by the 2S-SDMP for specific assumptions on this k-stage problem.

Proposition 1. If driver-selection can only be made in the first stage and rider matches are decided in the following stages as rider requests occur, this multi-stage problem variant reduces to the 2S-SDMP.

While the 2S-SDMP originally assumes that all rider requests are uncertain at the time of driver-selection, it can easily account for the case of gradual rider request release. Rider requests that have not yet been observed in the first stage are all considered uncertain, no matter when in the future they will be released. Rider requests that have already occurred can still be accounted for as an uncertain rider request, with an occurrence probability of 100%. As a result, the problem structure remains unchanged, as drivers still have to be selected in the first stage, while riders are uncertain.

In contrast, a gradual release of the drivers in a multi-stage context or the possibility of selecting drivers at later stages may impact the suitability of the 2S-SDMP problem setting, depending on the assumptions made on when those drivers can be selected. In many contexts, drivers will have to be selected at specific times, h hours before, for example, the beginning of the planning period. In the case of the 2S-SDMP where the planning may span a time period from 6am to 10am, the driver-selection planning may be required to be carried out at 10pm the day before, i.e., 8 hours before the beginning of the planning period (in order to give drivers a response sufficiently early). Even in a dynamic context, this is a realistic assumption: while driver and rider requests arrive gradually over the day, the planning is performed several times per day, h hours (e.g., 4 hours) before the beginning of the actual planning period.

Proposition 2. If drivers with itineraries within a given time-window have to be selected a specific time before the start of that time-window, and if each rider request appears in only one decision stage, then this multi-stage problem variant can be solved exactly by multiple executions of the 2S-SDMP.

This proposition follows from the fact that the planning over the entire planning horizon separates into independent planning problems, one for each planning period given by the various stages. Each of those planning problems then corresponds to an independent instance of the 2S-SDMP. Note that the assumption on the rider requests is quite realistic, given that it is unlikely that the combination of earliest departure time and latest arrival time allows for feasible driver matches in two consecutive planning periods. Finally, also note that this proposition assumes that rider-request uncertainty is time-independent, i.e., there is uncertainty whether a specific request occurs (such as in the case of morning/afternoon commuting), but no uncertainty regarding the time-window of the rider. This also appears to be a realistic assumption in several contexts, such as work-related trips, which typically occur around the same time.

If, however, a driver can be booked at any decision stage before the start of the time-window of her itinerary, or rider request uncertainty is time-dependent (i.e., there is additional uncertainty on the time for which the ride is requested), theoretically, a multi-stage problem is required. The 2-stage stochastic problem can be extended to approximate this multi-stage problem by using multiple time-periods in the second stage, in which drivers can be selected at a later moment in time and riders can be assigned to the selected drivers. An example of such a model is given by Homsi et al. (2020), which additionally allows for rematching participants in later time periods.

4 Mathematical Models

In this section, we formally describe the problem that we are studying through mathematical programming models that are general enough to encompass all problem variants discussed above (see Section 3.2). In Section 4.1, we first introduce a deterministic model, which assumes that the actual rider requests are known in advance. While this is unrealistic in practice (unless requests can be accurately predicted), the performance of the deterministic model operating under perfect information can be used as a benchmark. We then define a two-stage stochastic model in Section 4.2, explicitly addressing rider request uncertainty. Finally, we discuss several approximations of the two-stage model in Section 4.3.

4.1 The deterministic formulation

Let D be the set of driver requests and R be the set of rider requests, which jointly form the set of transportation requests with $D \cap R = \emptyset$. Let L be the set of contract levels on which a driver can be booked. In order to be available to participate in a rideshare, a driver $i \in D$ has to be booked at some contract level $\ell \in L$, implying a booking cost (e.g., a booking fee) of f_i^ℓ. A rideshare is a route that starts at the driver's origin location, then picks up and delivers one or more riders to their destination, and ends at the driver's destination.

Each booked driver and each rider can participate in at most one rideshare. Let $\mathbb{P}(R)$ be the power set of riders R, i.e., a set containing all subsets of R. For a given driver $i \in D$ booked at contract level $\ell \in L$, let $\Omega_i^\ell \subseteq \mathbb{P}(R)$ represent the set of feasible rider subsets that may share a ride with driver i. Any rider subset $\omega \in \Omega_i^\ell$ therefore represents a possible rideshare, and we here assume that the exact route for this rideshare is the most economical one, generating a total system revenue $r_{i\omega}^\ell$. The set of feasible rideshares Ω_i^ℓ, along with the corresponding revenues, can be enumerated beforehand. If a booked driver is not assigned to any rideshare, a penalty p_i^ℓ is be paid for compensation.

Let z_i^ℓ be a binary variable that takes value 1, if and only if driver $i \in D$ is booked at level $\ell \in L$. For each driver $i \in D$, level $\ell \in L$ and riders $\omega \in \Omega_i^\ell$, let $y_{i\omega}^\ell$ be a binary variable that takes value 1, if and only if the riders in ω share a ride with driver $i \in D$, booked at level $\ell \in L$.

Further, for each rider $j \in R$, let b'_j be a binary constant equal to 1, if and only if rider request j occurs, i.e., rider j is available for a rideshare. For a given driver $i \in D$, contract level $\ell \in L$ and rider subset $\omega \in \Omega_i^\ell$, let $a_{j\omega}$ denote a binary constant equal to 1, if and only if $j \in \omega$. A deterministic matching model can be written as:

$$\max \sum_{i \in D} \sum_{\ell \in L} -f_i^\ell z_i^\ell + \sum_{i \in D} \sum_{\ell \in L} \Big(\sum_{\omega \in \Omega_i^\ell} r_{i\omega}^\ell y_{i\omega}^\ell - p_i^\ell (z_i^\ell - \sum_{\omega \in \Omega_i^\ell} y_{i\omega}^\ell) \Big) \tag{1}$$

$$\sum_{\ell \in L} z_i^\ell \leq 1 \qquad \forall i \in D \tag{2}$$

$$\sum_{\omega \in \Omega_i^\ell} y_{i\omega}^\ell \leq z_i^\ell \qquad \forall i \in D, \ell \in L \tag{3}$$

$$\sum_{i \in D} \sum_{\ell \in L} \sum_{\omega \in \Omega_i^\ell} a_{j\omega} y_{i\omega}^\ell \leq b'_j \qquad \forall j \in R \tag{4}$$

$$z_i^\ell \in \{0, 1\} \qquad \forall i \in D, \ell \in L \tag{5}$$

$$y_{i\omega}^\ell \in \{0, 1\} \qquad \forall i \in D, \ell \in L, \omega \in \Omega_i^\ell. \tag{6}$$

Objective (1) maximizes the total system profit, given by the difference between the revenue generated by the rideshares, the fees for booked drivers and the penalties for drivers that have been booked, but have not been assigned to any rideshare. Constraints (2) ensure that drivers cannot be booked on more than one level. Constraints (3) ensure that a driver can serve at most one rideshare, and only if it is booked. Finally, Constraints (4) ensure that a released rider can rideshare with at most one driver. This model has the structure of a set-packing model with the addition of driver booking variables z_i^ℓ. As such, the problem is NP-hard.

Note that this problem formulation makes no assumption on how route feasibility is defined, but is flexible enough to accommodate many feasibility requirements while enumerating Ω_i^ℓ. The definition of feasibility may depend on the ridesharing system and the objectives of the operator. For the sake of our numerical analyses, we will focus on two intuitive feasibility criteria: vehicle capacity and time-window feasibility. The latter is even more important if the planning period spans a long time (e.g., several hours). We give further details on route feasibility when explaining our route enumeration procedure in Section 5.1.4.

While the formulation above is based on set-packing, the problem can theoretically also be modeled as a compact arc-flow formulation (see, e.g., Baldacci et al., 2004). Such arc-flow formulation would require to explicitly model all driver routes, including their feasibility requirements, on an (almost) complete graph. As such, the problem can be seen a stochastic extension of the Pickup and Delivery Problem with Time Windows (see Parragh et al. (2008a) and Parragh et al. (2008b) for extensive surveys), but with an excessively large number of vehicles. This problem generalizes the vehicle routing problem, which is known to be NP-hard (Lenstra and Kan, 1981). The corresponding formulations would therefore likely be intractable for general-purpose MIP solvers and the explicit representation of route constraints (e.g. time window constraints) may lead to a formulation that provides worse bounds than a set-packing formulation (see, e.g., Costa et al., 2019). Next to the issue of solving such formulations, they also limit the possible definitions for rideshare feasibility and revenue functions. While a set-packing formulation allows for the application of complex business rules and feasibility criteria during the route enumeration process, an arc-flow formulation is limited to linear additive rules.

4.2 The two-stage stochastic formulation

We now no longer assume that we have an a priori knowledge on whether a rider is released or not. Instead, we assume that we can characterize the uncertainty on the release of riders through probability distributions or historical observations. Thus, instead of having binary constants b'_j for the release of rider $j \in R$ as in the deterministic model, we represent this uncertainty through random binary variables $\tilde{b}_j, \forall j \in R$.

We are concerned with finding a set of booked drivers that allows for maximum profit as averaged over all (or most) realizations of $\tilde{\mathbf{b}}$. To this end, we formulate a two-stage stochastic problem that maximizes an expected second-stage value function. In the first stage, the model decides which drivers to book. Based on these decisions, one second-stage problem is solved for each realization of $\tilde{\mathbf{b}}$, where booked drivers and released riders are assigned to rideshares. The two-stage stochastic model can be written as follows:

$$\max \sum_{i \in D} \sum_{\ell \in L} -f_i^\ell z_i^\ell + \mathscr{Q}(\mathbf{z}) \tag{7}$$

$$\sum_{\ell \in L} z_i^\ell \leq 1 \qquad \forall i \in D \tag{8}$$

$$z_i^\ell \in \{0, 1\} \qquad \forall i \in D, \ell \in L, \tag{9}$$

where

$$\mathscr{Q}(\mathbf{z}) = \mathbb{E}_{\tilde{\mathbf{b}}}[Q(\mathbf{z}, \mathbf{b})]$$

is the expected second-stage value function. Objective (7) maximizes the expected profit of building rideshares in the second stage, minus the costs associated with driver booking in the first

stage. For a given realization (e.g. a scenario) $\mathbf{b}$ of $\tilde{\mathbf{b}}$, the second-stage value function $Q(\mathbf{z}, \mathbf{b})$ is defined as:

$$Q(\mathbf{z}, \mathbf{b}) := \max \sum_{i \in D} \sum_{\ell \in L} \Big(\sum_{\omega \in \Omega_i^\ell} r_{i\omega}^\ell y_{i\omega}^\ell - p_i^\ell (z_i^\ell - \sum_{\omega \in \Omega_i^\ell} y_{i\omega}^\ell) \Big) \tag{10}$$

$$\sum_{\omega \in \Omega_i^\ell} y_{i\omega}^\ell \leq z_i^\ell \qquad \forall i \in D, \ell \in L \tag{11}$$

$$\sum_{i \in D} \sum_{\ell \in L} \sum_{\omega \in \Omega_i^\ell} a_{j\omega} y_{i\omega}^\ell \leq b_j \qquad \forall j \in R \tag{12}$$

$$y_{i\omega}^\ell \in \{0, 1\} \qquad \forall i \in D, \ell \in L, \omega \in \Omega_i^\ell. \tag{13}$$

Objective (10) maximizes the profit of the selected rideshares, minus the costs associated with booked drivers that were not assigned to any rideshare. The second-stage problem is a set-packing problem and, as such, is NP-hard itself. Solving the two-stage model for all realizations of $\tilde{\mathbf{b}}$ would require to solve a formulation with an exponential number of variables, which would be computationally intractable for most probability distributions. For that reason, we now introduce three formulations that approximate the second-stage value function.

4.3 Approximations for the two-stage stochastic model

The two-stage problem may be challenging to solve, as the number of all possible rideshares and the number of all possible realizations for $\tilde{\mathbf{b}}$ are both exponential on the number of riders, which can lead to a very large model. To address this, we propose three approximations for the second-stage value function.

4.3.1 The sample average approximation problem

We first introduce a sample average approximation (SAA) for the two-stage stochastic formulation (Birge and Louveaux, 2011). To this end, we generate a set of scenarios $S = \{b^1, \ldots, b^{|S|}\}$ randomly sampled from $\tilde{\mathbf{b}}$, where $|S| \ll 2^{|R|}$. Alternatively, S can be obtained from historical observations. These scenarios are then used to approximate the second-stage value function $\mathscr{Q}(\mathbf{z})$ by

$$\mathscr{Q}_{\text{SAA}}(\mathbf{z}) = \sum_{s=1}^{|S|} \frac{1}{|S|} Q(\mathbf{z}, \mathbf{b}^s).$$

The corresponding model is solved under the same set of Constraints (8) and (9). Note that, if only a single scenario is used to build the SAA (i.e., $|S| = 1$), then the model is structurally identical to the deterministic model (1)–(6).

4.3.2 The sample average approximation problem with relaxed second stage

If the integrality gap of the previously defined SAA model is small, then relaxations of the second-stage function may still generate high-quality integer solutions. The relaxation of integrality constraints in stochastic programming has been applied to other problems. For example, Ahmed et al. (2003) study a multi-stage stochastic programming model for capacity expansion under

uncertainty. The authors relax the integrality constraints and then use an heuristic to transform the continuous relaxation solution into an integer solution.

We here pursuit a similar approach. We relax the integrality constraints (13) of the second-stage problems (i.e., $y^{\ell}_{i\omega} \in [0,1], \forall i \in D, \ell \in L, \omega \in \Omega^{\ell}_i$). First-stage decisions, however, remain binary. We refer to this relaxation as the SAA-R. Such a relaxation is likely to dramatically reduce the computing time required to solve the model, while still maintaining first-stage binary decisions. Such formulation would also allow for the use of a Benders decomposition algorithm to solve the model, if needed. In Section 5.2.2, we empirically evaluate if the second-stage value function approximation of the SAA-R is sufficiently strong to replace the SAA model, and provide insights on the performance of the SAA-R.

4.3.3 The expected value problem

When computational resources are limited and solutions are required in nearly real-time, it may be desirable to use an approximation of the second-stage value function that does not depend on multiple scenarios. Instead of considering the scenarios $S = \{b^1, \ldots, b^{|S|}\}$ explicitly in the formulation, we may use a formulation with a single sample mean realization:

$$\bar{\mathbf{b}} = \sum_{s=1}^{|S|} \frac{1}{|S|} \mathbf{b}^s.$$

The second-stage value function $\mathscr{Q}(\mathbf{z})$ is then replaced by

$$\mathscr{Q}_{\text{EVP}}(\mathbf{z}) = Q(\mathbf{z}, \bar{\mathbf{b}}).$$

This problem is commonly referred to as the expected value problem (EVP) (Birge and Louveaux, 2011). Similarly to Homsi et al. (2020), the EVP has binary variables on the left-hand-side and may have fractional coefficients on the right-hand-side of Constraints (12). This may render the formulation too restrictive or even infeasible, which therefore requires us to also relax the integrality constraints of the y variables.

5 Computational Study

In this section, we conduct extensive computational experiments on the proposed models. To provide insights on a wide range of ridesharing operating modes (see Section 3.2), we evaluate four problem variants: PEN-1, PEN-2, FEE-2 and PEN-FEE. We first introduce the set of benchmark instances used throughout our experiments in Section 5.1. We then evaluate the proposed models in Section 5.2, focusing on three types of key insights: the benefits of using one problem variant over another; an understanding on why certain models out- or underperform in specific settings; and, the impact of the input parameter values on the expected user satisfaction and system profitability.

5.1 Benchmark instances

We generate a benchmark set of 6 base instances following the generation process of Homsi et al. (2021), based on data from a Montreal ridesharing company. Then, we expand this base benchmark

set by varying the parameters of these instances to assess the impact on the platform profit and user satisfaction, while keeping the same network topology. Namely, we vary the values of the booking fees, revenues and penalties. As such, we have 39 instances for variant PEN-1, 6 instances for variant PEN-2, 6 instances for variant FEE-K, and 6 instances for variant PEN-FEE. An instance is composed of:

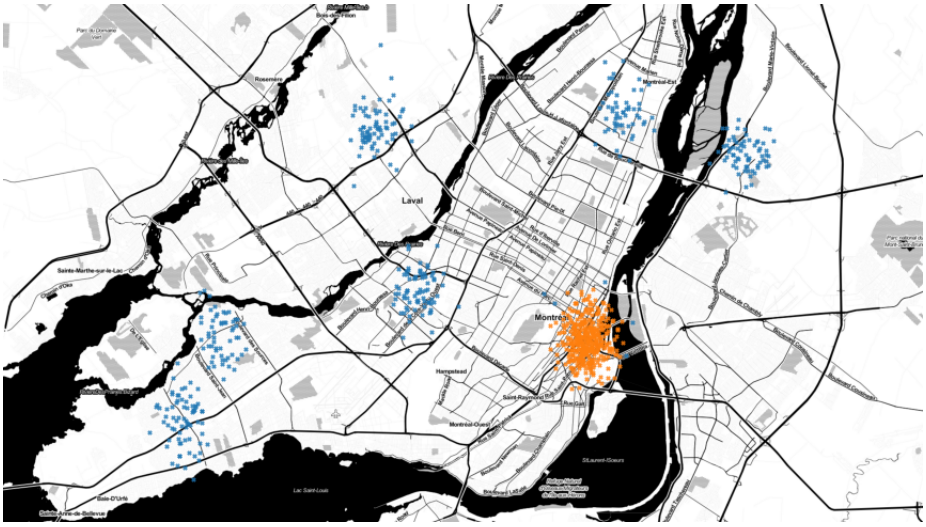
- a set D of 200 drivers, each with a specified origin, destination and latest arrival time;
- a set R of 200 riders, each with a specified origin, destination, latest arrival time, as well as the probability that its request occurs;
- a set of booking levels L, the booking fees coefficients $f_i^\ell, \forall i \in D, \ell \in L$, and the penalty coefficients $p_i^\ell, \forall i \in D, \ell \in L$;
- the availability window scaling parameters τ_{R} and $\tau_{\text{D}}^\ell, \forall \ell \in L, d \in D$;
- a set of feasible rideshares Ω_i^ℓ for each driver and booking level, that is constructed based on τ_{R} and $\tau_{\text{D}}^\ell, \forall \ell \in L, i \in D$;
- the revenues $r_{i\omega}^\ell$ associated with each rideshare $\omega \in \Omega_i^\ell, \forall \ell \in L, i \in D$;
- a ground truth set of 2000 scenarios, each containing a realization of $\tilde{\mathbf{b}}$, to validate the profit of the first-stage decisions generated by our models.

Next, we describe how these instances are generated.

5.1.1 Origin and destination locations

Inspired by Homsi et al. (2021), we randomly generate points based on seven different demand clusters around the metropolitan region of Montreal (see Figure 1). These demand clusters correspond to residential and downtown regions. Due to time-window restrictions and the requirement to generate rideshares with positive distance savings, many of the feasible rideshares satisfying these conditions have origin or destination requests within the same local region. Thus, such complex instances can be often separated into multiple instances and solved individually to provide close to optimal approximations for the larger instance. We therefore focus our study on the case where requests originate in different residential regions and have destinations in the downtown region. This represents, for example, the case of daily morning commute.

Figure 1: The geographical region considered around the metropolitan region of Montreal, along with the clusters for origins and destinations. Orange points correspond to the downtown region.



5.1.2 Probability of request occurrence

The probability that rider $j \in R$ submits a rider request for the upcoming planning (i.e., the probability that the random variable $\tilde{b}_j$ takes value 1) is a real number sampled uniformly from $[0, 1]$. We therefore sample realizations of $\tilde{\mathbf{b}}$ to generate independent request scenarios used in our models.

5.1.3 Latest arrival times and availability windows

We randomly generate the latest required arrival times for each request $i \in D \cup R$. Its latest arrival time b_i is a real number sampled uniformly from $[8, 10]$, representing the time range from 8am to 10am in the morning. The latest arrival times represent, for example, the latest time individuals may want to arrive at work. We hence assume that individuals are not willing to arrive later than the specified latest arrival time. Instead, they are willing to leave home earlier to accommodate for the delays involved in ridesharing. Thus, based on the latest arrival time, we calculate the earliest departure time a_i based on the travel time between an individual's origin $u_i \in \mathbb{R}^2$ and destination $v_i \in \mathbb{R}^2$ (in longitude and latitude), as follows:

$$a_i = b_i - \tau \cdot \frac{d(u_i, v_i)}{s}, \forall i \in D \cup R,$$

where $d(u_i, v_i)$ is the approximate routing distance in kilometers between u_i and v_i, $s = 40\ km/h$ is a constant speed assumed for all participants, and τ is an availability window scaling factor. Therefore, $d(u_i, v_i)/s$ represents the estimated individual travel time of user i if she travelled alone. When referring to a driver at level ℓ, we use the notation τ_{D}^{ℓ}. When referring to riders, we use the notation τ_{R}. When ridesharing, we assume that riders are willing to accept travel times up to 30% longer than individual travel times, i.e., their earliest departure time is computed based on $\tau_{\mathrm{R}} = 1.3$. As in Homsi et al. (2021), we estimate the true routing distance between two points

with the aid of a regression model trained with OSRM routing data (Luxen and Vetter, 2011), resulting in the following formula:

$$d(u_i, v_i) = 1.57 + 81.91 \cdot \bar{d}(u_i, v_i),$$

where $\bar{d}(u_i, v_i)$ is the Manhattan distance between u_i and v_i.

In problem variant FEE-K, drivers have no fixed destination and their time windows simply represent a time block during which they are available to provide transportation. For each level-1 driver $i \in D$, the earliest departure time a_i is set to as 45 minutes before b_i (i.e., $b_i = a_i + 45/60$). For a level-2 driver $i \in D$, a_i is set to one hour before b_i (i.e., $b_i = a_i + 1$). We uniformly sample a_i from the interval $[7, 9]$. When calculating the distance savings, we disregard the arc between the driver origin and the first pick-up. Note, however, that this distance is still relevant to determine whether a rideshare is time-feasible or not.

5.1.4 Route enumeration and route feasibility

We a priori enumerate feasible ridesharing routes with a depth-first search algorithm that incrementally builds routes. This algorithm (implemented in C++) evaluates partial routes with different sizes, and extends routes by adding new riders. The extension of a route stops as soon as the route is no longer time-window feasible (i.e., when it violates the earliest departure time and latest arrival time of participants). To speed up the enumeration, we restrict the set of possible routes to those that are user-friendly and that make sense in practice for ridesharing. Specifically, we limit each route to at most three riders, which accurately represents the capacity of most vehicles that have ridesharing-related availability window restrictions. As all destinations are in the downtown region, it is unlikely that routes with additional pick-ups after deliveries would be more advantageous than routes with all pick-ups before all deliveries. We therefore also limit the enumeration to routes that start dropping off riders to their destinations only after all riders have been picked up from their origins. Despite being an exponential procedure, many rideshares are not time-window feasible, which speeds up the route enumeration.

Table 1 shows average statistics for the enumeration procedure, grouped by problem variant. The first columns show the problem variant and total number of routes enumerated for all booking levels. The last column shows the CPU time in seconds required to enumerate these routes.

Table 1: Statistics for the route enumeration procedure by problem variant.

Variant	Routes	Routes (1 rider)	Routes (2 riders)	Routes (3 riders)	T (sec)
PEN-1	2,410	1,177	1,192	41	66
PEN-2	5,050	2,342	2,604	104	152
FEE-2	9,862	3,823	5,853	186	3,450
PEN-FEE	1,996	1,518	478	0	41

5.1.5 The revenue of a rideshare

We assume that the revenue $r^{\ell}_{i\omega}$ the system operator earns for each rideshare $\ell \in L, i \in D, \omega \in \Omega^{\ell}_i$ is computed as a share of the total distance savings $s_{i\omega}$ generated by the rideshare. A scaling parameter $\gamma^{\ell}_{\mathrm{R}}$ converts the total distance savings into a revenue value allocated to the ridesharing

operator:

$$r_{i\omega}^{\ell} = \gamma_r^{\ell} \cdot s_{i\omega}.$$

This allows us to adjust the value of a rideshare individually for each driver booking level.

5.1.6 Booking penalties

We define the booking penalties $p_i^{\ell}, \forall i \in D, \ell \in L$ as the product of a penalty scaling parameter γ_{P}^{ℓ} and the distance between driver origin and destination $d(u_i, v_i)$, as below:

$$p_i^{\ell} = \gamma_{\text{P}}^{\ell} \cdot d(u_i, v_i).$$

We can therefore represent situations where the system subsidizes a proportion of the driver's journey in case it is unable to find riders to form a rideshare.

5.1.7 Parameters used for different problem variants

We now describe the values of the parameters that are specific to each variant:

- **Single contract level with penalties** (PEN-1): We consider only one level of booking per driver (i.e., $L = \{1\}$) and no cost associated to driver booking (i.e., $f_i^1 = 0, \forall i \in D$). Studying this variant, we aim at understanding the benefits of increasing the size of the availability window, which results in a trade-off between profitability and user-friendliness. To this end, we evaluate instances with different driver availability windows by varying $\tau_{\text{D}}^1 \in \{1.3, 1.4, 1.5\}$. Furthermore, we test the following values for the revenue and penalty scaling coefficients (γ_{R}^1 and γ_{P}^1):

  1. We fix $\gamma_{\text{R}}^1 = 0.3$, and vary $\gamma_{\text{P}}^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$;
  2. We fix $\gamma_{\text{P}}^1 = 0.3$, and vary $\gamma_{\text{R}}^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

- **Multiple contract levels with penalties** (PEN-2): We consider two booking levels (i.e., $L = \{1, 2\}$) and no cost associated with driver booking (i.e., $f_i^1 = 0, \forall i \in D$). For level-1 drivers, we assume that $\tau_{\text{D}}^1 = 1.3$, $\gamma_{\text{R}}^1 = 0.2$ and $\gamma_{\text{P}}^1 = 0.3$. For level-2 drivers, we assume that $\tau_{\text{D}}^2 = 1.4$, $\gamma_{\text{R}}^2 = 0.2$ and vary $\gamma_{\text{P}}^2 \in \{0.5, 1, 2, 4, 8, 16\}$. Studying this variant, we aim at understanding the trade-off between availability window size and penalties.

- **Multiple contract levels with booking fees** (FEE-2): We consider two booking levels. All booked drivers are paid a fee in advance, and there are no penalties if they are not matched in the second stage (i.e., $\gamma_{\text{P}}^{\ell} = 0, \forall \ell \in L$). For level-1 drivers, we fix $\gamma_{\text{R}}^1 = 0.3$ and $f_i^1 = 1, \forall i \in D$. For level-2 drivers, we fix $\gamma_{\text{R}}^2 = 0.3$ and vary the booking fees $f_i^2 \in \{1.5, 2, 2.5, 3, 3.5, 4\}, \forall i \in D$. Level-1 drivers are available for 45 minutes, and level-2 drivers are available for 1 hour. We study this variant to explore which booking fees can be justified when increasing the availability window.

- **Penalties and booking fees** (PEN-FEE): The operator has the choice of selecting a driver either with a booking fee, or with a penalty as in variant PEN-2. Both levels have the same availability window $\tau_{\text{D}}^1 = \tau_{\text{D}}^2 = 1.3$ and revenue share $\gamma_{\text{R}}^1 = \gamma_{\text{R}}^2 = 0.3$. Level-1 drivers can be booked at no cost (i.e., $f_i^1 = 0, \forall i \in D$), but penalties are paid if the

booked drivers are not assigned to any rideshare on the second stage. To this end, we vary $\gamma_{\mathrm{P}}^1 \in \{ 0.5, 1, 1.5, 2, 2.5, 3 \}$. Level-2 drivers have no penalties (i.e., $\gamma_{\mathrm{P}}^2 = 0$), but a fee $f_i^2 = 1, \forall i \in D$ is paid to book them. We study this variant in order to gain an understanding of the interplay between the use of the two different driver booking types.

5.2 Computational results

In this section, we evaluate the solutions provided by the proposed models on our benchmark instances. All experiments have been executed on a machine with an AMD 3970X processor and 256 GB of memory. Models have been solved using CPLEX version 12.10. We limited CPLEX to one thread, set its execution time limit to one hour, and used Python to build all models.

5.2.1 Validation of first-stage decisions

To estimate the unbiased performance of the first-stage decisions (i.e., the values of the $\mathbf{z}$ variables) generated by each model, we evaluate these decisions on the ground truth validation set with 2,000 samples for $\tilde{\mathbf{b}}$. Let $\tilde{f}(\mathbf{z})$ be a model's optimal objective function value over these validation samples. We compare $\tilde{f}(\mathbf{z})$ against the objective function value of the *wait-and-see* problem (Madansky, 1960), which consists of solving several independent single-scenario models, one for each sample of $\tilde{\mathbf{b}}$. First-stage decisions are therefore tailored to each sample of $\tilde{\mathbf{b}}$, allowing us to obtain the best possible profit in case the planner has the ability to accurately predict the future realization of $\tilde{\mathbf{b}}$. While this is unrealistic in practice, the wait-and-see solution can be used as an upper bound to evaluate the quality of the solutions obtained from the other models. The wait-and-see objective f_{WS} is the average of the objective function values of each independent problem, and is an upper bound for $\tilde{f}(\mathbf{z})$. Thus, to estimate the quality of the first-stage solutions z we calculate the relative gap between $\tilde{f}(\mathbf{z})$ and f_{WS} as:

$$gap_{\mathrm{WS}} = \frac{f_{\mathrm{WS}} - \tilde{f}(\mathbf{z})}{f_{\mathrm{WS}}}.$$

It is interesting to note that, as the first-stage decisions of the wait-and-see problem are specific to each scenario and are aware of the actual rider requests, the wait-and-see solution will never book drivers that are not matched in the second-stage, given that this would incur unnecessary (and sub-optimal) penalties or booking fees.

5.2.2 Performance for different models on all problem variants

We now analyze the difficulty of solving the various models for each of the problem variants. Specifically, for each of the four problem variants, we evaluate the benefits of using a scenario-based approach such as the SAA over a simpler model such as the EVP. We also explore the tradeoff between representing the second-stage problem of the SAA formulation as a binary set-packing problem, or as a relaxation.

Table 2 summarizes the results averaged over the problem instances for each of the four problem variants (PEN-1, PEN-2, FEE-2, PEN-FEE) and each of the three models: the conventional SAA, the SAA with relaxed second-stage (SAA-R) and the EVP. Column "$|S|$" indicates the number of scenarios used in each model. Column "$\mathrm{profit}_{\mathrm{GT}}$" represents the average profit associated with the first-stage decisions and evaluated on the the ground truth set. Column "gap_{WS} (%)" indicates

the average percentage gap of the objective function value to the optimal objective function value of the wait-and-see solution. Column "T (sec)" represents the average CPU time (in seconds) required by CPLEX to solve the models. To assess the potential benefits of having more precise information on the occurrence of riders, we test our models with different numbers of scenarios (specifically, with 25, 50, 100 and 200 scenarios).

Table 2: Results for models on all problem variants.

model	$\|S\|$	$\text{profit}_{\text{GT}}$	gap_{WS} (%)	T (sec)	model	$\|S\|$	$\text{profit}_{\text{GT}}$	gap_{WS} (%)	T (sec)
PEN-1					PEN-2				
SAA	25	517.48	8.73	1.84	SAA	25	234.23	20.53	32.75
	50	518.69	8.47	8.85		50	253.49	14.02	56.68
	100	519.68	8.24	35.82		100	258.13	12.46	656.16
	200	519.95	8.16	163.75		200	262.48	10.99	1,711.35
SAA-R	25	517.46	8.73	0.74	SAA-R	25	237.83	19.32	7.96
	50	518.74	8.46	2.65		50	254.19	13.79	7.50
	100	519.66	8.25	9.88		100	258.20	12.43	53.67
	200	519.91	8.17	48.63		200	262.73	10.90	372.70
EVP	25	500.65	12.89	0.46	EVP	25	110.35	62.38	1.85
	50	499.60	13.07	0.84		50	103.31	64.76	1.77
	100	506.12	11.48	0.37		100	8s8	71.56	1.41
	200	506.46	11.43	0.53		200	91.74	68.67	1.77
FEE-2					PEN-FEE				
SAA	25	113.20	17.39	1,277.62	SAA	25	355.92	12.61	0.56
	50	113.19	17.39	2,973.47		50	355.98	12.59	3.75
	100	113.62	17.05	3,600.00		100	359.31	11.77	26.25
	200	113.63	17.03	3,600.00		200	360.74	11.43	378.09
SAA-R	25	112.90	17.62	574.76	SAA-R	25	355.90	12.61	0.44
	50	113.15	17.38	2,133.00		50	355.88	12.62	1.50
	100	113.59	17.06	3,411.86		100	359.19	11.80	9.95
	200	113.69	16.98	3,600.00		200	360.73	11.43	16.80
EVP	25	110.66	19.25	2,274.27	EVP	25	238.56	41.34	0.12
	50	110.88	19.01	3,600.00		50	247.95	39.04	0.67
	100	111.30	18.72	2,385.05		100	270.88	33.43	1.08
	200	110.70	19.15	1,952.12		200	270.56	33.51	0.92

The results highlight the benefits of using a stochastic model: for variant PEN-1 at 200 scenarios, the gaps of the SAA and the SAA-R are, on average, about 3 percentage points smaller than the gap of the EVP (8.16% and 8.17% versus 11.43%). Moreover, the results show that increasing the number of scenarios reduces the gap for all models, except for the EVP at problem variant PEN-2, where the average gap increases from 62.38% with 25 scenarios to 68.67% with 200 scenarios, while the average gaps of the SAA and SAA-R at 200 scenarios are 10.99% and 10.90%, respectively.

On the SAA-R performance. The results show that relaxing the second-stage problem for the SAA formulation can be an efficient strategy: for example, for variant PEN-2 with 200 scenarios, the model can be solved in about 20% of the original computing time (from 1711.35 seconds to 372.70 seconds), while the gaps are similar to those of the original SAA. We observe

similar reductions in computing time and similar gaps for all other problem variants, except for variant FEE-K where both models exceed the computing time limit, while the gaps are similar. This may suggest two further advantages associated with relaxing the second-stage problem: as the relaxed second-stage problems are easier to solve, we may be able to better approximate the second-stage value function by increasing the number of samples. Further, in the case where the original SAA formulation becomes too hard to solve, relaxing the second-stage allows us to apply Benders decomposition (which requires continuous subproblems).

To better understand the different levels of computational effort required to solve the SAA and the SAA-R, we conduct further experiments where we compare the optimal solution (or best known solution) value of each model with the solution value of its linear programming relaxation. Average results for the SAA and SAA-R with 200 scenarios are shown in Table 3. Column "profit_{200}" indicates the profit (objective function value) of the models over the 200 scenarios used to find the first-stage decisions. Column "int. gap (%)" shows the integrality gap between the optimal (or best known) solution and the linear programming relaxation. Column "opt. gap (%)" indicates the optimality gap as reported by CPLEX.

Table 3: Integrality and optimality gaps for the SAA and the SAA-R with 200 scenarios.

variant	model	profit_{200}	int. gap (%)	opt. gap (%)	T (sec)
PEN-1	SAA	521.05	0.15	0.00	186.10
	SAA-R	521.69	0.01	0.00	55.34
PEN-2	SAA	268.91	0.15	0.00	1,219.95
	SAA-R	269.23	0.03	0.00	238.49
FEE-2	SAA	115.70	4.17	1.24	3,600.00
	SAA-R	119.26	1.29	0.70	3,600.00
PEN-FEE	SAA	364.62	0.12	0.00	409.16
	SAA-R	365.04	0.00	0.00	18.34

As expected, the SAA-R model has smaller integrality gaps (often zero or close to zero) than the SAA model, which results in faster solution times. For cases where optimality is not proven (FEE-2), we observe that the optimality gaps of the SAA-R are still smaller than those of the SAA.

As the SAA-R is a relaxation of the SAA, the optimal value of the objective function of the SAA-R will be an upper bound to the optimal SAA objective. Column profit_{200} shows that this upper bound is relatively tight. For all variants except for FEE-2, the SAA-R objectives have a gap of at most 0.12% to the SAA objective. For variant FEE-2, none of the models has been able to prove optimality and the average gap between the best known solutions of the SAA-R and the SAA is 2.99% (115.70 and 119.26). This gap could be further improved by strengthening the relaxed second-stage set-packing formulation through valid inequalities (see, e.g. Borndörfer, 1998).

In an effort to gain an understanding of why the SAA-R provides such tight approximations for the second-stage value function, we inspect the proportion of the second-stage y variables that take value 1, as opposed to variables that take a fractional value greater than zero. Upon further inspection of a PEN-1, a PEN-2 and a FEE-2 instance solved by the SAA-R with 200 scenarios, we have observed that y variables with value 1 represent about 90% (for PEN-1), 72% (for PEN-2) and 50% (for FEE-2) of the total number of variables with non-zero values (i.e., variables with value 1, and variables with non-zero fractional values). The fact that more SAA-R variables are fractional

for variant FEE-2 explains the larger gap between the SAA-R and SAA objectives (2.99%, as opposed to 0.12% for other problem variants). We furthermore observed that the ratio between rideshares (non-zero y variables) and activated drivers is 1.04 (for PEN-1), 1.15 (for PEN-2) and 1.19 (for FEE-2). These ratios are significantly smaller than those in the EVP solutions (3.10, 3.29 and 5.00, respectively). This suggests that, even if the second-stage variables are relaxed, a significant proportion of the variables assume integer values, which leads to a better approximation of the second-stage value function, and explains the good performance of the SAA-R. Given the high solution quality induced by the SAA-R model on the ground truth and its small integrality and optimality gaps when compared to the SAA model, we restrict the following experiments to the SAA-R model using 200 scenarios.

On the EVP performance. Despite its short computing times, the EVP model does not reliably provide a reasonable approximation to the original SAA solution. This becomes evident, in particular, for variants PEN-2 and PEN-FEE, where its gaps are significantly worse than those of the other models.s Instead, it may be more effective to solve the other models with a smaller number of scenarios, which likely generates solutions of higher quality and CPU times comparable to the EVP. The bad performance of the EVP is due to the combination of fractional right-hand-side coefficients (for the mean sample realization $\bar{\mathbf{b}}$ of $\tilde{\mathbf{b}}$) and continuous second-stage variables, as $b_j \in [0,1], \forall j \in R$ are likely to be fractional, Constraints (12) will likely limit the creation of rideshares to fractional values (i.e., $y^{\ell}_{i\omega} < 1$). As a consequence, the EVP solution may partially open multiple rideshares for the same driver, such that the cost of the booking fee is offset. Indeed, upon further inspection of some instances, we have observed that drivers are partially assigned to, on average, 3.1 different rideshares for PEN-1, 3.29 for PEN-2, and 5.00 for FEE-2. The first-stage decisions are therefore not optimized under the assumption that drivers can be assigned to at most one rideshare, and consequently they may not be as profitable when evaluated on the ground truth. The performance of the EVP for variant PEN-2 is further discussed in Section 5.2.5.

5.2.3 PEN-1: impact of driver availability windows

In this and the following sections, we explore how the platform profit and user satisfaction are impacted under the different problem variants and parameter settings. With the overall objective to gain an understanding which parameter values are desirable such that both the platform and the participants benefit, we will examine several performance indicators: the platform profit, the proportion of booked drivers, the proportion of failed bookings, the proportion of satisfied riders, and the cost of the rideshare compared to driving alone.

We first focus on the impact of the driver availability window. We limit these experiments to variant PEN-1 and test availability window scaling coefficients $\tau^1_{\mathrm{D}} \in \{\, 1.3, 1.4, 1.5 \,\}$, along with different configurations of the revenue share scaling parameter γ^1_{R} and penalties scaling parameter γ^1_{P}, namely:

- we fix the penalty coefficients to $\gamma^1_{\mathrm{P}} = 0.3$, and vary the share of savings collected by the ridesharing platform $\gamma^1_{\mathrm{R}} \in \{\, 0.1, 0.3, 0.5, 0.7, 0.9 \,\}$, representing contexts where the platform passes almost all revenue to the users ($\gamma^1_{\mathrm{R}} = 0.1$) up to contexts where the platform keeps most of the revenue, and only little for its users ($\gamma^1_{\mathrm{R}} = 0.9$);
- we fix the coefficient for the share of savings collected by the ridesharing platform to $\gamma^1_{\mathrm{R}} = 0.3$, and vary the penalties $\gamma^1_{\mathrm{P}} \in \{\, 0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5 \,\}$. This represents contexts

where the platform subsidizes only 10% of a booked driver's trip in case it is not assigned to a rideshare ($\gamma_{\text{P}}^1 = 0.1$) up to contexts where drivers are highly compensated through penalties if they are not assigned to any rideshare ($1.1 \leq \gamma_{\text{P}}^1 \leq 1.5$).

We expect larger availability windows to contribute to a higher number of successful matches, and hence to more profitable solutions. However, larger availability windows may generate inconveniences for drivers. Operators are therefore interested in identifying a reasonable value for τ_{D}^1 that generates sufficient profit, while causing minimal inconveniences.

The results are summarized in Table 4. Column "book (%)" indicates the average proportion of booked driver over all drivers. Column "failed (%)" shows the average proportion of failed bookings (i.e., drivers that were booked but not assigned to any rideshare) over all booked drivers. Column "sat. riders (%)" represents the average proportion of satisfied rider requests (i.e., riders that have been assigned to a rideshare) over all released riders. Column "cost (%)" shows the percent average trip cost relative to individual trips (i.e., the ratio of shared distances plus platform cut over the sum of individual distances).

Table 4: SAA-R results for different values of τ_{D}^1 on variant PEN-1 (averaged over fixed $\gamma_{\text{P}}^1 = 0.3$ with $\gamma_{\text{R}}^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$; and fixed $\gamma_{\text{R}}^1 = 0.3$ with $\gamma_{\text{P}}^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$).

τ_{D}^1	$\text{profit}_{\text{GT}}$	gap_{WS} (%)	book (%)	failed (%)	sat. riders (%)	cost (%)	T (sec)
1.3	464.40	12.55	34.33	4.24	75.55	76.36	11.67
1.4	528.57	7.47	36.29	2.77	87.95	75.85	70.66
1.5	566.75	4.47	37.88	1.62	96.03	75.62	63.57

The results indicate that, as τ_{D}^1 increases, the platform profit also increases. This is explained by the fact that the platform is able to form more rideshares: there are more booked drivers and fewer failed bookings, as drivers can adjust more easily to different rideshares. Furthermore, a direct consequence of having less failed bookings is a smaller gap to the wait-and-see solution, which has no failed bookings. Rider demand can also be better met with larger driver availability windows: almost all riders (96.03%) are provided service when $\tau_{\text{D}}^1 = 1.5$, while the trips costs for users remain stable (around 76%, representing 24% cost savings). However, using such large availability windows would not provide the best user experience for drivers. A reasonable trade-off value for τ_{D}^1 seems to be 1.4, as it improves all performance metrics and does not require a large level of driver flexibility.

5.2.4 PEN-1: impact of revenue share and penalty levels

The goal of this section is to identify values for the revenue share and penalties such that the ridesharing platform attains a good profit while keeping users satisfied. To this end, we explore the impact of changing the platform revenue share and penalty levels independently. Then, we study the joint impact of these parameters.

Impact of revenue share. We analyze the impact of changing the scaling parameter γ_{R}^1 for the revenue share collected by the ridesharing platform (defined in Section 5.1.5). We limit these experiments to variant PEN-1. We fix the penalty coefficient to $\gamma_{\text{P}}^1 = 0.3$ and vary the share of

savings collected by the ridesharing platform $\gamma_R^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ (note that results are averaged over instances with driver availability window scaling coefficients $\tau_D^1 \in \{1.3, 1.4, 1.5\}$).

Table 5: SAA-R results for different values of γ_R^1 on variant PEN-1 (averaged over instances with $\gamma_P^1 = 0.3$ and $\tau_D^1 \in \{1.3, 1.4, 1.5\}$).

γ_R^1	profit_{GT}	gap_{WS} (%)	book (%)	failed (%)	sat. riders (%)	cost (%)	T (sec)
0.1	130.93	10.31	34.67	1.54	85.13	64.81	44.46
0.3	408.47	6.70	36.33	2.42	87.14	72.85	67.08
0.5	690.73	5.30	38.17	4.05	88.91	80.74	46.21
0.7	976.08	4.39	39.17	5.12	89.72	88.44	41.03
0.9	1,261.50	3.89	40.50	6.47	90.39	96.16	31.75

The results are summarized in Table 5. As expected, when increasing the revenue share, more matching decisions become economically profitable, which leads to more booked drivers and more matched riders, resulting in a larger profit for the platform. The platform profit also does not seem to be critically affected by the increasing proportion of failed driver bookings (which are compensated an equivalent of 30% of their individual trip costs, as $\gamma_P^1 = 0.3$). This is explained by the fact that the total paid penalties become negligible compared to the revenues generated from the rideshares. This disparity between revenue and penalty values also explains why larger revenue shares are linked to smaller gaps to the wait-and-see solution. Unfortunately, an increasing revenue share kept by the operator also results in higher costs for the users: with $\gamma_R^1 = 0.9$, the participants pay on average 96% of the costs of driving alone, which are rather unattractive cost savings. The revenue share γ_R^1 therefore has to be kept at reasonable levels (e.g., at 0.3) to ensure that users save around 30% when compared to driving alone.

Impact of different penalty levels. We now investigate how the results change according to different penalty levels γ_P^1. We limit these experiments to variant PEN-1 (for availability window scaling coefficients $\tau_D^1 \in \{1.3, 1.4, 1.5\}$), fix the share of savings collected by the ridesharing platform to $\gamma_R^1 = 0.3$, and vary the penalty scaling coefficients $\gamma_P^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$. Note that any value $\gamma_P^1 > 1$ implies that drivers are compensated more than their trip cost in case they are not assigned to any rideshare.

Table 6: SAA-R results for different values of γ_P^1 on variant PEN-1 (averaged over instances with $\gamma_R^1 = 0.3$ and $\tau_D^1 \in \{1.3, 1.4, 1.5\}$).

γ_P^1	profit_{GT}	gap_{WS} (%)	book (%)	failed (%)	sat. riders (%)	cost (%)	T (sec)
0.1	420.50	3.88	40.50	6.47	90.39	73.14	44.91
0.3	408.47	6.70	36.33	2.42	87.14	72.85	67.08
0.5	402.14	8.18	35.33	1.88	85.91	72.67	56.45
0.7	397.06	9.36	34.67	1.66	85.29	72.55	58.17
0.9	392.79	10.36	34.67	1.54	85.15	72.63	50.23
1.1	389.25	11.18	33.67	1.25	83.89	72.46	42.40
1.3	386.22	11.89	33.17	1.06	83.10	72.42	44.39
1.5	383.23	12.59	33.17	1.06	83.10	72.42	56.54

The results are summarized in Table 6. Surprisingly, larger penalties do not have a major impact on the system's profit: the proportion of failed bookings decreases quickly as the penalties

increase, which avoids large profit losses. For the smallest penalty level $\gamma_{\mathrm{P}}^1 = 0.1$, about 5 of the 81 booked drivers are paid penalties, while at the highest penalty level, this is the case for less than 1 booked driver on average. However, higher penalties result in less driver bookings, and therefore also in less rider matches. This explains the small decrease in the profit. Higher penalties may therefore be a good "marketing" strategy to attract new drivers, as they are likely to be matched; even if they are not, they can expect a large compensation. Additionally, from a user perspective, the average trip cost remains relatively stable. Penalty values at around $\gamma_{\mathrm{P}}^1 = 0.5$ therefore seem to provide a good trade-off, given that both the profit and the user booking percentages are relatively high, while a 50% compensation is still sufficiently high to attract new drivers to the user-base. Finally, we note that larger penalties increase the gap to the wait-and-see solution. This is explained by the larger penalty costs that are paid by the stochastic solution, while the wait-and-see solution avoids all penalties due to perfect knowledge.

Joint impact of different revenue share and penalty levels. While the above explores the impact of increasing revenue share and penalty levels separately, we now investigate the impact of changing both parameters at the same time. We consider all combinations of $\gamma_{\mathrm{R}}^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\gamma_{\mathrm{P}}^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$ (for a fixed driver availability window coefficient $\tau_{\mathrm{D}}^1 = 1.3$).

Figure 2: Impact of different values of γ_{R}^1 and γ_{P}^1 on key performance indicators (averaged over instances with $\tau_{\mathrm{D}}^1 = 1.3$, $\gamma_{\mathrm{R}}^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and $\gamma_{\mathrm{P}}^1 \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.1, 1.3, 1.5\}$).

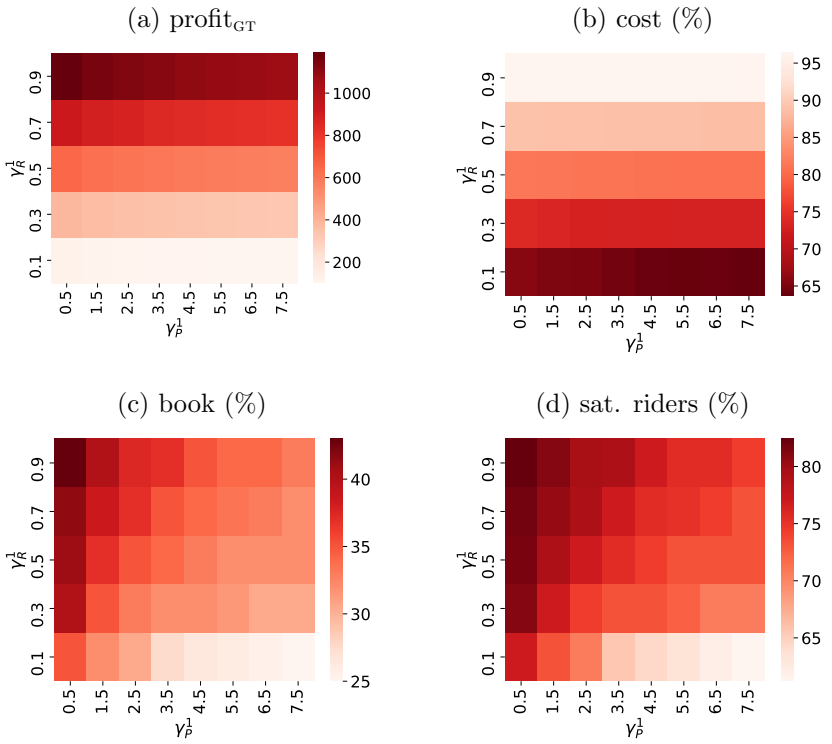


Figure 2 illustrates the impact of the different parameter value combinations on the four

most important key performance indicators: the operator profit, the remaining costs for users, the percentage of booked drivers and the percentage of matched rider requests. Generally, these results confirm the previous observations: operator profits and user costs are inversely related in regards to the profit share. It is interesting to note that system profit and user costs roughly sum to 100%, which is not obvious, given that the latter is computed as the cost savings (directly correlated to the total travel distance) that a user can obtain by sharing a ride as when opposed to driving alone. For the operator, this suggests an important trade-off, which will be mostly dictated by the importance of the user savings. For example, if the system aims at an average cost of about 70% for the riders and drivers, the operator should expect that the revenue share γ_{R}^1 has to be kept below 30% (i.e., a value of 0.3).

In contrast, the penalty level only marginally impacts the operator profit and user costs, but considerably impacts the percentages of booked drivers and matched rider requests. In order to keep users happy, it appears to be beneficial to keep the penalty value at a reasonably low value around $\gamma_{\text{P}}^1 = 0.5$. As shown in the previous study, this may slightly increase the percentage of drivers that have been booked, but not matched; however, in such rare cases, a 50% compensation of the total trip costs are likely to be sufficiently attractive to drivers.

5.2.5 PEN-2: larger availability windows at the cost of higher penalties

We now assess whether large availability windows are a more beneficial choice, even in the case of larger penalty values. To this end, we assess problem variant PEN-2 with level-1 driver availability windows of ($\tau_{\text{D}}^1 = 1.3$ and larger level-2 availability windows of $\tau_{\text{D}}^2 = 1.4$). Larger availability windows clearly lead to a higher matching flexibility, as more ridesharing routes become time-feasible. To properly compensate level-2 drivers for the inconvenience, we evaluate the impact of larger values of level-2 penalties. Specifically, we test $\gamma_{\text{P}}^2 \in \{\, 0.5, 1, 2, 4, 8, 16 \,\}$, while the penalty for level-1 drivers remains at the same (smaller) value of $\gamma_{\text{P}}^1 = 0.3$. Note that all experiments assume a fixed revenue share of $\gamma_{\text{R}}^1 = \gamma_{\text{R}}^2 = 0.3$, which has been found in previous experiments to be a beneficial trade-off between operator profits and savings for users.

Table 7: SAA-R results for different values of γ_{P}^2 on variant PEN-2 ($\tau_{\text{D}}^1 = 1.3$, $\tau_{\text{D}}^2 = 1.4$ and $\gamma_{\text{P}}^1 = 0.3$).

γ_{P}^2	$\text{profit}_{\text{GT}}$	gap_{WS} (%)	book L1 (%)	book L2 (%)	failed L1 (%)	failed L2 (%)	sat. riders (%)	cost (%)	T (sec)
0.5	271.22	8.04	4.04	30.81	10.27	0.61	87.32	68.71	412.58
1.0	269.32	8.68	6.26	29.42	10.53	0.28	87.84	69.00	801.47
2.0	266.79	9.53	7.73	26.96	9.10	0.14	86.47	68.66	333.48
4.0	263.26	10.73	8.19	26.97	8.97	0.13	86.45	68.89	241.93
8.0	256.42	13.03	8.20	25.97	8.90	0.12	85.79	68.54	179.23
16.0	249.34	15.43	8.26	25.48	8.26	0.09	84.95	68.46	267.50

The results are summarized in Table 7. Columns “book L1 (%)” and “book L2 (%)” indicate the average percentages of all drivers booked at levels 1 and 2, respectively. Columns “failed L1 (%)” and “failed L2 (%)” show the average proportions of booked, but unmatched level-1 and level-2 drivers, respectively. The results show that the optimal solutions book a much larger percentage with larger availability windows (i.e., level-2 drivers), even when faced with unrealistically high

penalties (such as $\gamma_{\text{P}}^2 = 16$). Indeed, larger availability windows reduce the likelihood of failing a booking (from about 10% for level-1 drivers to at most 0.61% for level-2 drivers), as drivers have more flexibility. In absolute terms, far less than 1 out of the more than 50 booked level-2 drivers remain unmatched on average. The monetary risk to the ridesharing platform is therefore marginal, even when penalties are high. Interestingly, trip costs for users remain stable. On the downside, larger penalties imply that a smaller proportion of the available drivers is booked, Nevertheless, high penalties may be an interesting temporary "marketing strategy" to attract a large driver user-base in the near-future without sacrificing much profit. Finally, as observed in previous experiments, larger penalties increase the gaps to the wait-and-see solution. This is expected, given that the latter has perfect knowledge on rider requests and can therefore avoid penalties.

On the EVP performance for variant PEN-2. Intrigued by the rather bad performance of the EVP for problem variant PEN-2 (see Table 2), we now explore the model performance for different penalty values γ_{P}^2 in Table 8. As before, these experiments involve high penalties for level-2 drivers, ranging from $\gamma_{\text{P}}^2 = 0.5$ to 16.

Table 8: EVP results for different values of γ_{P}^2 on variant PEN-2 ($\tau_{\text{D}}^1 = 1.3$, $\tau_{\text{D}}^2 = 1.4$ and $\gamma_{\text{P}}^1 = 0.3$).

γ_{P}^2	$\text{profit}_{\text{GT}}$	gap_{WS} (%)	book L1 (%)	book L2 (%)	failed L1 (%)	failed L2 (%)	sat. riders (%)	cost (%)	T (sec)
0.5	260.25	11.74	6.14	30.78	12.29	2.29	89.66	69.04	1.55
1.0	239.89	18.62	3.40	33.56	14.89	2.72	89.75	69.05	1.92
2.0	211.54	28.20	3.72	33.20	17.23	2.35	89.65	69.05	1.83
4.0	112.40	61.69	3.02	33.97	13.68	2.94	89.73	69.07	1.84
8.0	30.27	89.43	4.19	32.74	16.27	2.27	89.68	69.06	1.87
16.0	-303.91	202.33	4.42	32.57	11.64	2.76	89.65	69.01	1.62

The results are quite conclusive and highlight the negative consequences of the EVP's inability to discriminate the probabilistic risk of failed driver bookings. The EVP solution books more level-2 drivers than the SAA-R solution (32.57% vs. 25.48% for $\gamma_{\text{P}}^2 = 16$; compare Table 7), which allows for a high flexibility to match the fractional rider requests. However, the proportion of failed level-2 bookings remains relatively high when compared to the SAA-R solutions (2.76% vs. 0.09% for $\gamma_{\text{P}}^2 = 16$), resulting in several penalties when evaluated on the ground truth. We expect these results to generalize to other problem variants if large penalty values are evaluated. This, in turn, may ultimately jeopardize the profitability of the system. In practice, this is likely to be unacceptable and makes the EVP an unnecessarily risky choice as a planning model.

5.2.6 FEE-2: larger availability windows at the cost of higher booking fees

We now investigate whether the conclusions from the previous section also hold in the case of the problem variant, where booking fees are paid instead of penalties. To this end, we consider problem variant FEE-2 with two different booking levels. Drivers can be booked at level 1 for a fee of $f_i^1 = 1$, which are available for 45 minutes. Alternatively, drivers can be booked at level 2, for a larger time window of 1 hour and a higher booking fee of $f_i^2 \in \{ 1.5, 2.0, 2.5, 3.0, 3.5, 4.0 \}$.

Table 9: SAA-R results for different values of f_i^2 on variant FEE-2 ($f_i^1 = 1$; level-1 drivers are available for 45 minutes; level-2 drivers are available for 1 hour).

f_i^2	$\text{profit}_{\text{GT}}$	gap_{WS} (%)	opt. gap (%)	book L1 (%)	book L2 (%)	failed L1 (%)	failed L2 (%)	sat. riders (%)	cost (%)	T (sec)
1.5	148.71	10.82	0.68	1.31	16.44	12.58	13.49	78.89	71.63	3,600.00
2.0	131.71	14.16	0.93	3.07	14.06	23.29	12.14	76.19	71.47	3,600.00
2.5	117.34	16.69	1.04	4.01	12.58	27.00	10.15	73.74	71.42	3,600.00
3.0	105.98	18.01	0.43	5.03	11.08	22.62	7.68	71.27	71.46	3,600.00
3.5	94.32	20.24	0.61	5.46	10.16	27.19	7.67	69.07	71.41	3,600.00
4.0	84.04	21.99	0.57	5.93	9.33	25.92	6.68	66.92	71.53	3,600.00

The results are summarized in Table 9. As the models tend to exceed the CPU time limit for this problem variant, column "opt. gap (%)" shows the optimality gap of the best known solution as reported by CPLEX. Even though CPLEX did not prove optimality for any instance, the final optimality gaps are rather small. The difficulty of solving problem instances for this problem variant can be explained by the fact that drivers have no fixed destination, and are therefore more flexible. This leads to a larger number of feasible routes when compared to other problem variants (as previously shown in Table 1), which leads to larger formulations that require more computational resources to be solved.

Although the proportion of booked level-2 driver decreases and the proportion of booked level-1 drivers increases with larger level-2 penalties, the system tends to prioritize level-2 drivers. This highlights the benefit of larger availability windows, which is in line with previous findings for other problem variants (see Sections 5.2.3 and 5.2.5). However, in contrast to variant PEN-2, fees that are paid for every booked driver, and not only for those who remain unmatched, result in less driver bookings in general, and therefore in less matched rider requests. Larger level-2 booking fees naturally amplify this phenomena: the system starts matching more level-1 drivers, which further reduces the number of successful matches, and therefore reduces the overall system profit. The percentage of failed bookings are relatively high on both levels, given that no additional penalty is paid if a booked driver remains unmatched. For a similar reason, such higher number of failed booking do not imply less satisfied drivers, since drivers are compensated in either case. Finally, similar to previous results, the gaps to the wait-and-see solution are relatively high, given that the latter can avoid paying booking fees to drivers that are not matched.

5.2.7 PEN-FEE: impact of mixed booking costs and penalties

As a final set of experiments, we now focus on variant PEN-FEE, a context where drivers can be booked either by means of penalties (booking level 1) or by a fixed booking fee without risk of future penalties (booking level 2). Specifically, level-1 drivers can be booked with potential penalty scaling values $\gamma_{\text{P}}^1 \in \{ 0.5, 1, 1.5, 2, 2.5, 3, 8, 16 \}$. Level-2 drivers are booked without penalties, but with a constant booking fee $f_i^2 = 1, \forall i \in D$. Both booking levels have the same relative availability window scaling values $\tau_{\text{D}}^1 = \tau_{\text{D}}^2 = 1.3$ and revenue share $\gamma_{\text{R}}^1 = \gamma_{\text{R}}^2 = 0.3$.

Table 10: SAA-R results for different values of γ^1_{P} on variant PEN-FEE ($\tau^1_{\text{D}} = \tau^2_{\text{D}} = 1.3$, $\gamma^1_{\text{R}} = \gamma^2_{\text{R}} = 0.3$ and $f^2_i = 1$).

γ^1_{P}	$\text{profit}_{\text{GT}}$	gap_{WS} (%)	book L1 (%)	book L2 (%)	failed L1 (%)	failed L2 (%)	sat. riders (%)	cost (%)	T (sec)
0.5	369.27	9.34	29.08	6.41	1.41	32.48	80.65	73.59	16.39
1.0	363.25	10.81	25.33	9.51	0.68	26.86	80.28	73.26	16.77
1.5	360.86	11.40	24.38	10.36	0.50	26.03	80.28	73.21	14.77
2.0	358.69	11.93	23.41	11.32	0.39	24.51	80.28	73.21	19.34
2.5	356.75	12.40	22.43	12.13	0.33	24.21	80.38	73.07	17.23
3.0	355.56	12.69	21.94	12.35	0.26	22.83	79.95	73.03	16.29
8.0	346.81	14.84	20.96	13.60	0.18	22.27	80.24	73.06	26.64
16.0	338.49	16.87	19.97	14.59	0.13	21.14	80.24	73.05	23.69

The results, summarized in Table 10, indicate that, for lower penalty values, the system favors level-1 drivers (those that potentially imply penalties) over level-2 drivers (those paid booking fees). While the proportion of level-1 drivers decreases as the penalties increase, even for unrealistically large penalty values (i.e., $\gamma^1_{\text{P}} = 16$), we do not observe an inflection point at which booking fees become more favorable. Such behavior is mostly motivated by the underlying percentages of booked, but unmatched drivers: the proportion of failed level-1 drivers is negligibly small, while the proportion of failed level-2 drivers is quite high (but without economic impact). For $\gamma^1_{\text{P}} = 16$, the average penalty for an unmatched level-1 driver would is 335.85, which may pose considerable risks to the ridesharing platform, as a single penalty paid would almost be in the same orders of the entire profit. On average, however, only 0.05 level-1 drivers remain unmatched (200 drivers $\times 19.97\% \times 0.13\%$). Among all booked level-2 drivers, only 4 to 6 remain unmatched, on average (and in this case, without any economic impact).

Similar to previous results, larger penalties only have a limited negative impact on the profit, as the system rarely fails a level-1 booking and has to pay the penalty. Further, larger penalties lead to larger gaps to the wait-and-see solution, similar to previous results. All other performance metrics remain similar, which confirms the conclusions of previous experiments, suggesting that penalties are an attractive driver compensation mechanism to ridesharing operator (as opposed to booking fees), while maintaining an overall customer satisfaction.

6 Conclusions

We have introduced a general modeling framework for ridesharing systems, where drivers have to be selected before driver requests become available. This framework allows for the representation of many-to-one ridesharing systems, and addresses the uncertainty of rider requests while taking into consideration compensation strategies for drivers (booking fees and penalties for not providing a rideshare). The mixed-integer programming modeling framework consists of a two-stage set-packing model with stochastic nodes. We exemplify the usage of this model by means of three instances of ridesharing system operating modes and identify the conditions under which the here proposed problem also solves the more general multi-stage problem variant used in a dynamic context.

We evaluate three models that approximate the second-stage value function: the SAA model, the expected value model, and the SAA model with a relaxed second-stage problem (SAA-R).

Our computational results show that the SAA and SAA-R models generate better solutions, while the latter is solved within a a fraction of the required computing time.

We then carry out extensive sets of computational experiments in order to derive managerial insights on the benefits of the different operating modes, as well as on parameter values that ensure both the profitability of the system and a high user satisfaction. Specifically: 1) larger availability windows allow for more matching flexibility and ultimately lead to a higher number of successful matched and hence a higher platform profit; 2) large penalties do not necessarily jeopardize the profitability of the system, given that optimal planning solutions will adjust the driver bookings in order to keep failed bookings low, and therefore mostly avoid penalties; this implies that announcing temporarily high penalty compensation may be valid marketing strategy to increase the driver user-base; 3) if possible, a driver compensation mechanism via penalties may be preferable to system operators over fixed booking fees, if penalties can be mostly avoided; 4) while a higher revenue share kept by the operator also implies more profit, it is inversely related to the savings for the users; as such, we identify revenue share and penalty parameter values that provide an attractive trade-off between overall system profit and customer satisfaction.

This research opens several promising research directions. From a practical viewpoint, it may be worthwhile investigating multi-stage problem variants that cannot be solved exactly by the here proposed two-stage stochastic model (e.g., variants where drivers can be selected at any time during the day, as well as the possibility of un- and re-matching drivers and riders). From a methodological point of view, mathematical decomposition algorithms may be employed to solve the here proposed models faster and on larger scale. In this regards, integrating route-generation via column generation may be a promising avenue to solve the SAA, while Benders decomposition may be employed to solve the SAA-R.

Acknowledgements

We would like to thank Netlift and Mathieu Lozeau for their thoughtful input, domain knowledge, and for providing us with access to data. We are also grateful to Compute Canada for providing computational resources. The work of the first author was supported by the Chaire en transformation du transport (Université de Montréal/Polytechnique Montréal), which is funded by the Ministère de l'Économie, de l'Innovation et de l'Énergie du Québec. The work of the third author was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada under grant 2017-05224 and the Fonds de Recherche Nature et Technologies (FRQNT) under grant 70739202.

References

N. Agatz, A. Erera, M. Savelsbergh, and X. Wang. Optimization for dynamic ride-sharing: A review. *European Journal of Operational Research*, 223(2):295–303, 2012. ISSN 03772217. doi: 10.1016/j.ejor.2012.05.028. URL `http://dx.doi.org/10.1016/j.ejor.2012.05.028`.

N. A. Agatz, A. L. Erera, M. W. Savelsbergh, and X. Wang. Dynamic ride-sharing: A simulation study in metro Atlanta. *Transportation Research Part B: Methodological*, 45 '(9):1450–1464, 2011. ISSN 01912615. doi: 10.1016/j.trb.2011.05.017. URL `http://dx.doi.org/10.1016/j.sbspro.2011.04.530`.

S. Ahmed, A. J. King, and G. Parija. A multi-stage stochastic integer programming approach for capacity expansion under uncertainty. *Journal of Global Optimization*, 26(1):3–24, 2003.

R. Baldacci, E. Hadjiconstantinou, V. Maniezzo, and A. Mingozzi. A new method for solving capacitated location problems based on a set partitioning approach. *Computers and Operations Research*, 29(4):365–386, 2002. ISSN 03050548. doi: 10.1016/S0305-0548(00)00072-1.

R. Baldacci, V. Maniezzo, and A. Mingozzi. An Exact Method for the Car Pooling Problem Based on Lagrangean Column Generation. *Operations Research*, 52(3):422–439, 2004. ISSN 0030-364X. doi: 10.1287/opre.1030.0106.

D. Bertsimas, P. Jaillet, and S. Martin. Online Vehicle Routing: The Edge of Optimization in Large-Scale Applications. *Operations Research*, 67(1):143–162, 2019. ISSN 0030-364X. doi: 10.1287/opre.2018.1763.

J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer New York, 2011. doi: 10.1007/978-1-4614-0237-4.

R. Borndörfer. *Aspects of set packing, partitioning, and covering*. PhD thesis, 1998.

N. D. Chan and S. A. Shaheen. Ridesharing in North America: Past, Present, and Future. *Transport Reviews*, 32(1):93–112, 2012.

L. Costa, C. Contardo, and G. Desaulniers. *Exact branch-price-and-cut algorithms for vehicle routing*, volume 53. 2019.

L. F. Escudero, M. Landete, and A. M. Rodríguez-Chía. Stochastic set packing problem. *European Journal of Operational Research*, 211(2):232–240, 2011. ISSN 03772217. doi: 10.1016/j.ejor.2010.11.022.

M. Furuhata, M. Dessouky, F. Ordóñez, M. E. Brunet, X. Wang, and S. Koenig. Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46, 2013. ISSN 01912615. doi: 10.1016/j.trb.2013.08.012. URL `http://dx.doi.org/10.1016/j.trb.2013.08.012`.

G. Giuliano, D. W. Levine, and R. F. Teal. Impact of high occupancy vehicle lanes on carpooling behavior. *Transportation*, 17(2):159–177, 1990. ISSN 00494488. doi: 10.1007/BF02125334.

G. Homsi, B. Gendron, and S. D. Jena. *Dynamic and Stochastic Rematching for Ridesharing Systems : Formulations and Reductions*, volume 1. Springer International Publishing, 2020.

G. Homsi, B. Gendron, and S. D. Jena. Rolling horizon strategies for a dynamic and stochastic ridesharing problem with rematches. Technical report, CIRRELT, 2021.

A. Lee and M. Savelsbergh. Dynamic ridesharing: Is there a role for dedicated drivers? *Transportation Research Part B: Methodological*, 81:483–497, 2015. ISSN 01912615. doi: 10.1016/j.trb.2015.02.013. URL `http://dx.doi.org/10.1016/j.trb.2015.02.013`.

J. K. Lenstra and A. R. Kan. Complexity of vehicle routing and scheduling problems. *Networks*, 11(2):221–227, 1981.

C. Levinger, N. Hazon, and A. Azaria. Computing the shapley value for ride-sharing and routing games. *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2020-May:1895–1897, 2020. ISSN 15582914.

D. Luxen and C. Vetter. Real-time routing with openstreetmap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '11, pages 513–516, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1031-4. doi: 10.1145/2093973.2094062. URL `http://doi.acm.org/10.1145/2093973.2094062`.

A. Madansky. Inequalities for Stochastic Linear Programming Problems. *Management Science*, 6 (2):197–204, 1960.

L. d. C. Martins, R. de la Torre, C. G. Corlu, A. A. Juan, and M. A. Masmoudi. Optimizing ride-sharing operations in smart sustainable cities: Challenges and the need for agile algorithms. *Computers and Industrial Engineering*, 153(March 2020):107080, 2021. ISSN 03608352. doi: 10.1016/j.cie.2020.107080.

N. McGuckin and A. Fucci. Summary of travel trends: 2017 national household travel survey. *Washington, DC: Federal Highway Administration, US Department of Transportation*, 2018.

A. Mourad, J. Puchinger, and C. Chu. A survey of models and algorithms for optimizing shared mobility. *Transportation Research Part B: Methodological*, 123:323–346, 2019. ISSN 01912615. doi: 10.1016/j.trb.2019.02.003. URL `https://doi.org/10.1016/j.trb.2019.02.003`.

S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems. *Journal fur Betriebswirtschaft*, 58(1):21–51, 2008a.

S. N. Parragh, K. F. Doerner, and R. F. Hartl. A survey on pickup and delivery problems: Part II: Transportation between pickup and delivery locations. *Journal fur Betriebswirtschaft*, 58(2): 81–117, 2008b.

Z. Peng, W. Shan, X. Zhu, and B. Yu. Many-to-one stable matching for taxi-sharing service with selfish players. *Transportation Research Part A*, 160(April):255–279, 2022. ISSN 0965-8564. doi: 10.1016/j.tra.2022.04.012.

C. Riley, A. Legrain, and P. Van Hentenryck. Column generation for real-time ride-sharing operations. In L.-M. Rousseau and K. Stergiou, editors, *Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 472–487, Cham, 2019. Springer International Publishing. ISBN 978-3-030-19212-9.

H. Si, Y. Su, G. Wu, W. Li, and L. Cheng. Can government regulation , carbon-emission reduction certification and information publicity promote carpooling behavior ? *Transportation Research Part D*, 109(July):103384, 2022. ISSN 1361-9209. doi: 10.1016/j.trd.2022.103384.

UN. Revision of world urbanization prospects. *United Nations: New York, NY, USA*, 799, 2018.

X. Wang, N. Agatz, and A. Erera. Stable matching for dynamic ride-sharing systems. *Transportation Science*, 52(4):850–867, 2018. ISSN 15265447. doi: 10.1287/trsc.2017.0768.